



Prédiction de l'instabilité dynamique des réseaux électriques par apprentissage supervisé des signaux de réponses post-contingence sur des dictionnaires surcomplets

Thèse

Raoult Teukam Dabou

Doctorat en génie électrique
Philosophiæ doctor (Ph. D.)

Québec, Canada

© Raoult Teukam Dabou, 2022

**Prédiction de l'instabilité dynamique des réseaux
électriques par apprentissage supervisé des signaux de
réponses post-contingence sur des dictionnaires
surcomplets**

Thèse

Raoult Teukam Dabou

Sous la direction de :

Innocent Kamwa, Directeur de Recherche

Résumé

Ces dernières décennies, l'intégration aux réseaux électriques de capteurs intelligents incorporant la mesure synchronisée des phaseurs a contribué à enrichir considérablement les bases de données de surveillance en temps réel de la stabilité des réseaux électriques. En parallèle, la lutte aux changements climatiques s'est accompagnée d'un déploiement généralisé des sources d'énergies renouvelables dont l'intermittence de la production et le déficit d'inertie dû à l'interface de celle-ci par l'électronique de puissance, contribuent à augmenter les risques d'instabilité à la suite de contingences de réseau. Dans ce contexte, nous proposons d'appliquer aux données de synchrophaseurs de nouvelles approches d'intelligence de données inspirées par l'analyse massive des séries chronologiques et l'apprentissage sur des dictionnaires supervisés, permettant d'extraire des centaines d'attributs décrivant concisément les estimations d'état dynamique des générateurs de réseaux électriques. La mise en évidence d'une signification physique de ces attributs permet ensuite une classification de la stabilité dynamique qui s'éloigne de ce fait des boîtes noires produites par un apprentissage en profondeur « à l'aveugle » des séries chronologiques, pour évoluer vers une approche transparente plus adaptée à la salle de conduite des réseaux et acceptable pour les ingénieurs d'exploitation. Cette approche d'apprentissage machine « interprétable » par les humains, débouche de surcroît sur une détection fiable, utilisant de courtes fenêtres de données de vitesses d'alternateurs directement mesurées ou reconstituées par estimation d'état dynamique à partir de l'instant d'élimination du défaut, pour détecter toute instabilité subséquente, avec un temps de préemption suffisant pour activer des contremesures permettant de sauvegarder la stabilité du réseau et ainsi prévenir les pannes majeures.

Notre travail aborde l'exploitation de cette nouvelle niche d'information par deux approches complémentaires d'intelligence des données : 1) une analyse non parcimonieuse d'une base d'attributs se chiffrant par centaines, calculés automatiquement par l'analyse numérique massive des séries chronologiques de signaux de réponses post-contingence des générateurs; et 2) une analyse parcimonieuse exploitant l'apprentissage supervisée de grands dictionnaires surcomplets pour habiliter une prédiction de l'instabilité sur de courtes fenêtres

de données avec une représentation vectorielle creuse (contenant un grand nombre de zéros) et donc numériquement très efficace en plus de l'interprétabilité inhérente des atomes constituant les dictionnaires. Au niveau méthodologique, l'approche non parcimonieuse vise à implémenter plusieurs méthodes analytiques combinées (notamment la transformée de Fourier, la transformée en ondelette, la méthode de Welch, la méthode de périodogramme et les exposants de Lyapunov) pour extraire du signal de réponse de chaque générateur des centaines d'attributs labellisés et servant à construire un espace physique d'indicateurs de stabilité à haute dimension (HDSI). Ceux-ci sont ensuite utilisés pour développer les prédictors de stabilité sur la base d'algorithmes standard de machine learning, par exemple le *convolutional neural network* (CNN), *long short-term memory* (LSTM), support vector machine (SVM), AdaBoost ou les forêts aléatoires. L'approche parcimonieuse implémentée consiste à développer deux techniques complémentaires : 1) un dictionnaire d'apprentissage supervisé joint (SLOD) au classificateur et 2) vingt dictionnaires d'apprentissage séparés des signaux associés aux cas stable/instable. Alors que le SLOD utilise des dictionnaires adaptatifs inspirés des données mesurées et apprises hors-ligne, la deuxième approche utilise des dictionnaires fixes pour reconstruire séparément les signaux des classes stables et instables. Dans les deux cas, l'étape finale consiste à identifier automatiquement en temps réel, la classe d'appartenance d'une réponse par reconstruction des signaux associés à partir des dictionnaires appris hors-ligne.

L'analyse parcimonieuse des réponses des générateurs sur un dictionnaire d'apprentissage adaptatif joint au classificateur a été implémenté à partir de l'algorithme *K-singular value decomposition* (KSVD) couplé à l'*orthogonal matching pursuit* (OMP), afin de reconstruire et prédire la stabilité dynamique des réseaux électriques. De plus, vingt décompositions parcimonieuses des signaux sur des dictionnaires fixes (simples et hybrides) ont permis de développer des classificateurs prédisant chaque classe séparément sur la base de la transformée en cosinus discrète (DCT), en sinus discrète (DST), en ondelette (DWT), de la transformée de Haar (DHT), et le dictionnaire de Dirac (DI) couplés à l'*orthogonal matching pursuit* (OMP). Cette étude démontre que la décomposition parcimonieuse sur un dictionnaire adaptatif joint au classificateur offre une performance proche de l'idéal (c'est-à-dire : 99,99 % précision, 99,99 % sécurité et 99,99 % fiabilité) de loin supérieure à celle d'un classificateur à reconstruction de signaux basée sur les vingt dictionnaires fixes ou adaptatifs

séparés, et les classificateurs basés sur les moteurs de *machine learning* (SVM, ANN, DT, RF, AdaBoost, CNN et LSTM) implémentés à partir des indices HDSI extraits de la base de données des vitesses des rotors des réseaux IEEE 2 area 4 machines, IEEE 39 -bus et IEEE 68 -bus. Toutefois, le temps de resimulation (*replay*) en temps réel des dictionnaires fixes/adaptatifs séparés est nettement inférieur (de 30-40%) à celui observé pour le dictionnaire adaptatif à classificateur joint / SLOD, et les algorithmes modernes de *machine learning* utilisant les attributs de type HDSI comme intrants.

Abstract

In recent decades, the integration of smart sensors incorporating synchronized phasor measurements units (PMU) into power grids has contributed to a significant improvement of the databases for real-time monitoring of power grid stability. In parallel, the fight against climate change has been accompanied by a widespread deployment of renewable energy sources whose intermittency of production and the lack of inertia due to the interface of the latter by power electronics; contribute to increase the risks of instability following network contingencies. In this context, we propose to apply new data intelligence approaches inspired by massive time series analysis and supervised dictionary learning to synchrophasor data, allowing the extraction of hundreds of attributes concisely describing the dynamic state estimates of power system generators. The physical meaning identification of these attributes then allows for an online classification of dynamic stability, thus moving away from the black boxes produced by «blind» deep learning of time series to a transparent approach more suitable for the network control room and acceptable to operating engineers. This human-interpretable machine learning approach also leads to reliable detection, using short windows of generator speed data directly measured or reconstructed by dynamic state estimation from the instant of fault elimination, to detect any subsequent instability, with sufficient preemption time to activate false measures to safeguard the network stability and thus prevent major outages.

Our work addresses the exploitation of this new information through two complementary data intelligence approaches : 1) a non-sparse analysis of an attribute base numbering in the hundreds, computed automatically by massive numerical analysis of post-contingency response signal time series from generators; and 2) a sparse analysis exploiting supervised learning of large overcomplete dictionaries to enable instability prediction over short windows of data with a hollow vector representation (containing a large number of zeros) and thus numerically very efficient in addition to the inherent interpretability of the atoms constituting the dictionaries. Methodologically, the non-sparse approach aims to implement several combined analytical methods (including Fourier transform, wavelet transform, Welch's method, periodogram method and Lyapunov exponents) to extract hundreds of

labeled attributes from the response signal of each generator and used to construct a physical space of high-dimensional stability indicators (HDSI). These are used to develop stability predictors based on standard machine learning algorithms, e.g., CNN, LSTM, SVM, AdaBoost or random forests. The implemented sparse approach consists in developing two complementary techniques: 1) a supervised learning dictionary attached (SLOD) to the classifier and 2) twenty separate dictionaries learning of the signals associated with the stable/unstable cases. While the SLOD uses adaptive dictionaries inspired by the measured and learned offline data, the second approach uses fixed dictionaries to reconstruct the stable and unstable signals classes separately. In both cases, the final step is automatically identified in real time the status to which a response belongs by reconstructing the associated signals from the off-line learned dictionaries.

The sparse analysis of generator responses on an adaptive learning dictionary attached to the classifier was implemented using the K-singular value decomposition (KSVD) algorithm coupled with orthogonal matching pursuit (OMP), to reconstruct and predict online the dynamic stability of power systems. In addition, twenty sparse signal decompositions on fixed dictionaries (simple and hybrid) were used to develop classifiers predicting each class separately based on the discrete cosine transform (DCT), discrete sine transform (DST), wavelet transform (DWT), Haar transform (DHT), and Dirac dictionary (DI) coupled with the orthogonal matching pursuit (OMP). This study demonstrates that sparse decomposition on joined adaptive dictionary to the classifier provides near ideal performance (i.e.: 99.99% accuracy, 99.99% security, and 99.99% reliability) far superior to that of a classifier has signal reconstruction based on the twenty separate fixed or adaptive dictionaries and classifiers based on machine learning engines (SVM, ANN, DT, RF, AdaBoost, CNN, and LSTM) implemented from HDSI indices extracted from the rotor speed database of the IEEE 2 area 4 machines, IEEE 39 -bus, and IEEE 68 -bus test systems. However, the real-time replay time of the separate fixed/adaptive dictionaries is significantly lower (by 30-40%) than that observed for the adaptive dictionary with joint classifier/SLOD, and modern machine learning algorithms using HDSI-like attributes as inputs.

Table des matières

Résumé	iii
Abstract.....	vi
Table des matières	viii
Liste des figures.....	xiii
Liste des tableaux	xv
Liste des acronymes	xvi
Liste des symboles.....	xix
Remerciements	xxi
Liste des publications	xxii
Introduction Générale.....	1
0.1 Contexte de recherche.....	2
0.2 Problème de recherche.....	8
0.2.1 Problème général.....	8
0.2.2 Problèmes spécifiques	9
0.3 Objectifs de recherche.....	11
0.4 Contribution originale.....	13
0.4 Organisation de la thèse	16
Chapitre 1 — Revue Bibliographique	19
1.1 Introduction.....	19
1.2 Base de données des signaux électriques.....	19
1.3 Extraction et prédiction des indicateurs d’instabilités	20
1.4 Dictionnaires fixes et adaptatifs pour l’extraction et classification	25
1.5 Conclusion et pistes d’amélioration.....	29

1.5.1 Synthèse	29
1.5.2 Propositions.....	31
Chapitre 2 — Extraction et classification des attributs de stabilité à haute dimension.....	33
2.1 Introduction.....	33
2.2 Génération des bases de données de stabilité sur Simscape Power System (SPS)....	33
2.2.1 Estimation de l'état dynamique.....	33
2.2.2 Algorithme de génération graduelle des signaux de stabilité sur SPS	34
2.2.3 Détection d'instabilité du réseau électrique	35
2.3 Approche d'extraction des attributs de stabilité	36
2.3.2 Approche conventionnelle d'extraction à faible dimension.....	38
2.3.3 Nouvelle approche d'extraction à haute dimension	38
2.3.3.1 Les Exposants de Lyapunov	39
2.3.3.2 Analyse spectrale.....	40
2.3.3.3 Transformée en ondelette	41
2.4 Classification des indicateurs de stabilité	41
2.4.1 Texture des attributs pour la machine learning	42
2.4.2 Machine à vecteurs de support.....	43
2.4.3 Arbres de décision.....	49
2.4.4 Forêt aléatoire.....	50
2.4.5 Réseaux de neurones artificiels	52
2.4.6 Adaptive boosting	53
2.5 Évaluation des performances de classification	54
2.5.1 Précision.....	55
2.5.2 Fiabilité	55
2.5.3 Sécurité.....	55

2.6 Conclusion	58
Chapitre 3 — Dictionnaire d'apprentissage joint pour la prédiction de la stabilité dynamique	59
3.1 Introduction.....	59
3.2 Méthode de recherche des limites de stabilité	59
3.3 Développement d'un dictionnaire joint au classificateur.....	65
3.3.1 Extraction des attributs parcimonieux pour la DSP	66
3.3.2 Reconstruction et classification des réponses des générateurs.....	69
3.4 Comparaison du SLOD et des algorithmes de machine learning	75
3.5 Conclusion	81
Chapitre 4 — Apprentissage séparé des dictionnaires pour la prédiction de la stabilité dynamique	82
4.1 Introduction.....	82
4.2 Dictionnaire fixe et adaptatif stable/instable.....	83
4.2.1 Dictionnaire d'apprentissage fixe et simple.....	84
4.2.1.1 Transformée en Cosinus Discrète.....	84
4.2.1.2 Transformée en Sinus Discrète.....	84
4.2.1.3 Transformée de Haar Discrète.....	85
4.2.1.4 Transformée en Ondelette Discrète	85
4.2.1.5 Dictionnaire d'impulsion.....	86
4.2.2 Dictionnaires fixes d'atome hybride à deux concaténations.....	86
4.2.2.1 Dictionnaire hybride de Haar-Cosinus	86
4.2.2.2 Dictionnaire hybride en Ondelette-Haar.....	87
4.2.2.3 Dictionnaire hybride en Cosinus-Ondelette	87
4.2.3 Dictionnaires fixes d'atome hybride à trois concaténations.....	87
4.2.3.1 Dictionnaire hybride de Dirac-Cosinus-Sinus.....	87

4.2.3.2 Dictionnaire hybride de Sinus-Haar-Cosinus	88
4.2.3.3 Dictionnaire hybride en Ondelette-Haar-Sinus	88
2.4.4 Dictionnaires adaptatifs d'atomes	88
4.3 Reconstruction/classification sur des dictionnaires fixes et adaptatifs	89
4.4 Évaluation des performances des dictionnaires d'atome joint et séparé.....	90
4.4.1 Descriptif des données d'apprentissage	90
4.4.2 Extraction des atomes parcimonieux fixes et adaptatifs	91
4.4.3 Reconstruction des signaux à partir des atomes fixes et adaptatifs.....	93
4.4.4 Prédiction de la stabilité à partir des atomes fixes et adaptatifs.....	95
4.4.4.1 Prédiction de la stabilité à partir des dictionnaires simples et adaptatifs	95
4.4.4.2 Prédiction sur des dictionnaires fixes à deux concaténations et adaptatifs ...	96
4.4.4.3 Prédiction sur des dictionnaires fixes à trois concaténations et adaptatifs	97
4.4.4.4 Évaluation des métriques des dictionnaires fixes et adaptatifs.....	98
4.4.4.5 Temps d'apprentissage des dictionnaires fixes et adaptatifs	99
4.5 Conclusion	102
Conclusion Générale et Perspectives.....	103
Références Bibliographiques	105
Annexes	119
A.1 Données des réseaux IEEE 2 Area 4 Machines, IEEE 39 –bus et IEEE 68 –bus...	119
A.1.1 Données IEEE 2 Area 4 Machines	119
A.1.2 Données IEEE 39 -bus	119
A.1.3 Données IEEE 68 -bus	124
A.2 Modèles Simulink des réseaux IEEE 39 –bus et IEEE 68 -bus	129
A.2.1 Modèle Simulink du réseau IEEE 39 -bus	130
A.2.2 Modèle Simulink du réseau IEEE 68 -bus	131

A.3 Code Matlab de génération des signaux de stabilité des données.....	132
A.4 Code Matlab d'extraction des indices de stabilité à haute dimension (HDSI)	188
A.5 Code Matlab d'implémentation du dictionnaire joint au classificateur pour la DSP: DK-SVD	190
A.6 Code Matlab d'implémentation des dictionnaires séparés des classificateurs: DCT, DHT, DWT, K-SVD, [DHT DWT], [DCT DWT], [DHT DCT], [DST DHT DCT], [ID DCT DST], [DWT DHT DCT] pour la DSP.	192
A.7 Code Matlab d'implémentation de l'algorithme SVM pour la DSP.....	196
A.8 Code Matlab d'implémentation de l'algorithme DT pour la DSP	197
A.9 Code Matlab d'implémentation de l'algorithme RF pour la DSP	198
A.10 Code Matlab d'implémentation de l'algorithme ANN pour la DSP.....	198
A.11 Code Matlab d'implémentation de l'algorithme AdaBoost pour la DSP	199
A.12 Code Matlab d'implémentation de l'algorithme CNN pour la DSP	199
A.13 Code Matlab d'implémentation de l'algorithme LSTM pour la DSP.....	200

Liste des figures

Figure 01 : Décomposition parcimonieuse des signaux de stabilité sur un dictionnaire [28].6	
Figure 02 : Diagramme détaillé des différentes étapes constituant notre méthodologie de la recherche.....	15
Figure 1 : Schéma conventionnel (a) et proposé (b) pour l'extraction des attributs [96]....	37
Figure 2 : Bloc diagramme utilisé pour l'extraction des HDSI [96].	39
Figure 3 : Illustration de 10 signaux (a) des vitesses des rotors et (b) RoCoTE stable (bleu) et instable (rouge) [96].	42
Figure 4: Texture des HDSI issues des (a) vitesses des rotors et des indicateurs (b) RoCoTE (Modèle : IEEE 39 -bus) [96].	43
Figure 5: HDSI des (a) vitesses des rotors et (b) RoCoTE issues l'algorithme SVM (Modèle : IEEE 39 bus) [96].	44
Figure 6: Matrice de confusion des vitesses et RoCoTE (Modèle: IEEE 39 -bus) [96].....	45
Figure 7: Matrice de confusion des vitesses et RoCoTE (Modèle: IEEE 68 -bus) [96].....	46
Figure 8: Hiérarchisation des attributs les plus significatifs (Modèle : IEEE 39 -bus) [96].	47
Figure 9: Hiérarchisation des attributs les plus significatifs (Modèle : IEEE 68 -bus) [96].	48
Figure 10: Arbres de decision issus des HDSI des vitesses (Modèle: IEEE 39 bus) [96]. ..	49
Figure 11 : Gini impurity des HDSI des vitesses des rotors (Modèle: IEEE 39 bus) [96]...	50
Figure 12 : Boîte à moustache et distribution spectrale d'un attribut de Welch à HDSI issu des vitesses des rotors (Modèle: IEEE 39+68 bus) [96].....	50
Figure 13: Erreur des forêts aléatoires issue des HDSI (Modèle : IEEE 39 -bus).	51
Figure 14 : Illustration d'une architecture de réseaux de neurones artificiels.....	52
Figure 15: Matrice de confusion des RNNs issus des vitesses (Modèle : IEEE 39 -bus). ...	53
Figure 16 : Erreur du Adaptive Boosting issu des vitesses (Modèle : IEEE 39 -bus).....	54
Figure 17: Organigramme pour l'apprentissage supervisé du dictionnaire surcomplet [40].	60
Figure 18 : Allure des vitesses des rotors instables du IEEE 2 area 4 machines [40].....	63
Figure 19 : Allure des vitesses des rotors instables du IEEE 39 bus [40].	63

Figure 20 : Allure des vitesses des rotors instables du IEEE 68 -bus [40].....	64
Figure 21: Histogrammes des réseaux IEEE 39, -68 bus face aux défauts triphasés terre [40].	65
Figure 22: Décompositions sur des dictionnaires stables et les RMSE associées [40].	71
Figure 23: Décompositions sur des dictionnaires instables et les RMSE associées (en magenta) [40].....	72
Figure 24: Classifications sur des dictionnaires stables et les RMSE associées (en magenta) [40].	73
Figure 25: Classifications sur des dictionnaires instables et les RMSE associées [40].	74
Figure 26: Attributs de stabilité de Welch, du périodogramme et des ondelettes [40].	76
Figure 27: Matrices de confusion des HDSI issues des vitesses des réseaux tests [40].....	78
Figure 28: Organigramme pour d'apprentissage des dictionnaires surcomplets [35].....	83
Figure 29: Allure des vitesses des rotors stables (a) et instables (b) du IEEE 68 -bus [35].	91
Figure 30: Illustration des atomes K-SVD (a) et OHD (b) [35].....	93
Figure 31 : Reconstruction d'un signal stable et instable sur le dictionnaire K-SVD [35].	94
Figure 32 Reconstruction d'un signal stable et instable sur les dictionnaires OHD [35].....	94
Figure 33: Projection d'un signal stable ID 1 et instable ID 23 sur SDL et ADL [35].....	95
Figure 34: Projection d'un signal stable ID 1 et instable ID 23 sur 2H-OHD et ADL [35].	96
Figure 35: Projection d'un signal stable ID 1 et instable ID 23 sur 3H-OHD et ADL [35].	97
Figure 36: Degrés de stabilité des signaux tests ID 1 et ID 23 [35].	98
Figure 37: Comparaison des performances de classification des dictionnaires [35].....	100
Figure 38 : Diagramme unifilaire (a) et configuration Simulink (b) du réseau IEEE 39 –bus utilisée pour la génération des signaux de stabilité.	131
Figure 39: Diagramme unifilaire (a) et configuration Simulink (b) du réseau IEEE 68 –bus utilisée pour la génération des signaux de stabilité.	132

Liste des tableaux

Tableau I : Configuration des scénarios simulés sur IEEE 39 bus.....	35
Tableau II : Matrice de confusion pour DSP	54
Tableau III : Performance des algorithmes de machine learning pour la DSP.....	56
Tableau IV : CPU Time (in sec.) pour la DSP.	57
Tableau V : Configuration des réseaux électriques utilisée pour l'approche SLOD.....	61
Tableau VI : Structure des données d'apprentissage de la SLOD.....	66
Tableau VII : Comparaison de la SLOD et des algorithmes de machine learning.....	79
Tableau VIII : CPU time (in sec.) du SLOD et des algorithmes de machine learning.....	80
Tableau IX : Configuration des signaux utilisés pour la décomposition parcimonieuse.	91
Tableau X : Configuration des dictionnaires d'apprentissage supervisés.	92
Tableau XI : Métrique de comparaison des dictionnaires d'apprentissage supervisés.	99
Tableau XII : Temps CPU (en sec.) des dictionnaires d'apprentissage supervisés.....	100

Liste des acronymes

ADL	Adaptive dictionary learning
ANN	Artificial neural network
AdaBoost	Adaptive boosting
CNN	Convolutional neural network
DST	Discrete sine transform
DCT	Discrete cosine transform
[DHT DCT]	Concatenation of discrete Haar and cosine transforms
[DWT DCT]	Concatenation of discrete wavelet and cosine transforms
[DHT DWT]	Concatenation of discrete Haar and wavelet transforms
[DST DHT DCT]	Concatenation of discrete sine Haar and cosine transforms
[ID DCT DST]	Concatenation of discrete impulse, cosine, and sine transforms
[DWT DHT DCT]	Concatenation of discrete wavelet, Haar and cosine transforms
DSA	Dynamic stability assessment
DWT	Discrete wavelet transform
DHT	Discrete Haar transform
DT	Decision trees
DSP	Dynamic stability prediction
ECA	Expected classification accuracy
GMP	Global matching pursuit
HDSI	High dimensional stability indices
ILS-DLA	Iterative least squares dictionary learning algorithm
2H-OHD	Two overcomplete hybrid dictionaries
3H-OHD	Three overcomplete hybrid dictionaries
ID	Impulse dictionary
K-SVD	K-Singular value decomposition
LSTM	Long short term memory

MOD	Method of optimal direction
ORMP	Order recursive matching pursuit
OHD	Overcomplete hybrid dictionaries
OMP	Orthogonal matching pursuit
ODL	Online dictionary learning
PST	power system Toolbox
PSAT	power system analysis Toolbox
PYPower	Python for power
PYPSA	Python for power system analysis
PMU	Phasor measurement units
RF	Random forest
RoCoTE	Rate of change of transient energy
RLS-DLA	Recursive least squares dictionary learning algorithm
SPS	Simulink for power systems
SOD	Single overcomplete dictionaries
SVM	Support vector machine
SPE	Stable prediction error
Stable DHT	DHT dictionary learned from a stable dataset
Stable DWT	DWT dictionary learned from stable dataset
Stable DCT	DCT dictionary learned from stable dataset
Stable [DHT DCT]	[DHT DCT] dictionary learned from stable datasets
Stable [DWT DCT]	[DWT DCT] dictionary learned from stable datasets
Stable [DHT DWT]	[DHT DWT] dictionary learned from stable datasets
Stable [DST DHT DCT]	[DST DHT DCT] dictionary learned from stable datasets
Stable [ID DCT DST]	[ID DCT DST] dictionary learned from stable datasets
Stable [DWT DHT DCT]	[DWT DHT DCT] dictionary learned from stable datasets
TK-SVD	Transient K-singular value decomposition
TSC	Time series classification
TSA	Transient stability assessment
UPE	Unstable prediction error

Unstable [DWT DHT DCT]	[DWT DHT DCT] dictionary learned from unstable datasets
Unstable DHT	DHT dictionary learned from unstable dataset
Unstable DWT	DWT dictionary learned from unstable dataset
Unstable DCT	DCT dictionary learned from unstable dataset
Unstable [DHT DCT]	[DHT DCT] dictionary learned from unstable datasets
Unstable [DWT DCT]	[DWT DCT] dictionary learned from unstable datasets
Unstable [DHT DWT]	[DHT DWT] dictionary learned from unstable datasets
Unstable [DST DHT DCT]	[DST DHT DCT] dictionary learned from unstable datasets
Unstable [ID DCT DST]	[ID DCT DST] dictionary learned from unstable datasets
Unstable [DWT DHT DCT]	[DWT DHT DCT] dictionary learned from unstable datasets

Liste des symboles

x_n, \dot{x}	Variable d'état et variable d'état dérivative
(x_0, u_0)	Point d'état d'équilibre et point entrée
y, u_r	Vecteurs d'entrée et de sortie
γ, t	Cut-off stability value and time
$\overline{\Delta\delta}_i, \psi_i$	Déviations maximum et indice de stabilité
$\Delta u_i, \Delta x_i$	Petite déviation des variables d'entrée et des variables d'état
δ_i, W_{di}	Angle de rotor et taux de changement d'énergie transitoire (RoCoTE)
$\lambda_i(x)$	Exposants de Lyapunov
$\Lambda(x)$	Matrice de coefficient
T'_{d0i}, T'_{q0i}	Constante de temps transitoire en circuit ouvert d & q
Ω^c, N_d	Ensemble des perturbations et nombre de base de données
f_i, g_i	Système d'état et fonction de sortie
N_l, N_c, N_g	Nombre de ligne, de contingences et de générateurs
N_f	Nombre des attributs de HDSI pour le data mining
x_d, x_q	Réactance direct (d) & réactance quadrature (q)
$PC-1, PC-2$	Première et seconde composante principale
e'_d, e'_q	Tension transitoire électromotrice dans l'axe d & q
Ψ, ω	Transformée en ondelette et pulsation
x'_d, x'_q	Reactance transitoire dans l'axe d & q
x_w, R	Segment d'échantillon et taille de fenêtre des méthodes de périodogramme et de Welch
m, k	Nombre de block et segment utilisés pour les méthodes de périodogramme et de Welch

Je dédicace cette thèse doctorale à toute ma famille.

Remerciements

Par ces mots, je marque mon estime à mon directeur de thèse, le Professeur Innocent Kamwa, pour ses orientations, sa rigueur, sa vision et sa disponibilité avant et durant cette étude doctorale.

Je voudrais, remercier le Professeur Hoang Le-Huy, pour sa disponibilité, ses observations et pour l'éclairage permanent dans le domaine des réseaux électriques.

Ma gratitude va également à l'endroit du Professeur Jérôme Cros pour disponibilité et pour m'avoir permis de bien comprendre les relais de protection dans les réseaux électriques.

Mon estime va à l'endroit du Docteur. C. F. Mugombozi de l'Institut de recherche d'Hydro-Québec pour disponibilité et pour m'avoir permis de bien comprendre la modélisation sur Matlab/SimPowerSystems.

Ma pensée va également au Professeur C. Y. Chung de l'Université de Saskatchewan pour sa collaboration scientifique et ces conseils dans la rédaction et publication des articles.

Mon attention va également au Professeur Jacques Tagoudjeu de l'École Nationale Supérieure Polytechnique de Yaoundé pour sa collaboration scientifique et ces conseils dans la rédaction et publication des articles.

Une reconnaissance est adressée au Professeur Jean Kamdem, coordonnateur du pôle de recherche en mécatronique du CETIC de Yaoundé pour ses conseils scientifiques, sa disponibilité et sa présence durant toutes ces années d'études.

Ma considération va à l'endroit de Madame la Ministre de la Recherche scientifique et de l'innovation, la Dre Madeleine Tchuinté pour son écoute et ses conseils scientifiques réguliers et permanents.

Liste des publications

- [1] **T. D. Raoult**, I. Kamwa, T. Jacques and C. F. Mugombozi, "Supervised Learning of Overcomplete Dictionaries for Rapid Response-based Dynamic Stability Prediction," in *IEEE Transactions on Power Systems*, doi: 10.1109/TPWRS.2022.3156025.
- [2] **R. T. Dabou**, I. Kamwa, C. Y. Chung and C. F. Mugombozi, "Time Series-Analysis Based Engineering of High-Dimensional Wide-Area Stability Indices for Machine Learning," in *IEEE Access*, vol. 9, pp. 104927-104939, 2021.
- [3] **Dabou, Raoult T.**, Innocent Kamwa, Jacques Tagoudjeu, and Francis C. Mugombozi, "Sparse Signal Reconstruction on Fixed and Adaptive Supervised Dictionary Learning for Transient Stability Assessment," *Energies* 14, no. 23: 7995, 2021.
- [4] **R. T. Dabou** and I. Kamwa, "Rapid Design Method for Generating Power System Stability Databases in SPS for Machine Learning," *2020 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, pp. 1-6, 2020.

Introduction Générale

En raison de la lutte plus agressive contre les changements climatiques [1], on assiste à un essor fulgurant des ressources énergétiques à production intermittentes qui conduira à un réseau dominé par l'électronique de puissance. Cette mutation forcera du coup les services publics à s'intéresser à de nouvelles méthodes de détection de l'instabilité, centrées sur l'intelligence des données par opposition aux approches conventionnelles basées sur la reconnaissance des changements topologiques par des relais de protection et autres systèmes de logique préprogrammés dans les automatismes de réseaux. Ces nouvelles stratégies favorisant la prise de décision en temps réel, en s'appuyant directement sur les réponses mesurées par les PMU, vont faciliter un contrôle automatique qui s'avère de plus en plus nécessaire pour maximiser l'utilisation de l'électricité renouvelable (i.e. avec le moins de « curtailment » possible de production solaire/éolienne), sans nuire à la stabilité dynamique et la sécurité d'exploitation des réseaux [2]. D'autant plus que le déficit d'inertie des RED non seulement stimulera de grandes variations de fréquence, mais limitera la capacité du réseau à restaurer un fonctionnement normal à la suite des chocs de fréquence induits par les déséquilibres inopinés entre l'offre et la demande. Les solutions à ces problèmes de stabilité dynamique incluent certes des méthodologies basées sur le « matériel » permettant aux convertisseurs électroniques de puissance de former un réseau (plutôt que de le suivre), et des techniques d'émulation d'inertie qui transforment les RED en générateurs synchrones virtuels, capables de moduler leur production en fonction des variations de fréquence. Pour autant, on voit poindre les limites de ces solutions lorsque le réseau est surchargé de production renouvelable propulsée par l'électronique de puissance, incluant des boucles de verrouillage de phase susceptible d'induire des oscillations forcées, qui sont de nouvelles nuisances sur le réseau et donc impossible à anticiper avec les paradigmes conventionnels centrés sur la connaissance a priori de la topologie du réseau avant et après la contingence.

Dans ce contexte, on assiste dans la littérature à un renouveau marqué de la recherche sur l'évaluation de la stabilité dynamique des réseaux afin de développer une nouvelle génération d'automatismes inspirés par les données provenant de capteurs intelligents

(PMU) placés aux bornes de tous les générateurs du réseau [3-4]. Une telle approche, profitant des progrès rapides en matière d'estimation dynamique d'état [5] dans des réseaux dominés par l'électronique de puissance est capable de proposer des solutions qui s'adaptent en continu au contexte réel d'exploitation du réseau au lieu d'être prisonnière d'hypothèses fixées a priori alors que les conditions réelles du réseau changent au gré d'une production d'origine renouvelable dont la pénétration atteint aujourd'hui 80% en Australie méridionale à certaines heures du jour [6]. Ces changements, s'ils ne sont pas bien maîtrisés en planification ni en exploitation, créeront des conditions très propices aux instabilités de réseau, d'où l'intérêt soutenu pour ce sujet.

0.1 Contexte de recherche

Les perturbations de réseaux affectent fortement les industries provoquant des pertes estimées annuellement entre 104 et 164 milliards de dollars [7]. La réduction de cette perte économique nécessite une bonne gestion, analyse et prévision des conditions du système en temps réel [8]. La méthode la plus précise pour l'analyse de la stabilité transitoire est sans conteste la simulation dans le domaine temporel ; cependant, cette approche prend du temps, en particulier pour les systèmes électriques à grande échelle. Par ailleurs, même si la capacité de calcul était suffisante, une telle simulation est tributaire de la disponibilité de données précises sur le modèle du réseau et de la configuration topologique à chaque instant pendant la simulation. Pour ainsi dire, l'approche par simulation est difficile à envisager lorsque la décision doit être prise rapidement après un défaut afin d'activer un automatisme. Il faut impérativement recourir à une intelligence des données tests si on veut déployer des automatismes auto-adaptatifs en fonction des signaux de réponses enregistrés.

Grâce à la disponibilité croissante de données massives et synchronisées temporellement dans les systèmes électriques modernes, on s'attend à ce que les solutions centrées sur les données jouent un rôle crucial dans l'exploitation et le contrôle des systèmes électriques futurs [9-10]. L'intelligence des données, qui offre une capacité de modélisation puissante et flexible, ainsi qu'une capacité de prédiction rapide après l'obtention des attributs d'entrée, est aujourd'hui considérée comme une alternative crédible pour détecter l'instabilité dynamique en temps réel, par apprentissage automatique

des données de PMU générées via des simulations probabilistes de comportements dynamiques.

Les techniques d'extraction d'attributs et d'apprentissage automatique ont été proposées dans de nombreuses publications, en vue de classer les perturbations transitoires associées à la qualité de l'onde, détecter les vulnérabilités en sécurité de réseau, ainsi que les menaces et intrusions dans les données d'exploitation du réseau hébergées par les SCADA et prédire l'instabilité dynamique des systèmes électriques interconnectés [11-12]. L'exposant maximum de Lyapunov est implémenté dans [13-14] pour prédire l'instabilité de l'angle du rotor en temps réel à partir des données de PMU. Les auteurs soutiennent que, pendant le fonctionnement, les générateurs restent synchrones si le système linéarisé est asymptotiquement stable. Dans [15], des règles de classification pour l'évaluation rapide de la stabilité post-contingence sont développées à partir d'approches d'arbre de décision et de logique floue. Ces auteurs extraient et classent un petit ensemble d'attributs de stabilité appelés indices de sévérité, à partir de la base de données de monitoring du réseau d'Hydro-Québec après l'élimination de défauts de sévérité modéré à grave, en considérant des milliers de conditions d'exploitation (en termes de topologies et d'équilibre production-demande). Dans [16], l'accent est mis sur l'importance de l'analyse spectrale de la distribution temps-fréquence des oscillations d'énergie électromécanique du système électrique après une perturbation. La transformée de Fourier à court terme, la logique floue et les approches de réseaux de neurones (NN) sont utilisées dans [17] pour extraire et classer les attributs prédictifs de stabilité. Dans [18], un algorithme basé sur la transformée de Fourier discrète et l'interpolation est proposé pour identifier et reconfigurer la dynamique des générateurs à production variable grâce à la reconnaissance de formes d'indices de similarité de spectral.

L'arbre de décision appliqué au signal d'intégrale de l'angle de rotor de générateur au carré ou de l'angle de phaseur de tension de barre au carré, est utilisé dans [19] pour détecter les événements et prédire la perte de synchronisme afin de permettre des contre-mesures ponctuelles et bien minutées. L'article [20] effectue une prédiction de la stabilité de la tension dans les réseaux intelligents en utilisant une forêt aléatoire (RF), un NN et une machine à vecteurs de support (SVM) entraînés sous supervision. Alternativement, les

auteurs de [21] ont utilisé un ensemble de cinq fonctionnalités pour instruire des techniques d'arbre de décision afin de prédire la stabilité transitoire des générateurs synchrones dans divers scénarios d'exploitation conduisant à des pannes. Dans la référence [22], les caractéristiques de l'angle mécanique du rotor sont utilisées comme entrées d'un classificateur à apprentissage profond du type LSTM pour prédire l'instabilité transitoire des réseaux électriques. Les ELM sont utilisés dans [23], pour classer et prédire l'état de sécurité dynamique en temps réel alors que les auteurs de [24] proposent un algorithme de clustering C-means flou pour prédire l'état de stabilité de l'angle du rotor après une perturbation.

Selon l'état de l'art ci-dessus, la détection d'instabilité des réseaux électriques basée sur les attributs des signaux de séries chronologiques enregistrées après l'élimination des défauts a été largement étudiée, mais en utilisant un petit ensemble d'attributs possibles allant de 5 à 20 par générateur ou zone de contrôle. Une telle approche d'ingénierie des attributs, bien qu'adaptée au réseau spécifiquement à l'étude, nous apparaît difficile à généraliser sur de nouveaux réseaux ou de nouvelles configurations du réseau existant, car certains des attributs proposés sont fonction de l'inertie des machines et de la topologie spécifique du réseau [25]. Par ailleurs, on observe une tendance récente vers la classification de telles séries par des techniques d'apprentissage profond sans utilisation d'attributs interprétables. Bien que les résultats soient parfois remarquables en termes de temps prédiction et précision/fiabilité [26], il n'en demeure pas moins vrai que le côté opaque de telles solutions rebute les ingénieurs d'exploitation qui ont la mission critique de protéger les réseaux à tout prix contre les pannes majeures.

Pour surmonter ces limitations, plusieurs chercheurs ont converti les données de séries chronologiques en attributs explicables, en utilisant par exemple une analyse spectrale, des fonctions d'énergie ou des statistiques descriptives [27]. Cependant, le petit nombre de caractéristiques considérées a empêché ces approches de tirer le plein potentiel de la grande masse de données disponible. Avec une nouvelle perspective inspirée par la méthodologie d'analyse massive comparative des séries chronologiques (HCTSA), la porte est désormais grande ouverte pour étendre le nombre d'attributs candidats à des centaines, voire des milliers d'attributs à signification physique décrivant la série chronologique de manière

concise. Alors que les formes d'ondes brutes ne sont pas explicables, les statistiques ou les spectres de séries chronologiques post-contingence sont des caractéristiques intrinsèques qui peuvent être expliquées et exploitées par les ingénieurs pour qualifier la sévérité des instabilités de réseaux.

Une autre réponse possible aux enjeux précédents consiste à explorer une approche d'apprentissage sur des dictionnaires surcomplets [28], approche dont le potentiel a été démontré en classification offline des signaux de qualité d'onde [29], mais qui n'a jamais été employée jusqu'ici pour détecter en temps réel l'instabilité dynamique des réseaux. Au lieu de s'en remettre à des centaines d'attributs pour prédire l'instabilité (approche à cet égard non parcimonieuse bien qu'interprétable), on opte ainsi pour un dictionnaire surcomplet appris en temps différé de manière supervisée, sur lequel on décompose en temps réel les signaux de réponse post-contingence en minimisant le nombre de coefficients et d'atomes requis pour assurer une précision spécifiée (*sparse decomposition*). Il s'agit donc d'une approche à la fois parcimonieuse et capable d'interprétabilité car basée sur des dictionnaires ayant un nombre fini d'atomes, fixes ou déterminés automatiquement à partir d'une base de données représentative des phénomènes de stabilité sur un réseau donné.

La technique de classification sur un dictionnaire appris vise à trouver, moyennant un critère d'arrêt, le meilleur compromis existant entre une erreur de reconstruction et le nombre d'atomes utiles pour la représentation optimale du signal. La décomposition parcimonieuse de la base de données d'entraînement est utilisée pour reconstruire et prédire la stabilité des réseaux électriques face aux perturbations. Par analogie à la matière, un atome de stabilité est une petite partie du signal de même nature que la réponse originale. La représentation parcimonieuse permet d'exprimer les signaux $Y \in \mathbb{R}^{n \times N}$ sous la forme d'un dictionnaire d'atome $D \in \mathbb{R}^{n \times K}$ et d'un support $X \in \mathbb{R}^{K \times N}$. Le processus de décomposition parcimonieuse d'une base de données $Y \in \mathbb{R}^{n \times N}$, sur un dictionnaire d'apprentissage surcomplet $D \in \mathbb{R}^{n \times K}$ (c'est-à-dire: $K \succ n$) est présenté à la Figure 01, où n et K : représentent respectivement le nombre d'échantillons du signal et le nombre de colonnes du dictionnaire.

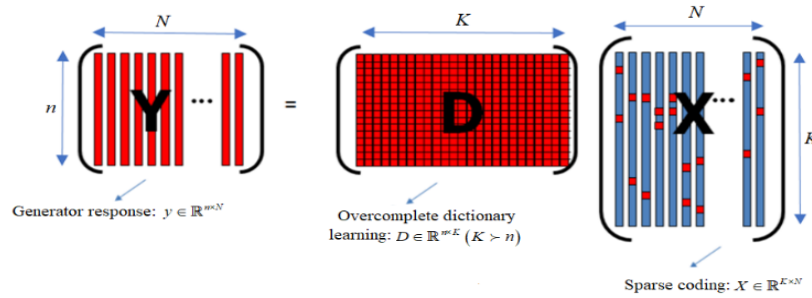


Figure 01: Décomposition parcimonieuse des signaux de stabilité sur un dictionnaire [28].

La décomposition de $Y \in \mathbb{R}^{n \times N}$ est considérée comme un changement vers un nouvel espace (ou base) parcimonieux, redondant, transparent et facilement exploitable. Pour une base de données de stabilité, il existe trois représentations possibles du jeu d'atome:

- 1) si $n = K$, alors on a un dictionnaire qui effectue une bonne reconstruction des nouveaux scénarios, et l'ensemble des atomes forme une famille libre (chaque atome apporte une contribution originale, non présente dans les autres atomes). Le dictionnaire peut être assimilé à une base de l'espace des signaux;
- 2) si $n < K$, alors on a un dictionnaire qui n'effectue pas une bonne reconstruction des scénarios d'apprentissage, et les signaux ne peuvent pas être décomposés correctement. Car tous les atomes du dictionnaire ne sont pas utilisés;
- 3) si $K > n$, alors on a une infinité de dictionnaires redondants permettant de décomposer les signaux de manière correcte. Chaque sous-ensemble du dictionnaire atome forme une base de décomposition parcimonieuse. On parle alors d'un dictionnaire surcomplet.

Où : n et K : représentent respectivement le nombre de lignes de la base de données et le nombre de colonnes du dictionnaire d'apprentissage. La transformation linéaire surcomplète décrite à la Figure 01 offre une infinité de solutions mathématiques, difficile à interpréter sur le plan physique. Le problème de la décomposition parcimonieuse surcomplète est de trouver le meilleur ensemble d'atomes et la façon de les combiner afin d'obtenir une erreur de reconstruction et de classification acceptable [30]. Le critère de choix du jeu d'atomes que nous avons sélectionné dans cette thèse est la parcimonie, car

elle permet de mettre en exergue la décomposition qui utilise le moins d'atomes possible. Il existe dans la littérature plusieurs approches d'apprentissage supervisé des dictionnaires regroupés en : 1) méthodes fixes et 2) méthodes adaptatives. Les méthodes fixes (DCT, DST, DHT, DWT, ID, etc.) sont développées en utilisant les noyaux mathématiques prédéfinis et physiquement interprétables. Toutefois, les méthodes adaptatives ou *data-driven* (K-SVD, MOD, RLS-DLA, etc.) ne s'appuient pas sur des formes de noyaux spécifiques, mais uniquement sur la structuration de la base de données [31]. L'ensemble de ces méthodes ont connu du succès en traitement d'image, dans la détection et la classification des PQs (*sag*, *swell*, *harmonic*, etc.) [32], dans la reconnaissance et la classification des défauts des machines [33-34], mais jamais dans l'analyse prédictive des signaux de stabilité post-contingence.

L'apprentissage par des dictionnaires d'atomes fixes consiste à appliquer sur une base de données, une ou plusieurs fonctions prédéfinies fixes (DCT, DST, DHT, DWT, ID, etc.) pour extraire les indicateurs parcimonieux capables de prédire dans un délai raisonnable la stabilité des réseaux électriques. Ces atomes (simple ou hybride) permettent de caractériser les structures transitoires et oscillatoires dormantes dans cette base de données. Fort de cela, trois structurations des dictionnaires d'atomes fixes sont implémentées sur MATLAB, il s'agit de: 1) dictionnaires simples (DCT, DHT et DWT), 2) dictionnaires hybrides à deux concaténations ([DHT|DCT], [DWT|DCT] et [DHT|DWT]) et 3) dictionnaires hybrides à trois concaténations ([DST|DHT|DCT], [ID|DCT|DST] et [DWT|DHT|DCT]). Le développement des représentations hybrides est exploré pour évaluer l'impact de la concaténation des dictionnaires simples sur l'extraction parcimonieuse des attributs et sur la prédiction de la stabilité [35].

Contrairement à l'approche fixe, la méthode *data-driven* est utilisée pour extraire les indicateurs d'instabilités présents dans la base de données des signaux non stationnaires de stabilité post-contingence. Cette méthode est principalement guidée par la structure et la nature intrinsèque des données. La décomposition en valeurs singulières (K-SVD) couplée à l'*orthogonal matching pursuit* (OMP) est l'approche *data-driven* la plus populaire de représentation parcimonieuse des signaux [36]. Cette décomposition présente l'avantage qu'elle n'est pas basée sur un noyau d'atome prédéfini. La procédure de recherche des attributs parcimonieux de cette approche s'effectue échantillon par échantillon jusqu'à

obtention de la plus petite erreur possible de reconstruction telle que $\xi \approx Y - DX$ (D'après la Figure 01). La décomposition en valeurs singulières permet de représenter la base de données $Y = U\Sigma V^* \in \mathbb{R}^{n \times N}$ sous la forme de trois matrices à savoir : 1) une matrice inversible de vecteurs propres de gauche $V \in \mathbb{R}^{N \times N}$, 2) une matrice inversible de vecteurs propres de droite $U \in \mathbb{R}^{n \times n}$ et 3) une matrice diagonale de valeurs propres $\Sigma \in \mathbb{R}^{n \times N}$ [37].

0.2 Problème de recherche

0.2.1 Problème général

Le contexte de recherche décrit à la section 0.1 permet d'identifier trois enjeux majeurs dans la détection d'instabilité dynamique basée sur l'analyse en temps réel des réponses post-contingences mesurées par des PMUs.

1) Approche par extraction d'attributs suivi d'un apprentissage machine: l'utilisation d'un nombre restreint de caractéristiques des signaux d'entrées aux algorithmes de classification ne favorise pas l'exploitation du plein potentiel de la grande quantité de données disponibles. En effet, les systèmes de surveillance du réseau sont capables d'enregistrer par des PMU les tensions et courants de chaque alternateur, données ensuite digérées par des estimateurs d'état dynamique pour reconstruire jusqu'à 10 signaux de réponses post-contingences par alternateur. Il est donc très réducteur de n'utiliser qu'un attribut par machine comme c'est le cas dans de nombreuses références citées plus haut.

2) Approche par apprentissage profond de series chronologiques sans extraction préalable d'attributs : l'opacité de cette approche constitue sa principale limitation. Jusqu'ici, les chercheurs en réseaux électriques ont peu exploré l'interprétabilité de leurs modèles de classification par *deep-learning*, ce qui soulève des questions légitimes sur le risque de surspécialisation à des cas particuliers [38]. Dans le futur, il vaudra mieux maîtriser l'interprétation des décisions produites par les classificateurs à *deep-learning*, ou alors favoriser des méthodes d'apprentissage supervisée qui sont intrinsèquement moins opaques, par exemples celles basées sur des dictionnaires interprétables.

3) Temps de réponse minimum (ou temps de préemption maximum): l'intelligence des données en temps réel (« *streaming analytics* ») signifie que des données sont

disponibles en quantité et qualité au moment de prendre la décision. Or ceci implique un nécessaire délai entre la collecte des données et la prise de décision, contrairement aux approches basées sur la détection des changements topologiques qui réagissent presque instantanément. Toutes les méthodes de détection d'instabilité basées sur l'analyse des réponses post-contingence font face à ce défi et les meilleures méthodes à cet égard sont celles capables de détecter l'instabilité à partir de la plus courte fenêtre de données possible. De plus, ces classificateurs doivent minimiser le temps de calcul en playback, car ce temps CPU s'ajoute au temps de collecte des données pour déterminer le temps de réponse qui doit être assez réduit afin de laisser un temps maximum pour déployer les mesures curatives.

0.2.2 Problèmes spécifiques

Le nombre limité d'indicateurs significatifs non parcimonieux extraits des bases de données ne favorise pas l'intégration de l'intelligence artificielle à l'étude de la stabilité des réseaux électriques, car les moteurs d'apprentissage de stabilité nécessitent idéalement un volume important d'attributs pour maximiser leur performance en termes de précision et fiabilité. Or à date, l'analyse prédictive de la stabilité ne s'effectue qu'à partir d'une dizaine d'attributs [39]. Ceci à cause de l'utilisation des méthodes non combinées d'extraction d'indicateurs d'instabilité (ou de sévérité des catastrophes) qui considèrent de manière cloisonnée les différents domaines physiques : temps versus fréquence, statistiques descriptives versus fonctions d'énergie, etc., au grès des choix arbitraires des auteurs concernés. Pourtant, les performances de prédiction dépendent aussi bien des algorithmes de prédiction que du volume et de la qualité des attributs d'apprentissage physiquement interprétables. L'approche d'extraction d'indices de stabilité à haute dimension appelée HDSI (non parcimonieux) basée sur une méthode combinée (plusieurs domaines d'interprétation physique considérés simultanément via HCTSA) améliorera la précision, la fiabilité et la sécurité des algorithmes de machine learning.

Les outils intelligents de prédiction par apprentissage machine (SVM, DT, RF, ANN, CNN et LSTM) sont à la base des boîtes noires difficilement interprétable sur le plan physique à moins au minimum d'utiliser des attributs qui le sont. Dans ce contexte, les structures de classification par dictionnaires adaptatifs ou fixes se positionnent comme une

alternative intéressante pour une analyse parcimonieuse, néanmoins interprétable et potentiellement plus efficace au niveau du temps de réponse, élément qui reste le principal goulot d'étranglement des méthodes de détection d'instabilité basées sur les réponses post-contingence du réseau comparativement aux approches par détection des changements topologiques. L'utilisation des dictionnaires appris sous supervision passe par l'intégration d'un terme discriminant dans la base de données d'entraînement responsable de classification des états de stabilité stables/instables dans le dictionnaire adaptatif. Cette nouvelle approche *data-driven* permettra de reconstruire et de prédire les réponses des générateurs immédiatement après l'élimination des défauts [40]. De plus, le développement des structures de dictionnaire fixes pour chaque état de stabilité considéré séparément (stable ou instable) et considérant des prédicteurs différents pour chaque classe, facilitera l'extraction de la base de données d'entraînement d'atomes permettant la reconstruction parcimonieuse des signaux de réponse du réseau. Ces atomes surcomplets (stable ou instable), physiquement interprétables, présentent l'avantage de pouvoir définir et dans un délai raisonnable, le degré d'appartenance du signal de réponse à chaque classe stable/instable. Considérant le problème général, les problèmes spécifiques de cette recherche doctorale sont présentés comme suit :

Problème 1 – Adressé au Chapitre 2 – La classification des séries temporelles basée sur des attributs transparents (non parcimonieux) est limitée à un petit nombre d'attributs (généralement de 1 à 10 selon les auteurs), avec une difficulté notable à généraliser sur des attributs ainsi choisis à de nouveaux réseaux électriques et de nouvelles configurations. Ce faible nombre d'indicateurs de stabilité extraits des réponses des générateurs freine l'amélioration des performances de prédiction (précision, fiabilité, et sécurité) des algorithmes de *machine learning* et le chapitre 3 permet de combler cet écart.

Problème 2 – Adressé au Chapitre 3 – Chaque échantillon des signaux de stabilité post-contingence contient une information dormante précurseur d'instabilité des réseaux électriques. La mise en exergue (prétraitement, labélisation, extraction etc.) de cette information nécessite une bonne expertise en traitement du signal et en opération des réseaux électriques. La plupart des méthodes d'extraction des attributs (parcimonieux) existantes n'intègre pas dans leurs processus d'apprentissage, l'état de stabilité

stable/instable des signaux. L'inexistence d'un dictionnaire joint au classificateur (qui ne nécessite aucune méthode préliminaire d'extraction des attributs), appris de la base de données d'entraînement, limite considérablement la précision de reconstruction et de classification de tout nouveau scénario et le chapitre 3 permet de combler cet écart.

Problème 3 – Adressé au Chapitre 4 – Les bases de données mixtes, souvent asymétriques (c'est-à-dire constituée d'un mixte de scénarios stables et instables, avec une forte majorité de cas stables) et permettant d'extraire des attributs parcimonieux de stabilité de la base de données d'entraînement, ont un impact sur la définition des degrés d'appartenances (stable/instable) des nouveaux scénarios. Par hypothèse, les dictionnaires parcimonieux stables (ou instable) minimisent considérablement l'erreur de reconstruction et le temps réponse des prédicteurs des signaux stables (ou instables). L'inexistence d'une méthodologie de construction de bases séparées de projection des signaux stable/instable rend impossible la définition du degré de stabilité des nouveaux scénarios et le chapitre 4 permet de combler ce gap.

0.3 Objectifs de recherche

A la lumière de la littérature, la décomposition parcimonieuse des signaux électriques s'est concentrée jusqu'à présent sur l'extraction, la classification des *power quality disturbances* ou *events* (PQs), la reconnaissance des défauts dans les machines et dans les réseaux électriques, mais n'a jamais été considérée dans l'analyse prédictive des signaux de stabilité post-contingence. La décomposition supervisée des signaux de stabilité sur des dictionnaires d'atomes surcomplets vise à extraire de la base de données d'apprentissage, la meilleure structure capable de reconstruire et de prédire tout nouveau scénario. Plusieurs approches prédictives de machine learning ont largement étudié ce problème (en se basant sur un petit nombre d'attributs extraits) d'analyse de la stabilité, mais aucune approche intelligente utilisant l'analyse massive des séries chronologiques pour déterminer des centaines d'attributs interprétables n'a été considérée jusqu'ici. En réponse à ces insuffisances, cette thèse se propose d'implémenter à partir de la base de données d'entraînement, un dictionnaire de stabilité surcomplet (joint et séparé du classificateur) capable de reconstruire et de prédire dans un délai raisonnable, tout nouveau scénario. Les

performances de ces dictionnaires seront comparées à celles des algorithmes classiques de machine learning implémentés à partir des indices de stabilité à haute dimension qui bénéficient ainsi d'une grande quantité d'attributs interprétables physiquement pour maximiser leur précision et leur fiabilité. De manière spécifique, il s'agira de :

- ❖ **Objectif 1) – Adressé au chapitre 2** – Proposer une méthode combinée d'extraction (non parcimonieuse mais interprétable) des attributs de stabilité à haute dimension basée sur les exposants de Lyapunov, la transformée de Fourier, la transformée en ondelette, la méthode de Welch et la méthode de périodogramme. Ces indicateurs d'instabilités seront utilisés pour développer les méthodes prédictives de machine learning (SVM, RF, DT, ANN, AdaBoost).
- ❖ **Objectif 2) – Adressé au chapitre 3** – D'apprendre conjointement de la base de données d'entraînement un dictionnaire (adaptatif ou *data-driven*) joint au classificateur à partir de l'approche K-SVD couplé à l'OMP. L'intégration d'un terme discriminatif des classes stables/instables introduite au problème d'optimisation des réponses des générateurs constitue l'innovation principale de cette analyse parcimonieuse. Ce dictionnaire discriminatif appris de la base de données d'entraînement sera utilisé pour reconstruire et prédire les réponses des générateurs tout en détectant l'instabilité dans un temps réponse plus court que celui du chapitre 2. Cette approche par dictionnaire surcomplet sera confrontée au *deep-learning* basé les moteurs modernes tels que le CNN et LSTM, en utilisant les attributs HDSI comme intrants.
- ❖ **Objectif 3) – Adressé au chapitre 4** – D'apprendre séparément des dictionnaires (simple et hybride) et des prédicteurs à partir des approches (fixe et adaptatif) d'extraction des attributs suivant : DCT, DST, DWT, DHT, [DHT|DWT]; [DWT|DCT]; [DHT|DCT], [DST|DHT|DCT]; [ID|DCT|DST], [DWT|DHT|DCT] et K-SVD. Chacun de ces vingt dictionnaires sera utilisé pour définir (en utilisant l'indice RMSE) le degré d'appartenance et classer chaque nouveau scénario d'apprentissage. Les dictionnaires étant fixes, le potentiel d'interprétabilité est plus grand, de même que la capacité à généraliser à de nombreux autres réseaux électriques.

0.4 Contribution originale

L'analyse massive des séries chronologiques (HCTSA) est appliquée à la détermination de centaines d'attributs interprétables pour fins d'amélioration de la précision et de la fiabilité de la détection d'instabilité par apprentissage machine de réponses post-contingence des réseaux électriques mesurées par des PMU et prédigérées par des estimateurs d'état dynamique. De plus, aucune publication dans la littérature ne traite de la décomposition parcimonieuse des signaux de stabilité post-contingence sur des dictionnaires appris, pour la reconstruction et la prédiction des signaux de réponses et l'état stable/instable des générateurs. Les principales contributions de cette thèse sont résumées comme suit :

- 1) Proposer une méthode nommée *high dimensional stability indices* (HDSI) permettant d'extraire pour chaque signal de réponse de générateur, mesuré directement ou obtenu par DSE, plus de 500 indicateurs de stabilité physiquement interprétable. Trente-une fonctions d'extraction combinées à savoir : l'analyse spectrale basée sur les méthodes de périodogramme et de Welch, les exposants de Lyapunov et les approches de transformation en ondelettes ont permis d'augmenter considérablement le volume d'attributs comparativement à l'état l'art, et d'ainsi améliorer les performances de prédiction des algorithmes de machine learning (SVM, RF, DT, ANN et AdaBoost). Il ressort de cette étude que les indices HDSI, extraits de la base de données des vitesses des rotors performant mieux que ceux implémentés à partir de l'indice *rate of change of transient energy* (RoCoTE) pourtant populaire auprès de certains auteurs. (Objectif 1)
- 2) Développer et implémenter une approche jointe de reconstruction et classification de signaux de stabilité nommée *supervised learning of overcomplete dictionary* (SLOD) utilisant la K-SVD pour apprendre des structures adaptatives à partir des signaux des vitesses des rotors labellisés. Un terme discriminatif, responsable de la classification des instabilités a été introduit dans le processus de décomposition parcimonieuse des vitesses des rotors. Ce dictionnaire surcomplet est capable de reconstruire et prédire conjointement les réponses des générateurs face aux contingences. Comparativement à l'approche HDSI basée sur les algorithmes de machine learning par CNN ou LSTM, la

SLOD a donné des performances presque parfaites (c'est-à-dire : ~99,99 % de précision, ~99,99 % de fiabilité et 100 % de sécurité) sur les données test des réseaux IEEE 2 area 4 machines, IEEE 39 -bus et IEEE 68 -bus. (Objectif 2)

- 3) Développer plusieurs approches (ADL, SDL, 2H-OHD et 3H-OHD) d'apprentissage séparé de cas stable/instable, basées respectivement sur la K-SVD, la DCT, la DWT, la DHT, la [DHT|DCT], la [DWT|DCT], la [DHT|DWT], la [DST|DHT|DCT], la [ID|DCT|DST] et la [DWT|DHT|DCT]. L'ensemble de ces méthodes ont permis de développer vingt dictionnaires (simple et hybride) stable et instable. Chaque dictionnaire de stabilité surcomplet est appliqué sur la base de données test des réseaux IEEE 2-area 4-machines, IEEE 39 -bus et IEEE 68 -bus pour définir le degré d'appartenance des signaux à partir de l'indice RMSE. La présente analyse a démontré que [DWT], [DHT|DWT] et [DST|DHT|DCT] sont de meilleurs descripteurs des cas stables/instables par rapport à K-SVD, [DHT], [DCT], [DCT|DWT], [DHT|DCT], [ID|DCT|DST] et [DWT|DHT|DCT]. Cependant, l'approche K-SVD est plus rapide à exécuter à la fois lors de l'entraînement et pendant le replay en temps réel, tout en offrant une précision satisfaisante dans la prédiction de la stabilité transitoire (c'est-à-dire : fenêtre de décision de 7,5 cycles après élimination des défauts). (Objectif 3)

La Figure 02 montre le diagramme détaillé des différentes étapes constituant notre méthodologie de la recherche.

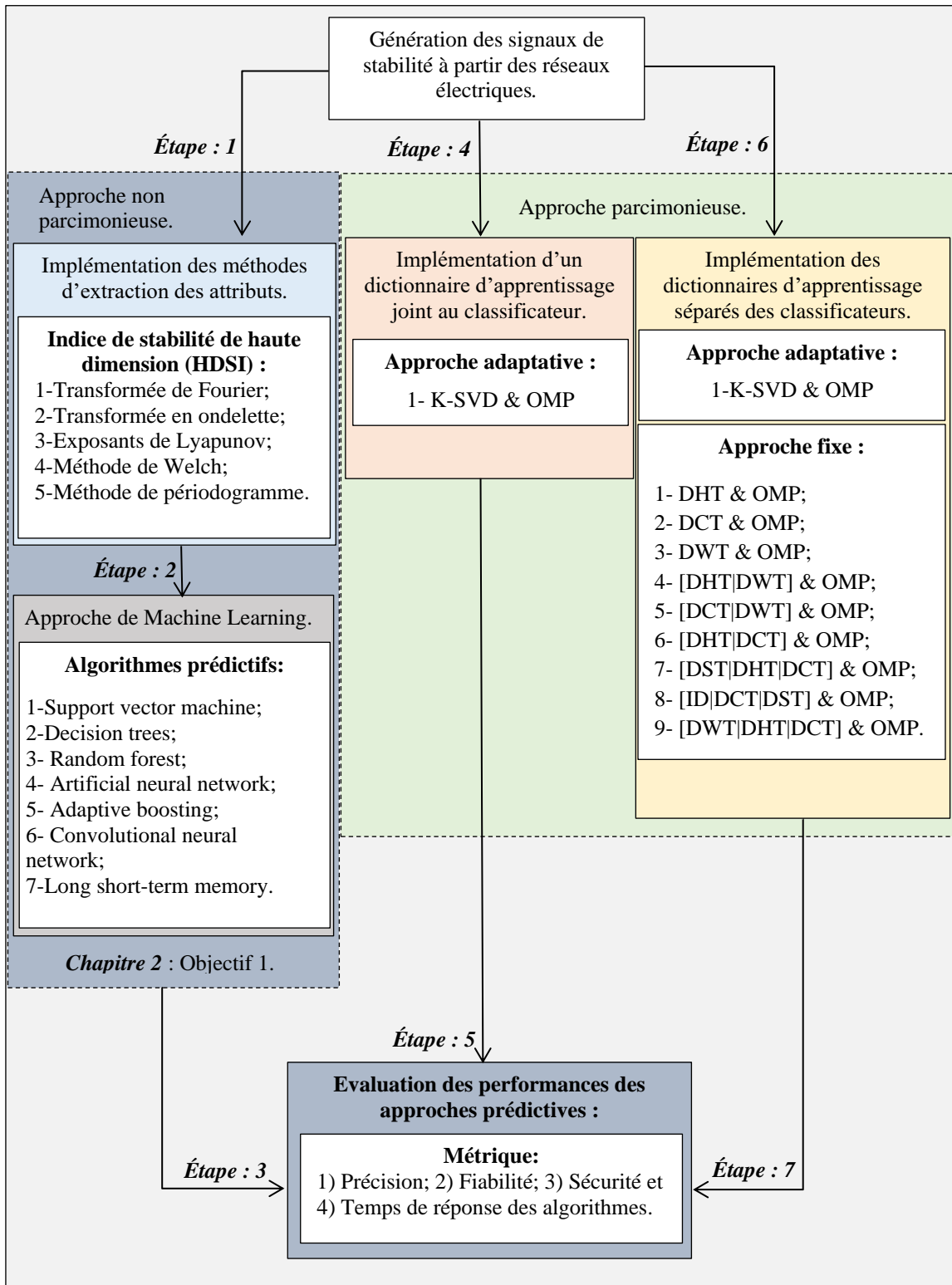


Figure 02 : Diagramme détaillé des différentes étapes constituant notre méthodologie de la recherche.

0.4 Organisation de la thèse

Cette thèse est organisée comme suit :

Chapitre — 1 présente l'état de la recherche en lien avec la génération massive et simple des signaux de stabilité, l'extraction des attributs et le développement des dictionnaires d'apprentissage parcimonieux. La section 1.2 met en exergue les difficultés d'accès (pour des besoins académiques) aux bases de données des signaux de stabilité suffisamment représentative de plusieurs scénarios (plus d'une centaine). La section 1.3 permet d'illustrer les méthodes conventionnelles d'extraction des attributs utiles pour le développement des outils de prédiction intelligents. La section 1.4 présente une revue de littérature sur les dictionnaires fixes et adaptatifs développés pour extraire et classer les indicateurs de stabilité parcimonieux et les *power quality events (PQs)*. Une conclusion partielle de la revue bibliographique sera présentée à la section 1.5.

Chapitre — 2 présente une méthode d'extraction et de classification des attributs labellisés dérivés des signaux des vitesses des rotors et des taux de changement d'énergies transitoires (RoCoTE). La section 2.2 développe une approche de modélisation et de simulation des réseaux électriques sur Simscape Power System (SPS). Ces signaux sont générés en simulant en un clic, une centaine des défauts sur chaque ligne et barre génératrice des réseaux électriques. Dans la section 2.3, il sera question de présenter l'approche conventionnelle existante et de proposer une nouvelle méthode d'extraction des 500 attributs de stabilité à haute dimension. La section 2.4 met en exergue les performances de classification de cinq moteurs d'apprentissage machine basés sur l'indice HDSI. La section 2.5 se focalise sur les principales métriques, à savoir la précision, la fiabilité et la sécurité utilisées comme base de comparaison des différents moteurs d'apprentissage machine. Une conclusion partielle de l'approche HDSI pour la prédiction de la stabilité dynamique sera offerte à la section 2.6.

Chapitre — 3 présente le processus d'implémentation d'un dictionnaire adaptatif joint pour la reconstruction des réponses des générateurs et la classification de la stabilité des réseaux électriques. La section 3.2 illustre une méthode de recherche graduelle de la limite de stabilité appliquée de chaque ligne et barre génératrice des réseaux tests (IEEE 39 –bus,

IEEE 68 –bus et IEEE 2 area 4 machines). La section 3.3 présente les étapes à suivre pour la décomposition parcimonieuse des signaux de la base de données d’entraînement, la reconstruction et la classification de la stabilité des nouveaux scénarios. La section 3.4 présente une étude comparative des performances de classification de cinq algorithmes d’apprentissage (SVM, DT, RF, ANN, AdaBoost, CNN et LSTM) et du dictionnaire d’apprentissage supervisé (SLOD). Une conclusion partielle de la décomposition parcimonieuse des signaux de stabilité sur un dictionnaire d’atome joint sera offerte à la section 3.5.

Chapitre — 4 présente les différentes étapes à suivre pour le développement des dictionnaires séparés fixes/adaptatifs en vue de la reconstruction des signaux de réponses post-contingence et la classification de la stabilité des réseaux électriques. La section 4.2 illustre le processus de construction de vingt dictionnaires fixes/adaptatifs stables et instables utiles pour la description/classification du comportement des signaux de réponse des machines synchrones. La section 4.3 illustre la capacité de reconstruction et de classification de chaque dictionnaire stable et instable appris de la base de données d’entraînement sur deux bases de données tests. La section 4.4 présente l’approche d’évaluation du degré d’appartenance des signaux tests basés sur les dictionnaires fixes et adaptatifs appris. Une conclusion partielle des dictionnaires d’apprentissage séparé sera présentée à la section 4.5.

Chapitre — 5 présente une conclusion et des perspectives de recherche. Cette thèse présente une avancée considérable réalisée dans la prédiction de la stabilité des réseaux électriques basée sur les attributs à haute dimension physique interprétable et sur les dictionnaires parcimonieux fixes/adaptatif. Elle permet d’améliorer les performances de prédiction et de s’éloigner de l’approche d’interprétation boîte noire des méthodes d’apprentissage supervisés. En guise de perspective, nous prévoyons : 1) améliorer le temps d’apprentissage de la base de données d’entraînement de l’approche SLOD, 2) évaluer la stabilité dynamique des réseaux électriques par l’approche SLOD sur un simulateur en temps réel OPAL-RT e-megasim et 3) évaluer les performances de l’approche SLOD sur les données des PMUs bruités d’Hydro-Québec.

Annexes — présente la configuration du modèle Simulink du réseau IEEE 39 -bus et les codes Matlab d'extraction des attributs et l'apprentissage supervisé des outils intelligents. La section A.1 présente les données utilisées pour la modélisation et la simulation du réseau IEEE 39 –bus. La section A.2 présente le modèle unifilaire et Simulink du réseau IEEE 39 –bus avec des configurations des lignes et des barres génératrices. La section A.3 présente le code développé pour la génération simple et rapide des signaux de stabilité. La section A.4 présente le code utilisé pour l'extraction des attributs de stabilité à haute dimension (HDSI). La section A.5 présente le code source développé pour l'apprentissage du dictionnaire surcomplet joint au classificateur (SLOD). La section A.6 présente les codes sources développés pour l'apprentissage des dictionnaires surcomplets séparés des classificateurs (DCT, DHT, DWT, K-SVD, [DHT|DWT], [DCT|DWT], [DHT|DCT], [DST|DHT|DCT], [ID|DCT|DST], [DWT|DHT|DCT]). La section A.7 présente le code source développé pour la prédiction de la stabilité dynamique à partir de l'algorithme SVM. La section A.8 présente le code source développé pour la prédiction de la stabilité dynamique à partir de l'algorithme DT. La section A.9 présente le code source développé pour la prédiction de la stabilité dynamique à partir de l'algorithme RF. La section A.10 présente le code source développé pour la prédiction de la stabilité dynamique à partir de l'algorithme ANN. La section A.11 présente le code source développé pour la prédiction de la stabilité dynamique à partir de l'algorithme AdaBoost. La section A.12 présente le code source développé pour la prédiction de la stabilité dynamique à partir de l'algorithme CNN. La section A.13 présente le code source développé pour la prédiction de la stabilité dynamique à partir de l'algorithme LSTM.

Chapitre 1 — Revue Bibliographique

1.1 Introduction

Les réseaux électriques conçus aujourd’hui sont soumis à diverses perturbations internes ou externes susceptibles d’entraîner une dégradation progressive de sa stabilité ou une perte de synchronisme. Cette perturbation se trouve accentuée du fait de l’ouverture du marché de l’électricité, des changements climatiques, de l’intégration des énergies renouvelables, etc. Fort de cela, il devient important d’anticiper sur l’état d’instabilité dynamique dans un délai raisonnable. Car toute vulnérabilité tardivement identifiée d’un équipement interconnecté aux réseaux électriques peut avoir un impact négatif sur l’onde électrique fournie aux consommateurs et la continuité de service. Pour bien anticiper cette instabilité, plusieurs dispositifs intelligents (tels que PMUs, FDR, etc.) d’enregistrement et de transmission des données brutes en temps réel sont déployés dans les réseaux électriques. Ceux-ci génèrent en permanence une masse de données riche en information et annonciatrice de la perte de stabilité. Toutefois, l’interprétabilité rapide et efficace des informations dormantes enfouies dans les bases de données constitue un enjeu majeur pour les ingénieurs.

1.2 Base de données des signaux électriques

Le succès des algorithmes de *machine learning* pour la prédiction de la stabilité transitoire des réseaux électriques dépend en partie du volume des données d’apprentissage. Pourtant ces signaux de stabilité ne sont pas toujours disponibles pour des besoins académiques. Ces données réelles sont généralement la propriété des gouvernements, des centres ou instituts de recherche, etc [41]. Cette non-disponibilité des données freine l’application du *machine learning* à l’analyse prédictive de l’instabilité des réseaux électriques. Pour remédier à cette difficulté, la littérature propose plusieurs logiciels de modélisation et de simulation des réseaux électriques, notamment Matpower, PST, PSAT, PYPower, PYPSA, SPS [42]. Pour des besoins de recherche et d’enseignement, il existe dans le Matlab Central, plusieurs réseaux électriques tests (IEEE 39 bus, IEEE RTS96, etc.) prêts à être utilisés pour la génération des signaux de

stabilité [43]. Mais, l'utilisation de ces modèles nécessite plusieurs modifications de configuration particulière, rendant ainsi fastidieux et ardu le processus de génération massive des signaux. Conformément au principe d'analyse de sécurité (n-1), aucune ligne/barre ou génératrice ne doit être exclue du processus d'analyse de la stabilité [44]. Pour simplifier cette difficulté, une méthodologie simple de simulation d'une centaine de défauts variés (c'est-à-dire : monophasé, biphasé et triphasé, etc.) et localisés sur chacune des lignes électriques a été proposée dans [45]. Car l'identification et la localisation des défauts qui surviennent dans les réseaux électriques jouent un rôle important dans l'analyse de la stabilité, la protection des équipements et la pénétration des systèmes de production renouvelable à contribution variable [46]. Fort de ce constat, la génération massive de données synthétiques représentatives d'un grand nombre de scénarios permet d'améliorer les performances des prédicteurs (algorithme d'apprentissage supervisé, algorithme d'apprentissage non supervisé et apprentissage par renforcement) de stabilité [47].

1.3 Extraction et prédiction des indicateurs d'instabilités

Les informations représentatives de la dynamique réelle des réseaux électriques sont généralement enfouies dans des bases de données. Leurs mises en exergue requièrent généralement une expérience considérable en ingénierie électrique et en traitement du signal. À l'ère de la massification progressive des données, un nombre important d'algorithmes a été développé pour extraire et classer efficacement les prédicteurs de stabilité inspirés des données. Une revue des codes sources de *Deep learning* est proposée en [48] pour la classification des séries chronologiques (TSC) univariées et multivariées. Un total de 8,730 algorithmes est entraîné à partir de 97 jeux de données. Pour cela, un ensemble de neuf méthodes (*Multi layer perceptron, fully convolutional neural networks, residual network, encoders, multi-scale convolutional neural network, time le-net, time convolutional neural networks, time warping invariant echo state network*) regroupées en une seule (*end-to-end deep learning classifiers*) a été ré-implémentée pour la TSC. Il ressort de cette étude que l'approche *end-to-end deep learning classifiers* performe mieux que des méthodes conventionnelles notamment *fully convolutional neural network* et le *deep residual network*. Pour prédire la stabilité transitoire et petit signal des angles rotoriques, [49] implémente les réseaux de neurones convolutionnels (CNN) et *long short*

term memory (LSTM) à partir des phaseurs de tensions collectées dans tout le réseau électrique. Le modèle CNN est entraîné pour prédire l'état de stabilité transitoire du réseau électrique lorsque celui-ci est soumis à une perturbation alors que le modèle LSTM est entraîné pour détecter les évolutions progressives des fréquences d'amortissements locales et interzones.

Les indices de sévérité extraits des signaux des PMUs ont été utilisés dans [25] pour implémenter des algorithmes de machine learning à savoir les arbres de décision, les forêts aléatoires, les réseaux de neurones, les machines à vecteurs de support et les modèles à logique flou. Parmi tous ces outils intelligents, cette référence démontre que les modèles flous sont transparents, facilement interprétables mais globalement moins précis. Les précurseurs d'instabilités ont été utilisés pour former une forêt aléatoire capable d'améliorer les performances des systèmes de protection en cas de perturbation [39]. En effectuant une analyse comparative, on y démontre qu'à partir d'un arbre de décision, on a un taux de fiabilité de 70 % et alors qu'à partir qu'une forêt à 210 arbres ont un taux de fiabilité de 99,9 %. Toutefois, l'article ne se prononce pas sur le temps de réponse de chaque algorithme d'apprentissage. Dans [17], les signaux des PMUs obtenus à partir des barres stratégiques sont utilisées pour analyser les contingences simples et multiples dans le domaine temps-fréquence. Pour cela, la transformée de Fourier à court terme a été implémentée pour extraire les pics de densité spectrale des angles des rotors, la fréquence du système et leurs produits scalaires. Ces indicateurs de stabilité associés à des règles de décision par logique floue et aux réseaux de neurones ont permis d'évaluer la fiabilité et la sécurité des réseaux électriques 2 à 3 secondes après l'élimination des contingences, ce qui est relativement long et donc mal adapté pour une utilisation dans les systèmes de protection pour la sauvegarde de l'intégrité des réseaux.

Plusieurs avancées significatives ont ainsi été menées en *machine learning* afin d'accroître l'observabilité et l'efficacité des automatismes présents dans les réseaux électriques modernes. Toute la question réside actuellement sur leurs particularités dans des contextes tels que : l'analyse de l'écoulement de puissance, l'analyse de la qualité d'onde, l'intégration des systèmes renouvelables, le transport intelligent et le contrôle des charges [50], il ressort de ce questionnement que les données massives à haute dimension

et les modèles de prédiction hybride ont un impact significatif sur les algorithmes d'apprentissage usuels.

Sur un autre plan, les surcharges, congestions et pertes transport et distribution résultant de l'intégration des éoliennes dans le réseau de distribution de l'Algérie (Adrar) ont été analysées par l'algorithme *extreme machine learning* (ELM) [51]. En se servant du flux de puissance unifiée et la structure à une seule couche cachée de l'ELM, cet article réussit à éviter les délestages, minimiser les pertes et les coûts de congestion dans le réseau Adrar. De même, il démontre que l'ELM présente de meilleures performances et temps d'entraînement comparativement aux moteurs d'apprentissage ANN et SVM. Le concept de cartographie de zone et de matrice de bus est introduit dans [52], pour extraire les caractéristiques pertinentes dans diverses zones locales. Ceux-ci sont utilisés dans l'extraction des caractéristiques de niveau supérieur, avant d'être introduits dans un réseau de neurones multicouches entièrement connecté pour la classification des creux de tension. Ainsi, quelles que soient les conditions de fonctionnement, l'approche par cartographie des bus, identifie avec précision les creux de tension dans les réseaux électriques. L'approche hybride *long short-term memory* et *le convolutional neural network* (LSTM-CNN) ont été utilisées dans [53] pour prédire à court terme, la consommation de charge horaire de la Malaisie et la consommation quotidienne de l'Allemagne. Pour chacune des bases de données, les performances du modèle LSTM-CNN sont évaluées à partir de l'erreur quadratique moyenne (RMSE), la moyenne du pourcentage de l'erreur absolue (MAPE) et le R-squared. Il ressort de ce qui précède que, le modèle hybride LSTM-CNN améliore les performances de précision de 83,17 % à 91,18 % (pour les données Allemandes) et 98,23 % (pour les données de la Malaisie).

L'approche *high-dimensional stability indices* (HDSI) est proposée dans [40], pour extraire plus de 500 attributs labellisés décrivant les signaux des générateurs, tels que les vitesses des rotors et des fonctions d'énergie transitoires (RoCoTE). Pour y parvenir, une combinaison de 31 fonctions a été implémentée en utilisant l'analyse du spectre basée sur la méthode du périodogramme, la méthode Welch, les exposants de Lyapunov, la transformée de Fourier et la transformée en ondelette. Il ressort de cette étude que l'indice HDSI extrait de la base de données des vitesses des rotors performe mieux que ceux extraits

des signaux RoCoTE obtenus à partir du dynamic state estimation (DES). Les simulations des transitoires réalisées sur les réseaux IEEE 118 -bus et 16 000 -bus ont permis à [54] de proposer une nouvelle approche par clustering des degrés de similarité basée sur *Quality Threshold* (QT) et *K-means* (KM) permettant d'extraire les attributs communs des signaux des tensions et des fréquences. Cette étude révèle que certaines données de stabilité transitoire, telles que les signaux de tension avec des plages variables entre les bus, nécessitent un prétraitement pour un regroupement approprié. Le prétraitement implique la normalisation des signaux de tension par rapport aux valeurs de pré-défaut. Les signaux de fréquence varient cependant dans une plage similaire dans les réseaux stables à la fois avant et après le défaut et peuvent être analysés sans aucun prétraitement.

Un algorithme de sélection des attributs basé sur les forêts aléatoires et l'élimination récursive des indicateurs est développé dans [55] pour évaluer la stabilité transitoire des réseaux électriques. Dans cette optique, une forêt aléatoire est implémentée à partir des scénarios instables existant dans la base de données. Le taux d'erreur OOB (out-of-bag) a été utilisé pour sélectionner ou éliminer les attributs stratégiques (ou non) pour l'apprentissage d'un SVM. Pour évaluer la sécurité dynamique en prenant en compte les conditions opérationnelles et le changement de topologie, [56] propose d'entraîner (i.e. sous supervision) un ensemble d'arbres de décision non élagués. De manière périodique, les poids de ces arbres sont mis à jour automatiquement pour s'adapter à de nouveaux scénarios. Ces arbres seront utilisés pour évaluer la probabilité d'appartenance et le poids des votes des nouveaux scénarios. Une étude comparative des différents indicateurs d'instabilités a été proposée dans [57]. En se basant sur les critères de précision, fiabilité, sécurité, fausses classifications, fausses alertes et d'exactitude dans l'évaluation de la stabilité/instabilité des réseaux électriques, cette étude taxonomique démontre que le choix d'un indicateur de stabilité est fonction du scénario à analyser, bien qu'en générale les indicateurs de sévérité proposés en [25]-[39] affichent une certaine supériorité au niveau d'un seuil clair d'évaluation entre les cas stables et instables.

Concernant le problème de prédiction de la stabilité dynamique (DSP), qui est plus large que l'évaluation de la stabilité transitoire (TSA), avec les avancées du deep learning appliqué aux séries temporelles, plusieurs auteurs se sont tournés vers l'analyse des signaux

post-contingence [58-61]. Ainsi, le convolutional neural network (CNN) est développé dans [62] sur la base des amplitudes de tension, des angles de rotor et des écarts de fréquence acquis par des PMUs connectées aux barres des générateurs. Le CNN proposé atteint une précision de classification supérieure à 97 % en testant des échantillons de données des PMUs. Une méthode tolérante aux données manquantes est ensuite proposée dans [63] pour l'évaluation de la tension à court terme (STVS) basée sur les données enregistrées après l'apparition d'un défaut. Sur une fenêtre d'observation de $T = 1.2 s$, l'approche STVS présente une meilleure performance par rapport à l'imputation moyenne et à l'arbre de décision. Dans [64], une autre méthode d'évaluation appelée STVS et basée sur les données de PMU, est développée en incorporant des caractéristiques spatio-temporelles des données dans un réseau convolutif de graphes (STGCN). Dans les environnements bruités, le STGCN présente une meilleure précision (98,6%), que les techniques du *spatial-temporal shapelet* (97.4%) et du *long short-term memory* (92.0%).

Le réseau de neurones artificiels (ANN) est proposé dans [65] pour le TSA d'un réseau électrique longitudinal qui comprenait des commandes supplémentaires discrètes tandis qu'un système intelligent hybride (HIS) composé d'un préprocesseur, d'un réseau de neurones et d'un interpréteur est considéré dans [66]. Les relations temporelles entre les caractéristiques de prédiction d'un algorithme LSTM récurrent sur une fenêtre d'observation de 60 ms post-défaut sont utilisées dans [67] pour prédire la TSA en temps réel avec un taux de fausses alarmes de 0,66 % et de faux rejet 0,29 % comparativement à 4,32 % et 4,44 % pour le noyau SVM. Cependant, ce travail a utilisé des ensembles de données artificiellement équilibrés pour l'entraînement et le test, alors qu'en pratique, les cas instables ne représentent qu'une petite fraction de la base de données globale. Une autre méthode de TSA inspirée par l'algorithme LSTM est développée dans [26] en se basant sur les phaseurs de tension de toutes les barres enregistrées sur une fenêtre de temps de $T = 5 s$. Pour ce temps de réponse moyen, l'algorithme LSTM donne une précision de 98,08 % sur les données du réseau test de la Nouvelle-Angleterre à 39-barres.

Un modèle d'apprentissage profond de bout en bout intégrant plusieurs auto-encodeurs de débruitage empilés (MSDAE), une couche de fusion et une couche de régression logistique, est développé dans [68] pour la TSA. Étant donné un SNR de 20 dB, le MSDEA

basé sur l'amplitude de la tension conduit à une précision de 97,23 %, meilleure que tous les classificateurs conventionnels étudiés dans [69]. Dans [70], une approche de simulation dans le domaine temporel montre comment déduire la condition de stabilité à partir des sorties actuellement disponibles en s'appuyant sur des CNN distinctes pour l'extraction de caractéristiques et la prédiction de stabilité. Cette cascade CNN a donné une précision satisfaisante de 99,6 %, meilleure que quatre méthodes existantes basées sur l'apprentissage machine des données (y compris SVM, DT, CNN unique et RNN).

La prédiction de la stabilité transitoire de l'angle du rotor est abordée dans [71] en utilisant un *deep-belief network* (DBN) pour intégrer la réponse du signal dans un espace de représentation de faible dimension et le LSTM pour extraire des caractéristiques de séries temporelles permettant la prédiction des trajectoires des générateurs. L'approche DBN+LSTM a montré une précision supérieure par rapport au prédicteur SVM. Alternativement, un cadre d'apprentissage basé sur le *denoising autoencoder* (DAE) et l'*adaptive synthetic sampling* (ADASYN), est développé dans [72] pour analyser la TSA, cartographier l'ensemble d'apprentissage, réduire et synthétiser les dimensions des échantillons dans l'espace des différentes classes. Par la suite, les données synthétisées sont décodées pour améliorer l'ensemble d'apprentissage et atténuer le déséquilibre des deux classes dans l'ensemble de données d'origine. DAE+ADASYN offre une précision de 98,78%, supérieure celle de l'*extreme machine learning* utilisant des données d'entrée non traitées telles que la puissance active/réactive du générateur, le phaseur de tension du bus, la puissance active/réactive de la branche, etc.

1.4 Dictionnaires fixes et adaptatifs pour l'extraction et classification

Les références [73-75] ont introduit l'information de label dans les algorithmes d'apprentissage de dictionnaire discriminatif pour la reconnaissance, l'analyse et la représentation de modèle parcimonieux. Cette approche *Label consistent K-SVD* présente l'avantage de pouvoir introduire une information labellisée dans chaque colonne du dictionnaire. Cette information discriminative est combinée à l'erreur de reconstruction et à l'erreur de classification pour construire une fonction objective utile à l'apprentissage du dictionnaire surcomplet joint à un classificateur linéaire. Dans [76], l'*orthogonal matching*

pursuit (OMP) est proposé pour reconstruire les supports (*sparse coding*) à haute dimension de signaux contenant du bruit de mesure. Sous la contrainte d'incohérence mutuelle et de l'amplitude minimale des composantes non nulles, cette approche *data-driven* reconstruit exactement le signal en sélectionnant à chaque étape, la colonne la plus corrélée au résidu courant. De même, trois approches à savoir, le dictionnaire discriminatif, l'analyse représentative, et le classificateur ont été appris et introduits dans un modèle joint pour séparer les différentes classes existantes dans une base de données test [77]. Pour y parvenir, l'erreur de reconstruction minimale et l'analyse des incohérences sont utilisées pour définir les sous-dictionnaires associés à chaque classe. Par la suite, la norme ℓ_1 est imposée au support pour limiter son temps de calcul et définir simultanément le pouvoir de discrimination et de représentation des signaux.

Dans le domaine des réseaux électriques, les premières recherches sur l'apprentissage discriminatif par dictionnaire se sont principalement concentrées sur l'extraction et la classification des perturbations liées aux événements de qualité d'onde (*power quality events*). La décomposition parcimonieuse sur un dictionnaire hybride surcomplet (OHD) est proposée dans [78] pour détecter et classer les perturbations simples et multiples présentes dans les signaux. Ce dictionnaire hybride est construit en concaténant les dictionnaires d'atome de Dirac, en cosinus et en sinus. L'OHD permet de décomposer chaque réponse en un signal approximatif et en un signal de détail. Il ressort de cette étude que le signal approximatif, permet de mettre en exergue papillotement, creux de tension, interruption et *dc-offset*, alors que le signal de détail définit correctement les impulsions, oscillations, harmoniques et *notch*. Une représentation parcimonieuse informative et compressive basée sur une formulation mathématique linéaire, qui est intrinsèquement rapide est proposée dans [79] pour extraire et classer les perturbations (PQs) qui affectent la qualité de l'onde électrique. Cette approche ne nécessite pas un entraînement préalable, mais uniquement une formulation mathématique intrinsèque pour classer dix-huit PQs présentes dans les signaux. Une approche conjointe d'intégration cohérente des labels et des dictionnaires d'apprentissage est implémentée dans [80] pour fournir un auto-extracteur parcimonieux et un classificateur multi-classes à partir de l'erreur de reconstruction, du support discriminant parcimonieux, du code d'approximation et de l'erreur de classification. La projection de ces auto-attributs sur les dictionnaires

d'apprentissage a permis de classer les PQs. Un dictionnaire semi-supervisé cohérent labellisé est proposé dans [81] pour estimer et classer des défauts qui affectent les machines. Pour mener à bien cette étude, l'article en question utilise un modèle prédictif des labels existant pour prédire les labels de nouveaux signaux d'entraînement. Ces données étiquetées ont permis de construire un dictionnaire de classification cohérent basé sur le poids de reconstruction adaptatif et la matrice des supports parcimonieux.

La représentation parcimonieuse et la réduction de dimensionnalité aléatoire ont été développées dans [82] pour classer les défauts qui affectent les réseaux électriques. Pour y parvenir, chaque signal test est représenté comme une combinaison linéaire parcimonieuse et surcomplète des signaux d'entraînement. La projection des réductions de dimensionnalité aléatoire est utilisée pour extraire les attributs des défauts et construire une matrice de représentation parcimonieuse. Le type de défauts est ainsi déterminé en fonction du résidu minimum existant entre les échantillons et sa représentation parcimonieuse. La détection, la reconnaissance et la localisation temporelle des défauts multiples (déclenchement des générateurs, des lignes et les pertes de charge) dans les réseaux électriques ont été abordées dans [83]. Pour y parvenir, les attributs présents dans les multiples scénarios générés sont extraits par la méthode du *cluster-based sparse coding* (CSC). Cette étude démontre que le degré de similarité des régions est indépendant du type de défauts. Elle permet ainsi d'approcher la réaction de chaque région comme une combinaison linéaire des défauts. En se basant sur les données des angles de phase des tensions, [84] propose un algorithme en temps quasi réel capable de reconnaître les contingences (simples et multiples) de ligne. Cela passe par la résolution du problème de reconstruction parcimonieuse défini par des calculs itératifs. Cette reformulation qui incorpore les informations internes sur le système et les paramètres des défauts de ligne, permet ainsi de caractériser le flux de puissance linéaire à partir d'un ensemble de coefficients parcimonieux surcomplets.

Dans [85], l'OMP et la K-SVD sont utilisées pour extraire les défauts existants dans la partie rotative du système électromécanique d'une boîte à vitesse. À partir d'un critère d'arrêt prédéfini, l'OMP supprime les signaux modulés et les harmoniques de manière adaptative. Par segmentation périodique et décalage circulaire, cette approche reformule le

signal résiduel et construit un dictionnaire de transitoires unidimensionnel. L'ensemble de ces transitoires répétitifs permet de former un dictionnaire optimisé. L'approche *deep belief network (DBN)* basée sur la représentation non linéaire est proposée dans [86], pour pré-entraîner de manière non supervisée la distribution des données de stabilité transitoire sur la base des paramètres de l'indice de classification de la précision (ECA). Cette structure apprise qui prend en compte la topologie du réseau électrique est combinée à un modèle linéaire simple pour classer les signaux stables et les signaux instables. Cette approche a démontré une bonne performance sur les données régionales de la Chine centrale, par rapport à la machine à vecteurs de support (SVM). Dans [87], une nouvelle approche de détection et de localisation automatique des transitoires est proposée pour analyser les perturbations des ondes électriques à partir du dictionnaire d'atome surcomplet et de la norme ℓ_1 de minimisation. Ce dictionnaire est construit en utilisant une base d'identité et une base en cosinus. Comparativement à la méthode basée sur la décomposition en ondelettes, cette méthode a présenté de bonnes performances sur quatre signaux transitoires générés.

Un dictionnaire parcimonieux d'apprentissage est proposé dans [88] pour classer rapidement et avec une grande précision les perturbations des ondes électriques générées par la simulation des réseaux électriques. Ce dictionnaire est construit en combinant une matrice identité, l'algorithme K-SVD et le *fast adaptive matching pursuit* (FAMP). Les attributs et classificateurs primaires flous obtenus à partir des coefficients de la décomposition parcimonieuse sont utilisés pour entraîner les arbres de décision. Une méthode fiable basée sur l'algorithme de *matching pursuit* (MP) est présentée dans [89], pour détecter et caractériser les oscillations transitoires des réseaux électriques. Elle définit une perturbation oscillatoire, en utilisant le coefficient de corrélation entre le dictionnaire & le signal test, la fréquence et le facteur d'amortissement. Cette méthode est très précise pour des signaux transitoires bruités et de faible amplitude. La *KSVD-compressive sampling matching pursuit* (KSVD-CoSaMP) est proposée dans [90] pour améliorer les performances de reconstruction de la *power quality events*. Ce travail démontre par simulation expérimentale que pour neuf types de perturbations différentes (c'est-à-dire ayant rapport signal/bruit de 30 dB) du signal de tension, la KSVD-CoSaMP performe mieux que les dictionnaires fixes basés sur la *discrete cosine transform* (DCT).

1.5 Conclusion et pistes d'amélioration

1.5.1 Synthèse

Comme il ressort de la revue de la littérature, le problème de la TSA a été extensivement étudié à la fois dans le passé et tout récemment via l'intelligence des données. Bien que l'approche par attribut ait été investiguée par différents auteurs [25]-[39], ce n'est que tout récemment que l'auteur de cet ouvrage en a généralisé l'utilisation en exploitant la méthodologie HCTSA pour combiner plusieurs familles d'attributs utilisés séparément par le passé, à savoir : les exposants de Lyapunov, les ondelettes et transformées de Fourier, ainsi que l'énergie spectrale à l'aide des méthodes du périodogramme et de Welch. On peut ainsi pour chaque centrale du réseau, extraire plusieurs centaines d'attributs précurseurs de stabilité par analyse de séries chronologiques de vitesses de rotor de générateur et de signal de RoCoTE dérivés de l'estimation de l'état dynamique (DSE) à partir du streaming des données PMU centralisées en centre conduite [91].

Les informations sur la dynamique du système électrique provenant des exposants de Lyapunov permettent l'extraction d'attributs liés à la stabilité ou à l'instabilité en quantifiant les variations des valeurs propres et le degré de sensibilité du système linéarisé sous-jacent. De même, l'analyse spectrale des signaux de réponse du générateur capture les processus énergétiques perturbateurs conduisant le système au bord de l'instabilité. Cette analyse est mise en œuvre en appliquant des fenêtres rectangulaires et de Hamming, ce qui donne les méthodes du périodogramme et de Welch. Une décomposition temps-fréquence de niveau 2, 3, 4, 5, effectuée sur les ondelettes de Daubechies et Symlet, facilite l'extraction statistique des attributs énergétiques contenus dans les signaux de stabilité lorsque ceux-ci sont fortement instationnaires. Ces méthodes complémentaires permettent de définir une matrice de clusters d'attributs caractérisant l'état de stabilité dynamique du système électrique en phase post-contingence, lorsque sa stabilité est à risque.

La revue de littérature à la section 1.4, démontre que l'approche d'apprentissage par dictionnaire a surtout été appliquée en classification des signaux PQs et, dans une moindre mesure, dans la reconnaissance des défauts des machines et des systèmes électriques. Cet outil pourtant prometteur n'a donc pas encore été démontré dans la prédiction de la stabilité dynamique des réseaux électriques. De plus, bien que les classifications par apprentissage

profond conduisent à une précision relativement satisfaisante, le moteur de décision basé sur le signal brut est opaque et difficile à expliquer aux experts responsables de la planification et de l'exploitation du réseau. L'arbre de décision est pour sa part une approche plus interprétable, mais sa précision et sa fiabilité sont généralement insatisfaisantes pour les fenêtres de décision courtes [57]. Pour combler les gaps ainsi identifiés dans la recherche, on peut envisager un nouvel apprentissage supervisé des signaux du DSE sur un dictionnaire surcomplet pour la prédiction de la stabilité dynamique avec un temps de réponse rapide après l'élimination des défauts. Il s'agirait alors du tout premier cadre systématique comprenant des algorithmes numériques détaillés pour prédire la stabilité du système électrique après perturbation à l'aide de signaux de réponse basés appris sur un dictionnaire surcomplet et un codage parcimonieux. L'auteur démontre dans un article soumis pour publications au IEEE Trans. On PWRs [40] qu'il est possible de construire hors-ligne des dictionnaires aux caractéristiques clairsemés pour reconstruire les réponses des signaux de vitesse du rotor et détecter simultanément l'état de stabilité dynamique associé. De plus, le nouveau D-KSVD bidirectionnel proposé dans l'article susmentionné pourrait être utilisé pour détecter les échantillons manquants, résultant de la perte de données lors de la transmission des données PMU et combler le trou en utilisant une prédiction basée sur un dictionnaire.

Toujours dans une approche d'apprentissage de signaux de stabilité sur un dictionnaire, Teukam et al. ont proposé d'utiliser des dictionnaires indépendants de la base de données comme ceux utilisés en [11] ou spécifiés dans le Toolbox d'ondelettes de Matlab à savoir les dictionnaires basés sur plusieurs concaténations de deux (c'est-à-dire [DHT|DCT], [DCT|DWT] et [DHT|DWT]) et trois (c'est-à-dire [DST|DHT|DCT], [ID|DCT|DST] et [DWT |DHT|DCT]) sous-dictionnaires fixes. Cette approche à double dictionnaires fixes vise à reconstruire les signaux de réponses sur chaque dictionnaire stable et instable séparément, pour ensuite prédire le statut de la stabilité par évaluation du degré d'appartenance à chaque label. Concrètement, l'article suscitè développe et met en œuvre un nouvel arrangement d'atomes capable d'améliorer la surveillance et le contrôle de la stabilité à grande échelle en utilisant une approche de reconstruction de signal à double dictionnaire. À cette fin, des structures fixes et adaptatives sont étudiées sur la base de l'apprentissage des dictionnaires à double ou triple concaténations pour la reconstruction

parcimonieuse des vitesses de rotor. Ensuite, l'état de stabilité est déterminé sur la base de la RMSE associée à la prédiction de la vitesse du rotor en utilisant chaque dictionnaire stable/instable pour concevoir le niveau d'appartenance à chaque classe. Le rapport de RMSE de ces deux prédictions séparées est proposé comme mesure de la stabilité (ou de l'instabilité) du signal post-perturbation observé donné sur une courte fenêtre de données de 4,5 cycles.

1.5.2 Propositions

Considérant les limites identifiées dans cette revue de littérature, notre travail aborde l'exploitation des données de réponses post-contingence par deux approches complémentaires d'intelligence des données : 1) une analyse non parcimonieuse d'une base d'attributs se chiffrant par centaines, calculés automatiquement par l'analyse numérique massive des séries chronologiques de signaux de réponses post-contingence des générateurs; et 2) une analyse parcimonieuse exploitant l'apprentissage supervisée de grands dictionnaires surcomplets pour habiliter une prédiction de l'instabilité sur de courtes fenêtres de données avec une représentation vectorielle creuse (contenant un grand nombre de zéros) et donc numériquement très efficiente en plus de l'interprétabilité inhérente des atomes constituant les dictionnaires. Au niveau méthodologique, l'approche non parcimonieuse vise à implémenter plusieurs méthodes analytiques combinées (notamment la transformée de Fourier, la transformée en ondelette, la méthode de Welch, la méthode de périodogramme et les exposants de Lyapunov) pour extraire du signal de réponse de chaque générateur des centaines d'attributs labellisés et servant à construire un espace physique d'indicateurs de stabilité à haute dimension (HDSI). Ceux-ci sont ensuite utilisés pour développer les prédicteurs de stabilité sur la base d'algorithmes standard de machine learning, par exemple le CNN, LSTM, SVM, l'AdaBoost ou les forêts aléatoires. L'approche parcimonieuse implémentée consiste à développer deux techniques complémentaires : 1) un dictionnaire d'apprentissage supervisé joint (SLOD) au classificateur et 2) vingt dictionnaires d'apprentissage séparés des signaux associés aux cas stable/instable. Alors que le SLOD utilise des dictionnaires adaptatifs inspirés des données mesurées et appris hors-ligne, la deuxième approche utilise des dictionnaire fixes pour reconstruire séparément les signaux des classes stables et instables. Dans les deux cas, l'étape finale consiste à identifier automatiquement en temps réel, la classe d'appartenance

d'une réponse par reconstruction des signaux associés à partir des dictionnaires appris hors-ligne.

L'analyse parcimonieuse des réponses des générateurs sur un dictionnaire d'apprentissage adaptatif joint au classificateur sera implémenté à partir de l'algorithme *K-singular value decomposition* (KSVD) couplé à l'*orthogonal matching pursuit* (OMP), afin de reconstruire et prédire la stabilité dynamique des réseaux électriques. De plus, vingt décompositions parcimonieuses des signaux sur des dictionnaires fixes (simples et hybrides) permettront de développer des classificateurs prédisant chaque classe séparément sur la base de la transformée en cosinus discrète (DCT), en sinus discrète (DST), en ondelette (DWT), de la transformée de Haar (DHT), et le dictionnaire de Dirac (DI) couplés à l'*orthogonal matching pursuit* (OMP). L'utilisation de dictionnaires fixes se traduit par un apprentissage séparé des cas stable et instable avec en temps réel une reconstruction des signaux selon chaque dictionnaire considéré séparément, suivi de l'identification du niveau d'appartenance d'un signal de réponse post-contingence à chaque classe, en fonction de l'importance relative du RMSE de reconstruction sur chaque dictionnaire stable ou instable.

Chapitre 2 — Extraction et classification des attributs de stabilité à haute dimension

2.1 Introduction

Les séries chronologiques de signaux de réponse post-perturbation contiennent des informations dormantes et non transparentes nécessaires pour une bonne compréhension de la dynamique des systèmes électriques en temps réel. L'analyse et l'interprétation rapide de la signification physique de ces séries temporelles nécessitent généralement une expérience considérable en connaissance réseaux et en traitement du signal. Il convient donc de fournir aux experts en intelligence artificielle (IA) et aux ingénieurs une méthode simple pour extraire et classer des attributs de stabilité transparents et facilement interprétables [45]. Mettre en évidence la signification physique de ces attributs permet de s'éloigner de l'approche boîte noire de l'évaluation de la stabilité dynamique. À la lumière de la littérature ci-dessus, ce chapitre se concentre sur l'utilisation combinée des exposants de Lyapunov, des transformées en ondelettes et de Fourier, et de l'énergie spectrale en utilisant des périodogrammes et des méthodes de Welch pour extraire les attributs précurseurs de la stabilité des séries chronologiques des vitesses des rotors et des indices RoCoTE dérivés de l'estimation de la dynamique du système de puissance.

2.2 Génération des bases de données de stabilité sur Simscape Power System (SPS)

2.2.1 Estimation de l'état dynamique

L'analyse de la stabilité dynamique des réseaux électrique, passe par la linéarisation des équations différentielles/algébriques définies par [92]:

$$\dot{x}(t) = f(x(t), y(t), u(t), p) \quad (1)$$

$$0 = g(x(t), y(t), u(t), p) \quad (2)$$

$$x_i^{min} \leq x_i \leq x_i^{max}, i \in \Xi ; u_j^{min} \leq u_j \leq u_j^{max}, j \in \Omega ; p_l^{min} \leq p_l \leq p_l^{max}, l \in \Gamma .$$

Où: $x \in \mathbb{R}^n$, $y \in \mathbb{R}^m$, u, p, f, g, Ξ et Γ représentent respectivement le vecteur d'état du système, le vecteur d'état de l'équation algébrique, le vecteur d'entrée du système, les paramètres du modèle, la fonction non linéaire différentielle, la fonction non linéaire algébrique, les limites inférieures & supérieures du contrôleur, les limites des variables d'état dynamique, les limites des entrées utilisées pour le contrôle du modèle [5]. La fonction $g(x)$ représente les contraintes algébriques dictées par les équations d'équilibre en régime permanent. La plupart des logiciels (Matlab/Simulink, PSSE, PSLF, PowerWorld, DSATools, etc.) implémentent Eq. (1-2) [93-95]. Après l'élimination du défaut, les signaux des générateurs synchrones (mesurés directement ou extrapolés à partir de mesures directes) sont utilisés pour calculer des indicateurs de stabilité décentralisés, à savoir le RoCoTE et TEF [96].

2.2.2 Algorithme de génération graduelle des signaux de stabilité sur

SPS

Le IEEE 39 -bus, est modélisé sur SPS pour simuler l'algorithme de recherche de la limite de stabilité des machines face aux perturbations. L'utilisation de cet algorithme nécessite une configuration particulière des différents points d'application des défauts. L'originalité de cette méthode est qu'elle permet, en un clic, de simuler plusieurs types de défauts (monophasé, biphasé et triphasé, etc.) sur toutes les lignes et proche des barres génératrices. Pour faciliter cette simulation, un fichier Excel contenant des informations sur chaque ligne/barre est créé et enregistré dans le fichier courant de Matlab. Ces informations vont faciliter la simulation ciblée des défauts, l'activation et désactivation des disjoncteurs de protection. Chaque ligne génératrice est équipée de deux disjoncteurs triphasés permettant d'isoler le défaut et de ralentir sa propagation dans le réseau. Ainsi, ces défauts sont appliqués à $\tau = \chi$ secondes et sont éliminés à $\tau = (\chi + \omega/60)$ secondes, permettant ainsi une propagation d'un défaut de w cycles. Immédiatement après la simulation d'une contingence sur une barre/ligne, le statut du commutateur de phase du bloc « Three-Phase Fault » est automatiquement désactivé par le pseudo-code MATLAB.

La Tableau I présente l'ensemble des configurations des scénarios simulés sur les réseaux électriques IEEE 39 -bus.

Tableau I : Configuration des scénarios simulés sur IEEE 39 bus.

Perturbations		Localisation des défauts et configuration
Défauts triphasés terre (3-Ph G)	Load-on Events	Défauts sur chaque barre génératrice, Variation de charge: +20% MW Gain.
	Load-off Events	Défauts sur chaque barre génératrice, Variation de charge: -20% MW Loss.
	Déclenchement de ligne	Défauts sur chaque ligne: Appliqué à 50%
	Perte de générateur	Défauts sur chaque barre génératrice, Variation de charge: -10% & -30% MVA Loss.

Pour une simulation temporelle de k secondes, lorsqu'un défaut est appliqué sur la ligne/barre j , le temps de commutation de tous les autres disjoncteurs des lignes est fixé à au moins $(k+1)$ secondes. En d'autres termes, ce temps empêchera les disjoncteurs $(j-1)$ des autres lignes de déclencher. Une fois le pseudo-code simulé, les signaux de stabilité sont ré échantillonnés sur une fenêtre fixe afin de faciliter leurs stockages et leurs explorations. Immédiatement après l'élimination des défauts, 150 échantillons sont extraits par signaux générés immédiatement après l'élimination des défauts.

2.2.3 Détection d'instabilité du réseau électrique

En présence d'une perturbation, la différence des valeurs absolues des angles rotoriques de toutes les combinaisons (i, m) des machines est calculée et comparée à 180° . Si la différence des angles $\overline{\Delta\delta}_i(t)$ ne dépasse pas 180° , ou si la perturbation provoquant l'oscillation du rotor est rapidement supprimée, la machine peut rester en synchronisme avec le système [97]. Considérons un ensemble de contingence $i \in \Omega^c$, l'angle du rotor $\delta_i^p(t)$ du générateur p est utilisé pour calculer l'écart maximal $\overline{\Delta\delta}_i(t)$ entre n'importe

quelle paire $p, m \in \Omega^g$ de générateurs à l'instant t et le comparer à une valeur de coupure $\gamma = \pi$.

$$\psi_i(t) = \frac{\gamma - \overline{\Delta\delta}_i(t)}{\gamma + \overline{\Delta\delta}_i(t)} = \begin{cases} 1 & \text{Stable si } \psi_i > 0 \\ 0 & \text{Instable si } \psi_i \leq 0 \end{cases} \quad (3)$$

$$\overline{\Delta\delta}_i(t) = \max(|\delta_i^p(t) - \delta_i^m(t)|) \quad (4)$$

Lorsqu'une perturbation affecte un système électrique, deux cas d'étude sont envisageables : si $\overline{\Delta\delta}_i(t)$ est élevé, les générateurs auront tendance à se désagréger et la différence avec γ (qui est prise égale à π) sera faible et donc l'indicateur sera proche de zéro. Ainsi, à chaque étape de simulation, une décision binaire sur l'état de stabilité des réponses des générateurs est calculée et simultanément stockée dans la base de données.

2.3 Approche d'extraction des attributs de stabilité

La classification des séries temporelles basée sur des attributs transparents était limitée à un petit nombre d'attributs possibles (généralement de 1 à 10 selon les auteurs), avec une difficulté notable à généraliser les attributs ainsi choisis à de nouveaux réseaux et de nouvelles configurations, car certains des attributs proposés sont fonction de l'inertie des machines et de la topologie du réseau [39]. La Figure 1 résume un ensemble de différence existantes entre l'approche conventionnelle et l'approche d'extraction des indices de stabilité à haute dimension (HDSI).

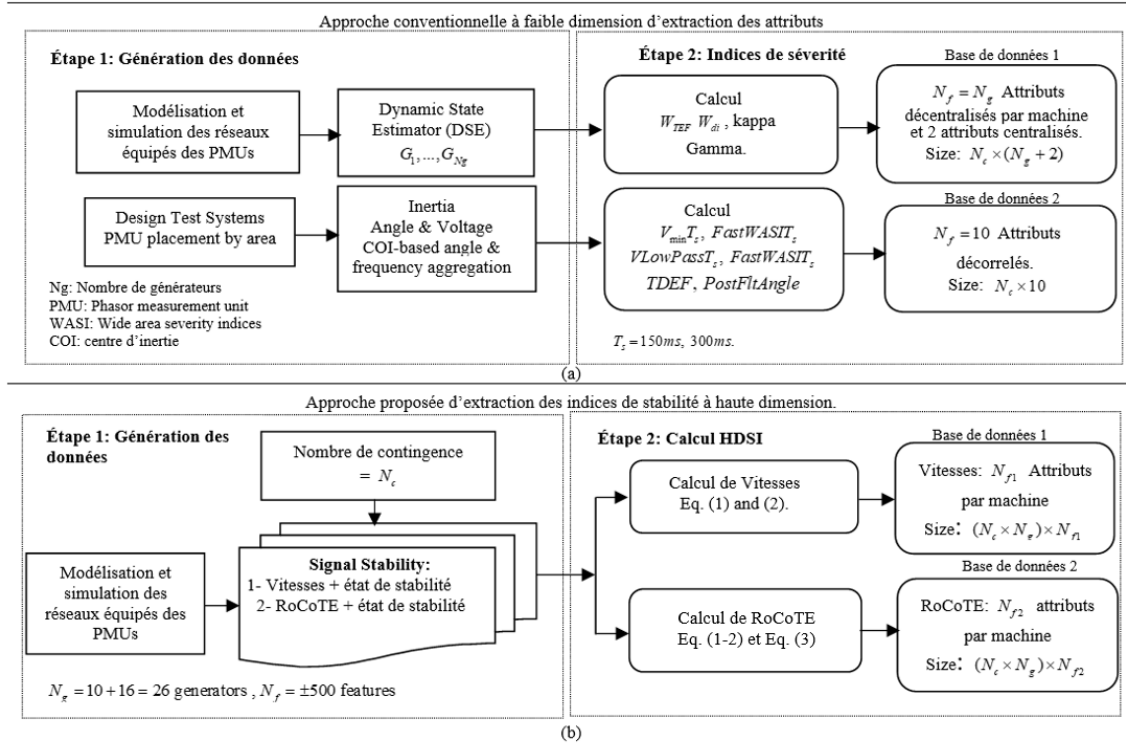


Figure 1 : Schéma conventionnel (a) et proposé (b) pour l'extraction des attributs [96].

Dans une approche conventionnelle, les signaux de stabilité enregistrés au niveau des machines sont utilisés pour calculer des indicateurs centralisés (*kappa* et *gamma*) et décentralisés (W_{di} et W_{TEF}), voir Figure 1(a). En revanche, le nouveau paradigme HDSI permet d'extraire plus de 500 attributs labellisés de chaque signal (vitesses des rotors et RoCoTE, etc.), voir Figure 1(b). Cette section propose de surmonter ces limitations en adoptant une approche de séries chronologiques hautement comparatives (HCTSA) [92] pour extraire un nombre massif d'attributs décrivant les réponses des machines, sans faire d'hypothèses sur la topologie et les paramètres du réseau électrique. Le principe central de l'approche proposée est qu'en augmentant considérablement le type et le nombre d'indicateurs d'instabilités basés sur les données des PMUs, nous devons être en mesure d'améliorer les performances et la transparence des prédictors de stabilité.

2.3.2 Approche conventionnelle d'extraction à faible dimension

Dans une approche conventionnelle, Figure 1(a), les signaux de stabilité enregistrés au niveau des machines sont utilisés pour calculer les indices RoCoTE et la fonction d'énergie transitoire (TEF). Cette approche fait référence aux travaux de [98] qui utilisent moins de 10 attributs pour analyser la stabilité et estimer le *remedial action scheme* (RAS) des réseaux électriques. Deux principaux attributs ont véritablement retenu notre attention, il s'agit: des indicateurs décentralisés (W_{di}) ou des indicateurs centralisés (χ_{COP} et κ_{COP}).

$$W_{di} = \frac{dW_i}{dt} = - \left[\frac{T'_{d0i}}{(x_d - x'_d)} \left(\frac{de'_{qi}}{dt} \right)^2 + \frac{T'_{q0i}}{(x_q - x'_q)} \left(\frac{de'_{di}}{dt} \right)^2 \right] \quad (5)$$

$$\chi_{COP} = \sum_{i=1}^{Ng} \delta_i (\omega_i - \omega_i^{COP}); \quad \kappa_{COP} = \sum_{i=1}^{Ng} \omega_i (\delta_i - \delta_i^{COP}) \quad (6)$$

La collecte des vitesses des rotors, et la résolution des Eqs. (3), (4) et (5), permettent d'extraire les attributs caractéristiques de la dynamique des réseaux électriques grâce à l'approche HCTSA. De plus, ces séries temporelles (vitesse des rotors, RoCoTE et TEF) collectées sont instantanément labellisées, suivant le principe décrit à l'Eq. (3).

2.3.3 Nouvelle approche d'extraction à haute dimension

Ce nouveau paradigme, Figure 1(b), d'extraction des indicateurs basé sur l'approche HDSI, consiste à appliquer quatre fonctions de Lyapunov, cinq fonctions d'analyse spectrale et vingt-deux transformations en ondelettes, à chaque réponse de générateur. Ces méthodes ont permis d'extraire plus de 500 attributs par signal. La Figure 2 présente un résumé de l'ensemble des méthodes d'extraction appliquées sur les bases de données des vitesses des rotors (D_1) et des indices RoCoTE (D_2).

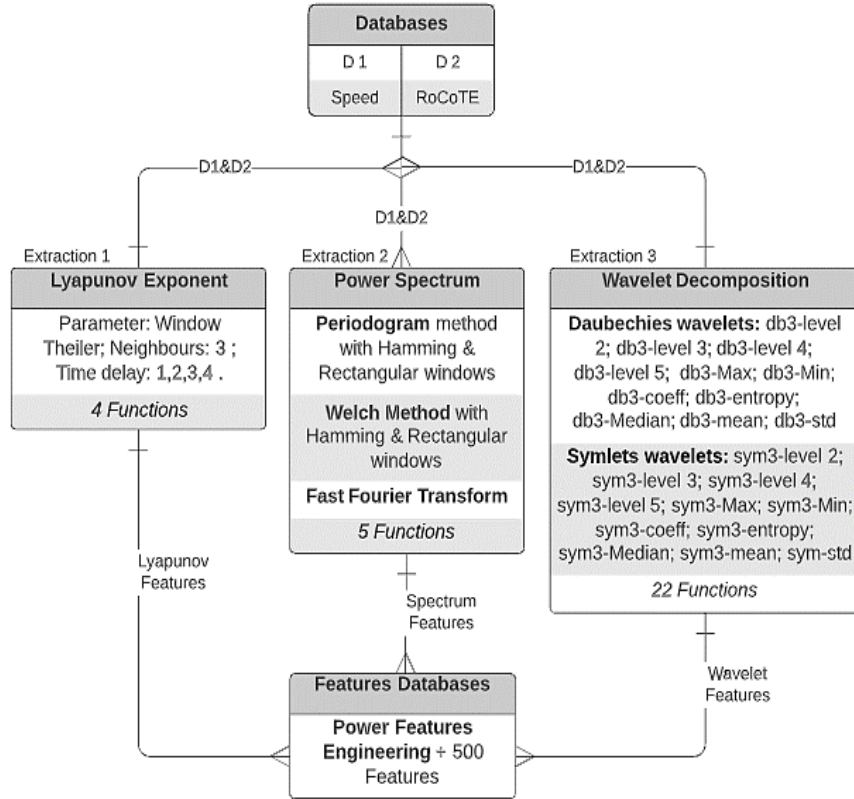


Figure 2 : Bloc diagramme utilisé pour l'extraction des HDSI [96].

2.3.3.1 Les Exposants de Lyapunov

Les exposants de Lyapunov permettent de prédire la stabilité ou l'instabilité pendant la période post-perturbation, à partir des trajectoires des variables (c'est-à-dire : vitesse, tension, RoCoTE, etc.). Le plus grand exposant de Lyapunov est considéré comme une métrique appropriée pour la prédiction de stabilité. L'Eq (8) permet de caractériser la dynamique des réseaux électriques immédiatement après l'élimination des défauts.

$$\Lambda(x) = \lim_{t \rightarrow \infty} \left[\frac{\partial \Phi(t, x)^T}{\partial x} \frac{\partial \Phi(t, x)}{\partial x} \right]^{1/2t} \quad (7)$$

Où : $\Phi(t, x)$ représente la solution de la matrice limite Λ . Soit $\Lambda_i(x)$ les valeurs propres de la matrice limite $\Lambda(x)$ [97]. Les exposants de Lyapunov $\lambda_i(x)$ sont définis :

$$\lambda_i(x) = \log \Lambda_i(x) \quad (8)$$

La valeur négative (positive) du maximum des exposants de Lyapunov $\lambda_i(x)$ implique que le réseau électrique est stable (instable) autour d'un point d'équilibre. Les informations sur la dynamique du système électrique provenant des exposants de Lyapunov permettent l'extraction d'attributs liés à la stabilité ou à l'instabilité en quantifiant les variations des valeurs propres et le degré de sensibilité du système.

2.3.3.2 Analyse spectrale

La transformée de Fourier rapide, le périodogramme et la méthode de Welch permettent d'estimer la densité spectrale de chaque série temporelle sur des fenêtres (rectangulaires et de Hamming) de courte durée immédiatement après l'élimination du défaut.

2.3.3.2.1 Transformée de Fourier rapide

La transformation de Fourier est une opération qui permet de représenter en fréquence les signaux qui ne sont pas périodiques. Elle permet de mettre en évidence les transitoires introduits par les défauts ainsi que leurs décroissances. Considérons un signal dépendant du temps $x(t)$, sa transformée de Fourier est définie par [99]:

$$X(f) = \mathcal{F}(x(t)) = \int_{-\infty}^{+\infty} x(t)e^{-2i\pi ft} dt \quad (9)$$

2.3.3.2.2 Méthode de périodogramme

La méthode du périodogramme divise la caractéristique de stabilité en plusieurs segments de longueur précise qui sont progressivement décalés et traités par la transformée de Fourier rapide $X(f)$. Le périodogramme $P_{f,M}(\omega)$ est défini comme la grandeur au carrée de la transformée de Fourier à temps discret de x divisé par M , selon [100]:

$$x_w(n) = w(n)x(n) \quad (10)$$

$$P_{f,M}(\omega) = \frac{1}{M} \left| \sum_{n=0}^{M-1} x_w(n)e^{-j\omega n} \right|^2 \quad (11)$$

Où : M et $w(n)$ représentent le nombre de segments dans le signal et la fenêtre de segmentation. $\omega = 2\pi f/F_e$, désigne la pulsation numérique, $f = k \times F_e/M$, $k = 0, \dots, M - 1$.

2.3.3.2.3 Méthode de Welch

La mise en œuvre de la méthode de Welch revient à appliquer plusieurs périodogrammes à l'aide d'une fenêtre glissante [101]:

$$x_m(n) = w(n)x(n + mR) \quad (12)$$

$$W_{f,M}(k) = \frac{1}{M} \left| \sum_{n=0}^{M-1} x_m(n) e^{-j2\pi nk/M} \right|^2 \quad (13)$$

Lorsqu'une perturbation est appliquée sur un réseau électrique, elle est soumise à des variations lentes ou brèves des conditions initiales.

2.3.3.3 Transformée en ondelette

La transformée en ondelette est une fonction qui permet de mettre en évidence les différentes fréquences d'oscillations (dans le plan temps-échelle) du signal. Les transformées en ondelettes de Daubechies et Symlet de niveau 2, 3, 4 et 5 sont appliquées à la base de données de stabilité pour extraire les informations discriminantes dans le domaine fréquentiel [102-103]. L'expression de la transformée en ondelette continue, calculée à partir de l'ondelette mère Ψ , est définie par :

$$\langle x, \Psi_{a,b} \rangle = \frac{1}{\sqrt{a}} \int_t x(t) \Psi\left(\frac{t-b}{a}\right) dt \quad (14)$$

Où : a et b représentent les facteurs de dilatation et de translation. La décomposition en ondelettes permet donc de détecter le point de changement introduit dans le réseau électrique par l'apparition d'un défaut.

2.4 Classification des indicateurs de stabilité

Les indicateurs de stabilité extraits de la base de données d'entraînement sont utilisés pour former des outils capables de prédire l'instabilité des différents scénarios. Une comparaison préliminaire de la distribution statistique des attributs de stabilité extraits des bases de données est réalisée afin d'inspecter visuellement et mettre en évidence leur capacité à discriminer la dynamique des réseaux électriques. Les cinq principales approches d'apprentissage (*support vector machine, decision trees, random forest, artificial neural network and adaptive boosting*) sont mises en œuvre à partir d'une base

de données contenant des informations labellisées.

2.4.1 Texture des attributs pour la machine learning

Les signaux de stabilité (vitesses des rotors et RoCoTE) générés à partir du réseau IEEE 39 -bus sont labellisés suivant un code couleur : bleu pour des réponses stables et rouges pour des réponses instables. Les Figures 3(a) et 3(b) illustrent les allures des bases de données des vitesses des rotors et des indices RoCoTE obtenues en simulant les défauts sur les lignes du réseau IEEE 39 -bus. Chaque signal de la base de données d’entraînement (représenté par 150 échantillons extraits immédiatement après l’élimination des défauts) est labellisé à partir des informations suivantes : 1) le numéro d’identification du générateur qui a produit la réponse enregistrée, 2) la ligne sur laquelle le défaut a été appliqué, 3) le type de défaut et 4) l’état de stabilité globale du réseau électrique.

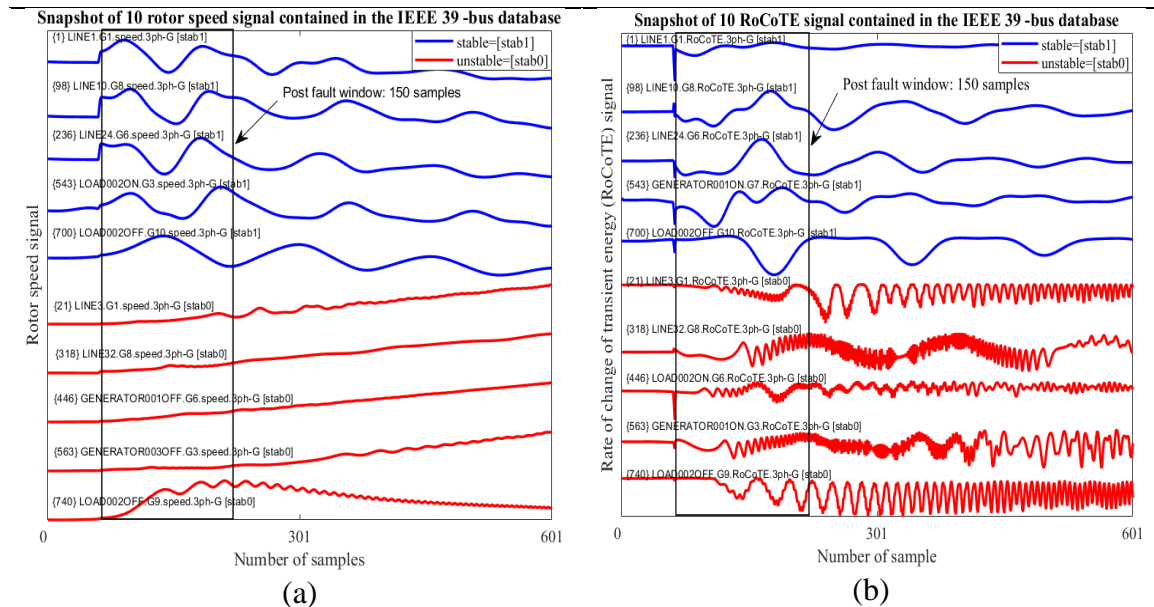


Figure 3 : Illustration de 10 signaux (a) des vitesses des rotors et (b) RoCoTE stable (bleu) et instable (rouge) [96].

Les 31 fonctions d’analyse appliquées sur la base de données des vitesses des rotors (ou des RoCoTE) du réseau IEEE 39 -bus permettent d’extraire 584 attributs (605 attributs) significatifs, voir Figure 4. Les lignes de la matrice d’attributs représentent le nombre de

signaux présents dans la base de données tandis que les colonnes représentent le nombre d'attributs extraits de chaque signal.

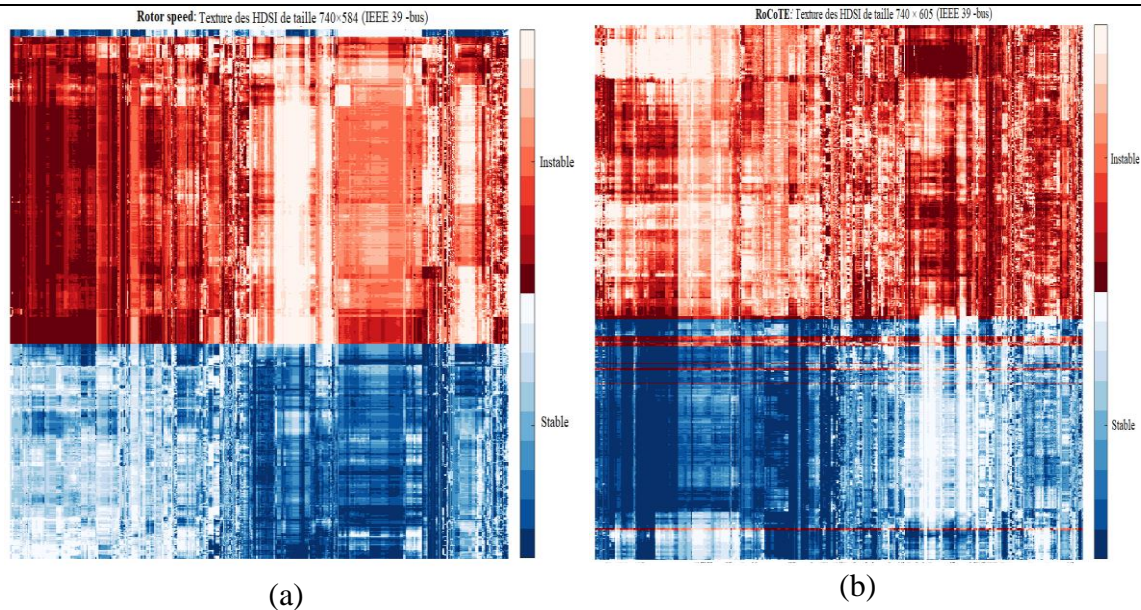


Figure 4: Texture des HDSI issues des (a) vitesses des rotors et des indicateurs (b) RoCoTE (Modèle : IEEE 39 -bus) [96].

2.4.2 Machine à vecteurs de support

La *support vector machine* (SVM) est l'un des classificateurs les plus puissants en termes de compromis entre efficacité et complexité, en particulier dans le contexte de la prédiction de stabilité, elle a été sélectionnée pour implémenter l'indicateur HDSI de base dans Matlab. Son utilisation dans l'analyse prédictive de la stabilité dynamique permet de séparer les signaux de la base de données en deux classes : stables ou instables. Pour améliorer sa capacité de généralisation, l'algorithme SVM proposé utilise la validation croisée [104-105] pour garantir que chaque série temporelle disponible est utilisée à la fois dans les ensembles d'apprentissage et de validation. Cette approche minimise l'erreur globale attendue dans la base de données complète en sélectionnant l'hyperplan qui discrimine les signaux de stabilité afin de maximiser la distance entre sa marge et l'échantillon le plus proche. Étant donné un ensemble $f(x, \alpha)$ de modèles indexés par un paramètre d'accord α , on désigne par $f^{-k}(x, \alpha)$ l'ajustement du α^{th} modèle avec la k^{th} partie des données. Ensuite, pour cet ensemble de modèles, nous définissons :

$$CV(\hat{f}, \alpha) = \frac{1}{N} \sum_{i=1}^N L(y_i, \hat{f}^{-k}(x_i, \alpha)) \quad (15)$$

Où $CV(\hat{f}, \alpha)$ fournit une estimation de la courbe d'erreur, et nous trouvons le paramètre d'accord qui minimise $\hat{\alpha}$. Notre modèle final est $f(x, \hat{\alpha})$, que nous adaptons ensuite pour toute notre base de données. La SVM prédit la stabilité des signaux des vitesses des rotors du réseau IEEE 39 -bus en utilisant deux attributs PC-1 (avec une précision de 96 %) contre 49 % pour PC-2, voir Figure 5(a). La SVM prédit la stabilité du RoCoTE du réseau IEEE 39 -bus en utilisant deux attributs PC-1 (avec une précision de 91 %) contre 57 % pour PC-2, voir Figure 5(b).

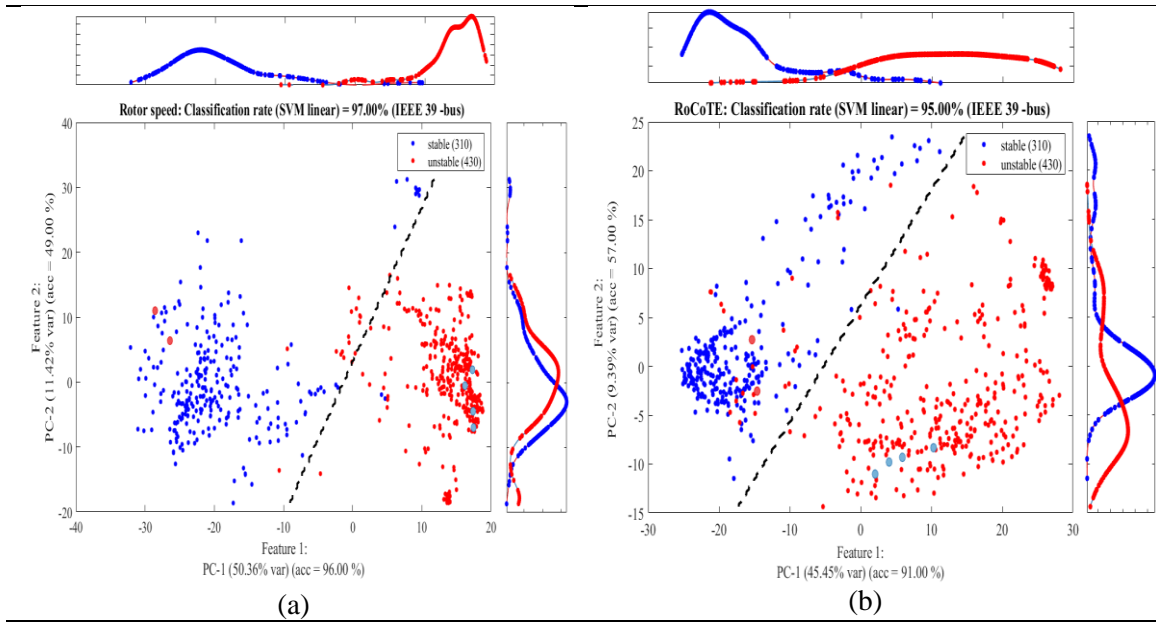


Figure 5: HDSI des (a) vitesses des rotors et (b) RoCoTE issues l'algorithme SVM (Modèle : IEEE 39 bus) [96].

Pour l'apprentissage, les signaux sont répartis de manière aléatoire en $K = 10$ groupes. Chaque fois que, l'un des groupes est utilisé comme ensemble de tests, les neuf autres groupes sont utilisés comme ensemble d'apprentissages à chaque itération. En observant les bases de données (c'est-à-dire : vitesse du rotor et RoCoTE), nous observons sur les Figures 5(a)&5(b), que PC-1 et PC-2 suffisent à séparer approximativement par une ligne, la plupart des bleus et des rouges. Pour améliorer la séparation, il faudrait un hyperplan d'au moins 10 dimensions (nombre de composantes principales significatives selon Rattle).

La distribution en bleu et rouge confirme que PC-1 a une valeur très négative lorsqu’il est instable alors que PC-2 a un chevauchement prononcé entre le bleu et le rouge qui donne une précision limitée pour séparer les cas stables et instables. Une précision de 50 % signifie que la classification n’est pas plus précise, ce qui confirme que le chevauchement des distributions rend la séparation stable/instable très mauvaise.

La mise en œuvre de la SVM basée sur les vitesses des rotors et les indices RoCoTE seules aboutit aux matrices de confusion illustrées par la Figure 6 pour IEEE 39 –bus et Figure 7 pour IEEE 68 –bus. La SVM des vitesses des rotors détecte correctement 310 signaux stables (sur 310 réponses stables) et 430 signaux instables (sur 430 réponses instables), voir Figure 6(a). Alors que la SVM des indices RoCoTE détecte correctement 310 signaux stables (sur 310 réponses stables) et 426 signaux instables (sur 430 réponses instables), voir Figure 6(b).

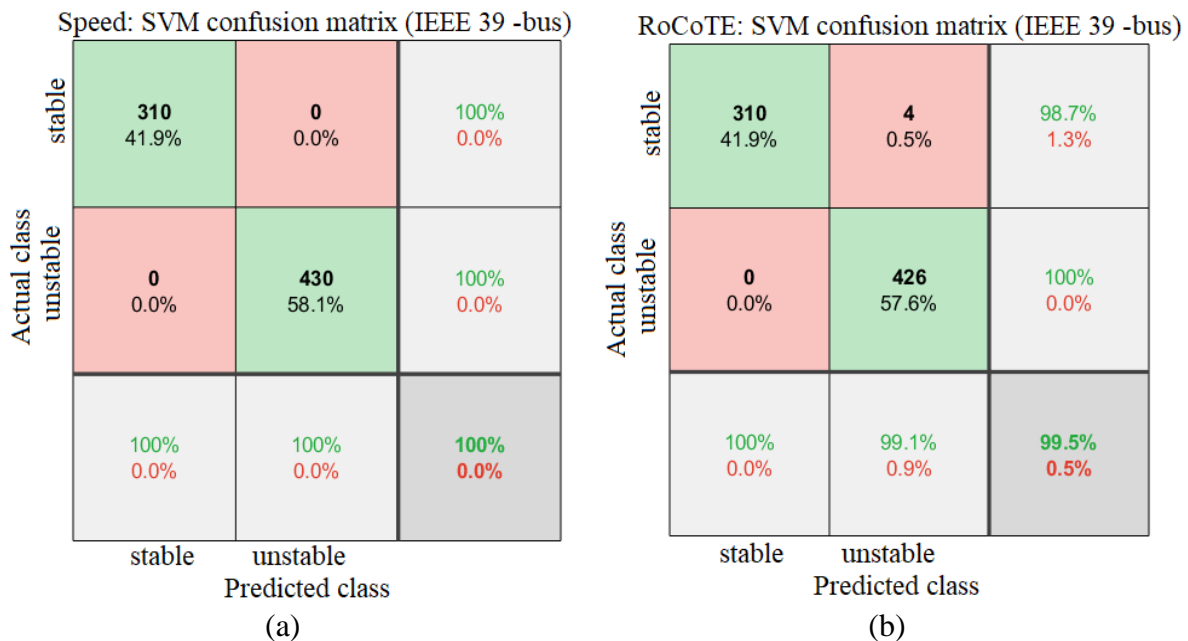


Figure 6: Matrice de confusion des vitesses et RoCoTE (Modèle: IEEE 39 -bus) [96].

En répétant la même expérience à partir des données issues du réseau IEEE 68 -bus, on constate que la SVM des vitesses des rotors détecte correctement 1261 signaux stables (sur 1264 réponses stables) et 812 signaux instables (sur 816 réponses instables), voir Figure 7(a). De même, la SVM des indices RoCoTE détecte correctement 1176 signaux

stables (sur 1264 réponses stables) et 754 signaux instables (sur 816 réponses instables), voir Figure 7(b).

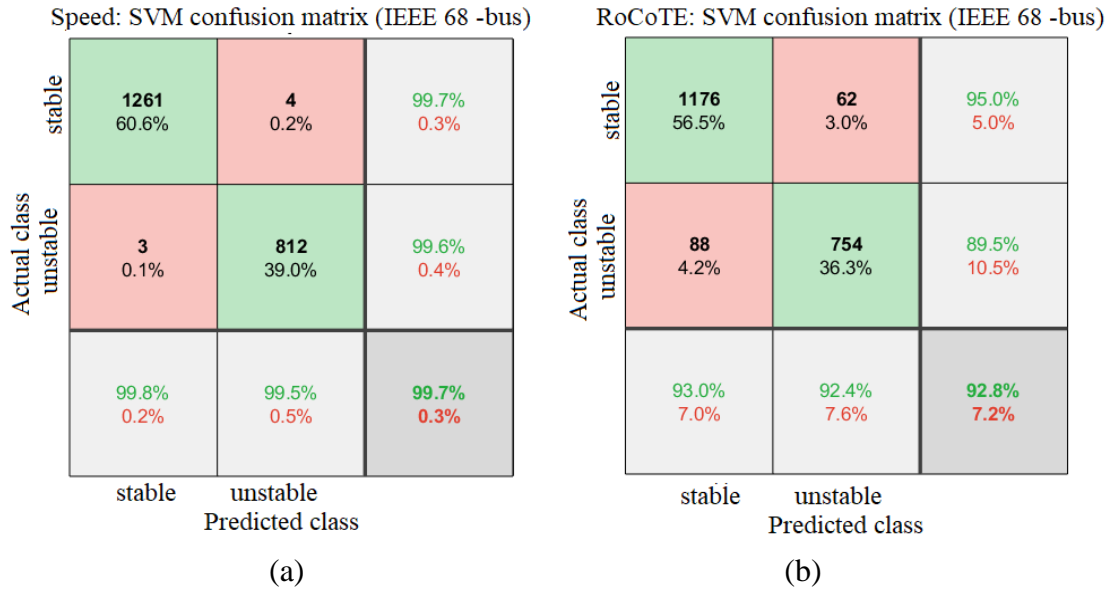
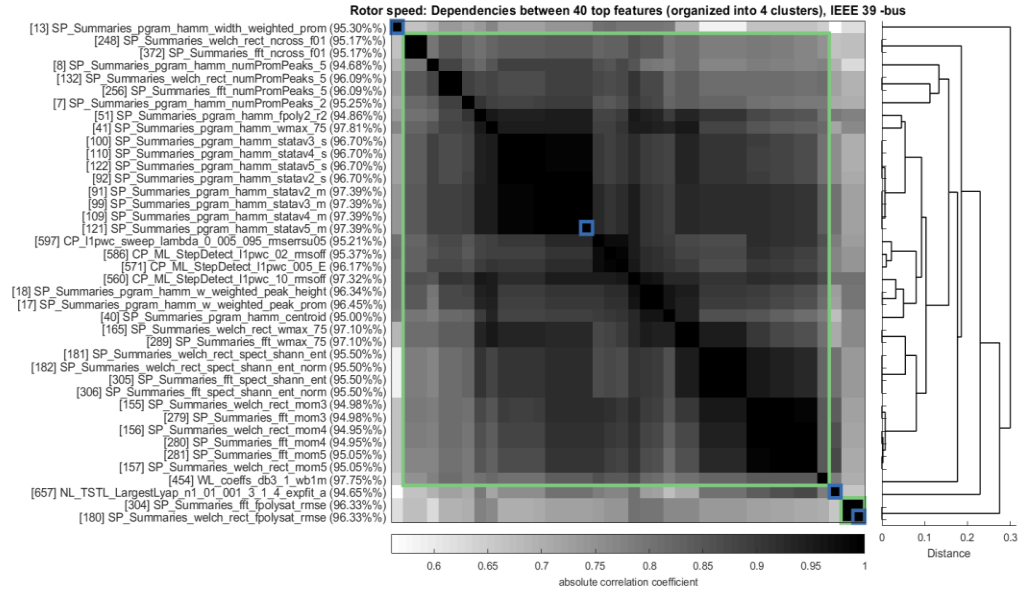
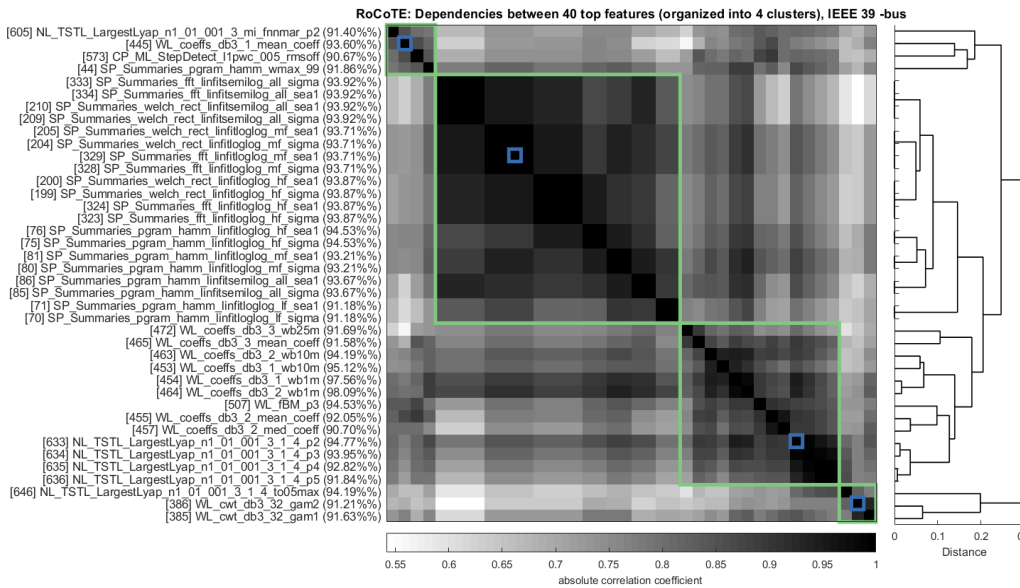


Figure 7: Matrice de confusion des vitesses et RoCoTE (Modèle: IEEE 68 -bus) [96].

Les Figures 8(a) & 8(b) présentent les 40 attributs les plus significatifs des bases de données des vitesses des rotors et des indices RoCoTE issues du réseau IEEE 39 -bus (obtenus en minimisant la redondance et la corrélation entre les attributs). On peut ainsi constater que, la transformée en ondelette de Daubechies (degré d'importance allant jusqu'à 97,75 %) et la transformée de Fourier rapide (degré d'importance allant jusqu'à 97,81 %) discriminent le mieux les vitesses des rotors par rapport aux exposants de Lyapunov (degré d'importance allant jusqu'à 94,65 %), voir Figure 8(a). De même, la transformée en ondelette de Daubechies (degré d'importance allant jusqu'à 98,09 %) et les exposants de Lyapunov (degré d'importance allant jusqu'à 94,77 %) caractérisent suffisamment les indices RoCoTE par rapport la transformée de Fourier rapide (degré d'importance allant jusqu'à 93,92 %), voir Figure 8(b). Pour ce cas d'étude, on peut en conclure que la transformée en ondelette permet de bien décrire la stabilité transitoire comparativement à la transformée de Fourier et les exposants de Lyapunov.



(a) : Les 40 meilleurs attributs issus de la base de données des vitesses des rotors.

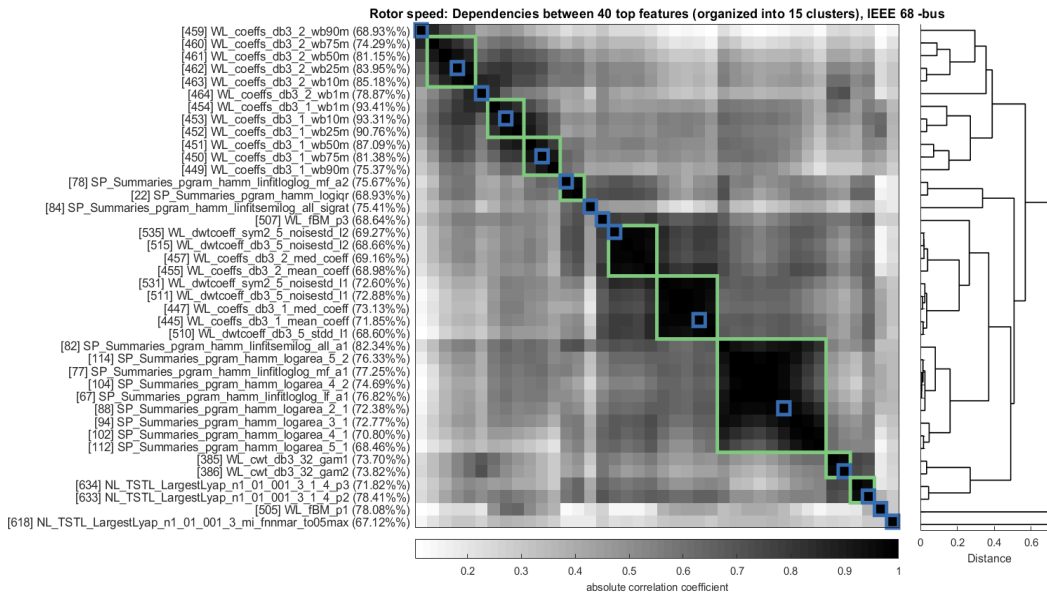


(b) : Les 40 meilleurs attributs issus de la base de données des RoCoTE.

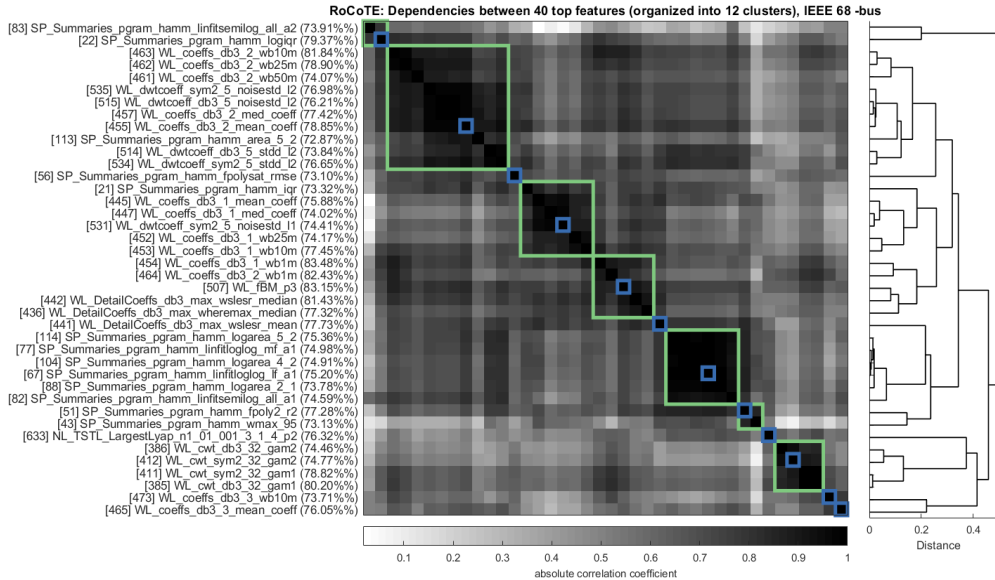
Figure 8: Hiérarchisation des attributs les plus significatifs (Modèle : IEEE 39 -bus) [96].

De manière similaire, en observant les données du réseau IEEE 68 -bus, on constate que la transformée en ondelette de Daubechies (degré d'importance allant jusqu'à 93,41 %) et la transformée de Fourier rapide (degré d'importance allant jusqu'à 82,34 %) définissent

bien les vitesses des rotors par rapport aux exposants de Lyapunov (degré d'importance allant jusqu'à 78,41 %), voir Figure 9(a). L'ondelette de Daubechies (degré d'importance allant jusqu'à 83,48 %) et la transformée de Fourier rapide (degré d'importance allant jusqu'à 79,37 %) discriminent mieux les indices RoCoTE par rapport, aux exposants de Lyapunov (degré d'importance allant jusqu'à 76,32 %) voir Figure 9(b).



(a) : Les 40 meilleurs attributs issus de la base de données des vitesses des rotors.



(b) : Les 40 meilleurs attributs issus de la base de données des RoCoTE.

Figure 9: Hiérarchisation des attributs les plus significatifs (Modèle : IEEE 68 -bus) [96].

2.4.3 Arbres de décision

L'arbre de décision est un outil intelligent qui utilise le nombre de divisions, la profondeur, le nombre minimum d'observations pour décrire la relation cachée existante entre l'ensemble des attributs et l'état de stabilité (stable ou instable) du réseau [106]. À titre d'illustration, malgré le nombre élevé des indicateurs, l'arbre de décision de la Figure 10 est assez simple voire trivial à comprendre, car il n'utilise que 3 variables parmi les +500 attributs pour obtenir une prédiction d'instabilité avec une fiabilité de 99 %.

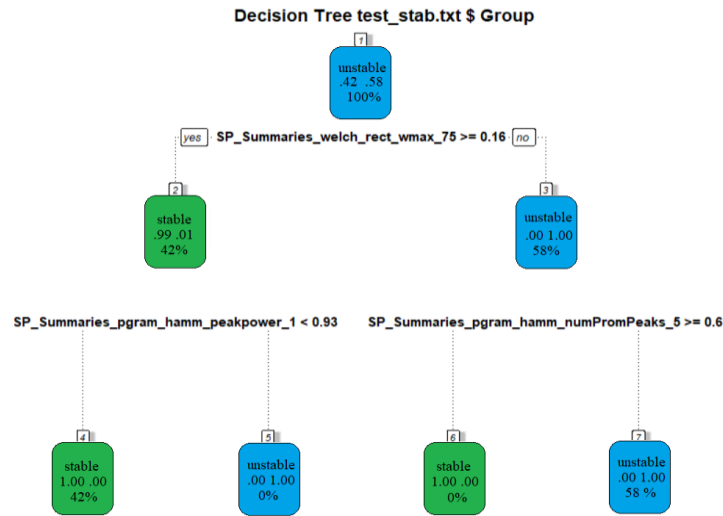


Figure 10: Arbres de décision issus des HDSI des vitesses (Modèle: IEEE 39 bus) [96].

Une question ouverte qui reste est de savoir quels attributs parmi les 500 extraits sont les plus pertinents en termes d'efficacité discriminante. La Figure 11 permet ainsi de mettre à l'évidence l'importance des attributs de stabilité en termes de *Gini impurity*, dans une forêt aléatoire [107]. Ces résultats sont très cohérents avec l'approche de clustering de la Figure 9, montrant que les attributs du domaine fréquentiel présentent les indices de sévérité décentralisés les plus efficaces [56]. En effet, le premier coefficient basé sur les exposants de Lyapunov est classé 61. Pour mettre en évidence l'importance critique de cette analyse, la Figure 12 présente la boîte à moustache (a) et la distribution (b) de l'une des variables du domaine à trois fréquences par rapport au meilleur exposant de Lyapunov. On observe ainsi que la distribution de Welch se chevauche et que les valeurs extrêmes correspondent à des cas instables existant dans la base de données jointe des réseaux IEEE 39+68 bus.

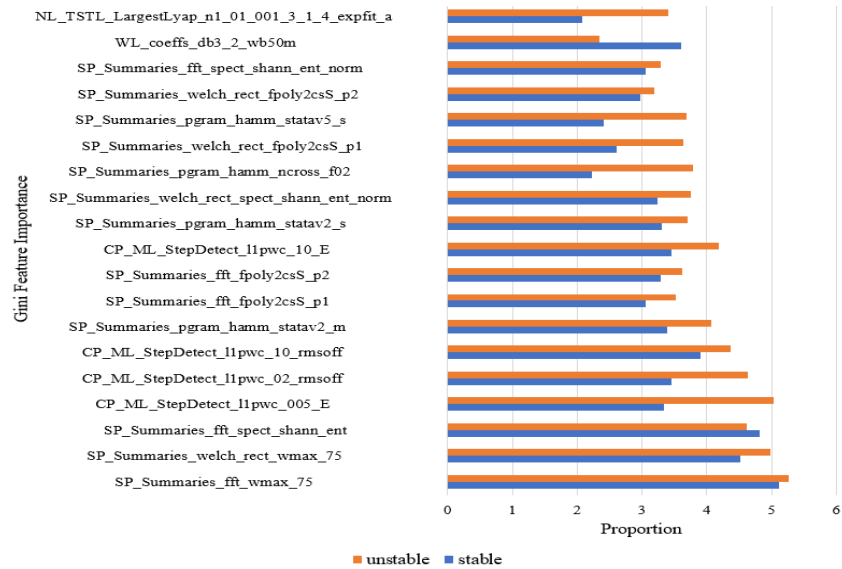


Figure 11 : Gini *impurity* des HDSI des vitesses des rotors (Modèle: IEEE 39 bus) [96].

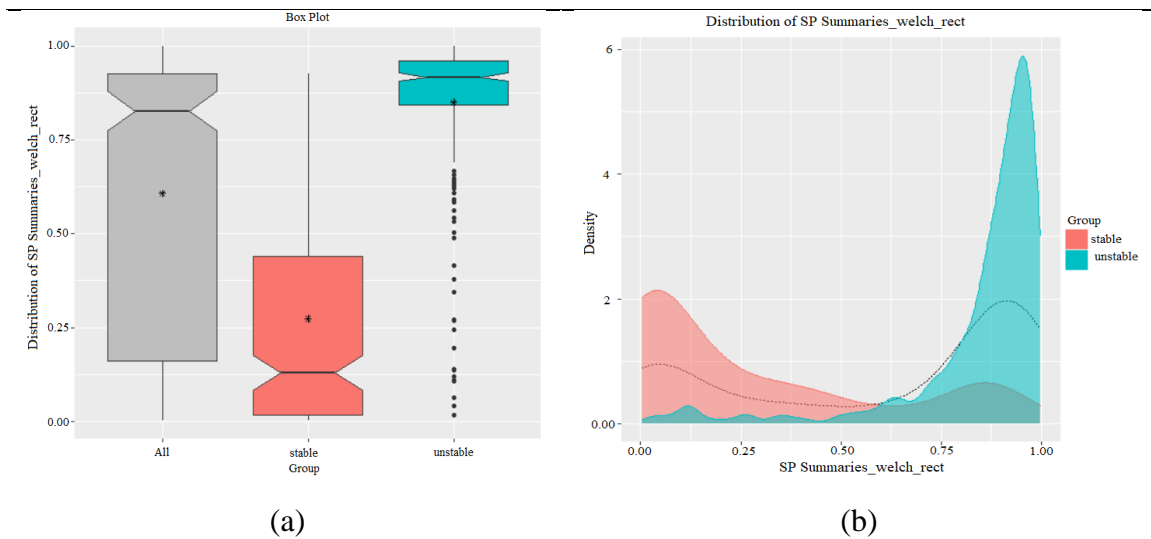


Figure 12 : Boîte à moustache et distribution spectrale d'un attribut de Welch à HDSI issu des vitesses des rotors (Modèle: IEEE 39+68 bus) [96].

2.4.4 Forêt aléatoire

Une forêt aléatoire est un ensemble (c'est-à-dire une collection) d'arbres de décision capable de séparer les signaux de stabilité présent dans la base de données d'apprentissage en deux groupes : stable ou instable. L'erreur de l'ensemble de l'arbre de décision doit être faible pour que le classificateur soit compétitif. Cela nécessite un nombre optimal d'arbres, une corrélation, et une force dans la forêt. De plus, la forêt doit être robuste à la variance

et aux biais existant entre les deux classes. La probabilité q ou $r=1-q$ d'avoir une bonne division des attributs stables u et instables v , s'exprime par [108]:

$$q_u \approx -\left(\frac{\Delta r}{\Delta u}\right)_{B,v} = -\frac{v!(u+v-1-f)!f}{(v-f)!(u+v)!} \quad (16)$$

$$q_v \approx -\left(\frac{\Delta r}{\Delta v}\right)_{B,u} = -\frac{(v-1)!(u+v-1-f)!uf}{(v-f)!(u+v-1)!(u+v)} \quad (17)$$

La force de la forêt (n_s) en tant que la probabilité que tous les nœuds d'au moins un arbre soit bien subdivisé s'exprime par:

$$n_s = \sum_{\tau=1}^B \binom{B}{\tau} (q^{N_{av}})^{\tau} (1-q^{N_{av}})^{B-\tau} = 1 - (1-q^{N_{av}})^B \quad (18)$$

Où : N_{av} et $q^{N_{av}}$ sont le nombre et la probabilité de bonnes divisions dans tous les nœuds. La Figure 13 présente l'erreur de reconstruction d'une forêt aléatoire à 150 arbres construite à partir des attributs de stabilité à haute dimension extraits de la base de données des vitesses des rotors du réseau IEEE 39 -bus. A partir de plus de 40 arbres, les erreurs de reconstruction des signaux stables et instable varient peu et restent inférieures à 0.0025.

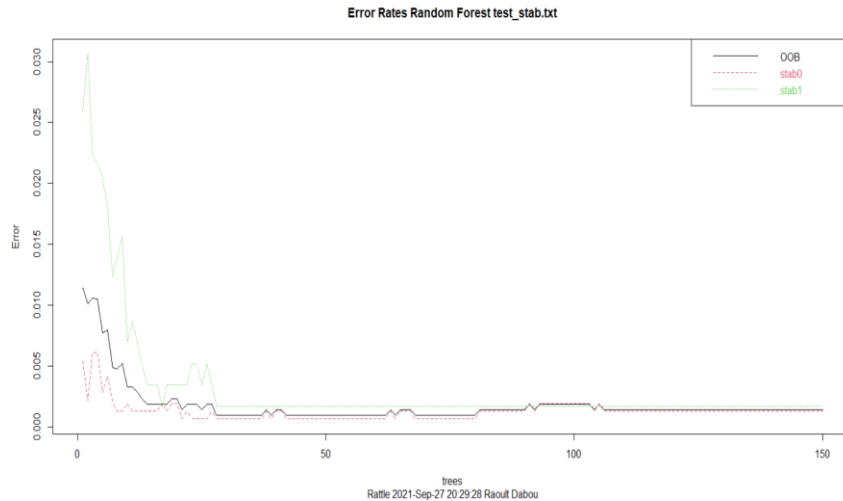


Figure 13: Erreur des forêts aléatoires issue des HDSI (Modèle : IEEE 39 -bus).

2.4.5 Réseaux de neurones artificiels

Un réseau de neurones est un groupe de nœuds interconnectés donc la fonction principale est d'optimiser le processus de prédiction de la stabilité des machines synchrones. À titre illustratif, la Figure 14 présente l'architecture de l'ANN utilisée pour la prédiction de l'état de stabilité des réseaux électriques à partir des indices HDSI extraits de la base de données des vitesses des rotors et des RoCoTE.

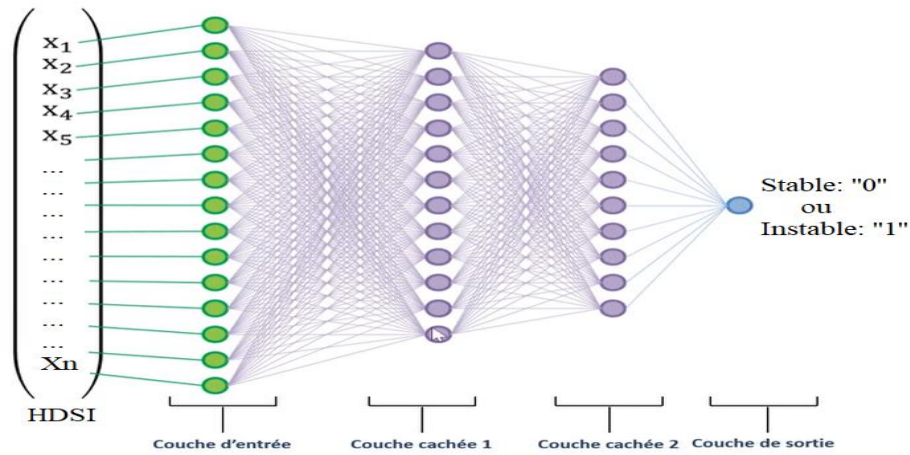


Figure 14 : Illustration d'une architecture de réseaux de neurones artificiels.

Pendant le processus d'entraînement du réseau de neurones, l'état cible (stable : "0" ou instable : "1") est exprimé en fonction de l'entrée x_i , du poids ω_i et du biais b_i , comme indiqué ci-dessous :

$$F_L(x_j) = \sum_{i=1}^L x_i \omega_i + b_i \quad (19)$$

L'erreur quadratique moyenne entre le point de données observé $\{(x_i, y_i)\}_{i=1}^J$ et la valeur du réseau neuronal est définie [109]:

$$\ell_u(x, \theta) = \frac{1}{N} \sum_{i=1}^J (y_i - \hat{u}(x_i, y_i))^2 \quad (20)$$

En observant la matrice de confusion de la Figure 15, on peut constater qu'à partir de 10 couches cachées (à l'exception des couches d'entrées et de la couche de sortie), le réseau de neurones artificiel implémenté en utilisant l'indice HDSI extraits des vitesses des rotors

du réseau IEEE 39 -bus, discrimine correctement la base de données test. La fonction d'activation de Heaviside, est utilisée pour définir l'état de stabilité ou le label du signal de la base de données.

```

Error matrix for the Neural Net model on test_stab.txt (counts):

      Predicted
Actual stab0 stab1 Error
stab0  2210    0    0
stab1    0   850    0

Error matrix for the Neural Net model on test_stab.txt (proportions):

      Predicted
Actual stab0 stab1 Error
stab0  72.2  0.0    0
stab1  0.0  27.8    0

Overall error: 0%, Averaged class error: 0%

Rattle timestamp: 2021-11-07 10:56:18 Raoult Dabou
=====

```

Figure 15: Matrice de confusion des RNNs issus des vitesses (Modèle : IEEE 39 -bus).

2.4.6 Adaptive boosting

L'algorithme AdaBoost se caractérise par une plus grande attention sur l'échantillon erroné lors de l'apprentissage, en ajustant la distribution de l'échantillon jusqu'à l'obtention d'une valeur spécifique acceptable [110]. Il utilise un classificateur faible $h_t(x)$ pour apprendre un classificateur fort $H(x)$ défini par :

$$H(x) = \text{sign}(\sum_{t=1}^T \alpha_t h_t(x)) \quad (21)$$

Où : sign est une fonction *signum*. Le poids α_t du classificateur est défini en fonction de l'erreur de classement ε_t par :

$$\alpha_t = \frac{1}{2} \ln \left(\frac{1-\varepsilon_t}{\varepsilon_t} \right) \quad (22)$$

La distribution de poids D_{t+1} dans le processus d'entraînement AdaBoost est calculée en fonction de l'initialisation $D_1 = \frac{1}{m}$ et de l'état cible par y_i :

$$D_{t+1}(x) = D_t(x) \times \begin{cases} \exp(-\alpha_t) & \text{if } h_t(x_i) = y_i \\ \exp(\alpha_t) & \text{if } h_t(x_i) \neq y_i \end{cases} \quad (23)$$

La Figure 16 présente l'évolution de l'erreur d'entraînement en fonction du nombre d'itérations de l'algorithme AdaBoost. L'utilisation d'un ensemble de 8 arbres (avec un taux d'apprentissage de 0.3, une profondeur de 6, une fonction objective binaire) permet de faire converger vers zero l'erreur de classification l'algorithme AdaBoost.

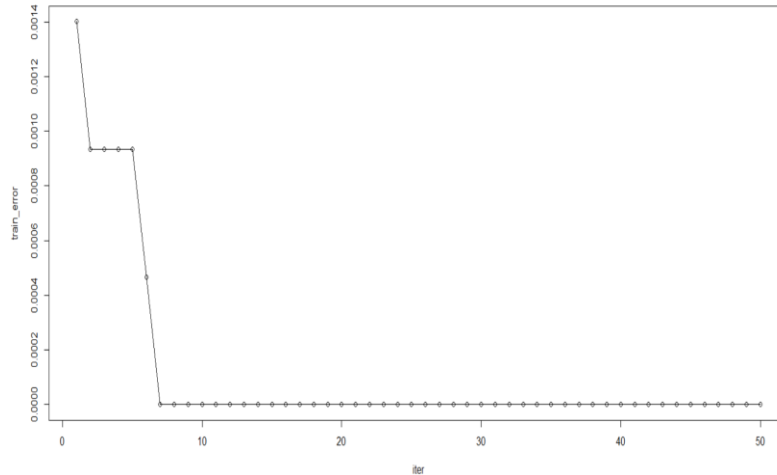


Figure 16 : Erreur du Adaptive Boosting issu des vitesses (Modèle : IEEE 39 -bus).

2.5 Évaluation des performances de classification

L'évaluation des performances de prédiction des algorithmes de machine learning (*support vector machine, decision trees, random forest, artificiel neural network, adaptive boosting*) est effectuée à partir de la matrice de confusion. Son architecture est présentée à la Tableau II.

Tableau II : Matrice de confusion pour DSP

Observation		Classe Prédite	
		<i>Positif == Stable</i>	<i>Négatif == Instable</i>
Classe actuelle	<i>Positif == Stable</i>	<i>TP</i>	<i>FN</i>
	<i>Négatif == Instable</i>	<i>FP</i>	<i>TN</i>

Où : TP (*True positif*) : signal stable prédit comme étant stable, TN (*True négatif*) : signal instable prédit comme étant instable, FP (*False positif*) : signal instable, prédit comme étant stable et FN (*False négatif*) : signal stable, prédit comme étant instable. Cette

matrice permet de calculer trois métriques à savoir : précision, la sécurité et la fiabilité, qui seront utilisées pour comparer les différents algorithmes de *machine learning*.

2.5.1 Précision

La précision définit la capacité de l'algorithme d'apprentissage à classer correctement les réponses des générateurs stables et instables de la base de données de test :

$$Accuracy (\%) = \frac{TP + TN}{TP + TN + FN + FP} \quad (24)$$

2.5.2 Fiabilité

La fiabilité définit la capacité qu'à l'algorithme de machine learning à classer correctement les réponses des générateurs instables de la base de données de test :

$$Reliability (\%) = \frac{TN - FN}{TN} \quad (25)$$

2.5.3 Sécurité

La sécurité définit la capacité qu'à l'algorithme de machine learning à classer correctement les réponses des générateurs stables de la base de données de test :

$$Security (\%) = \frac{TP - FP}{TP} \quad (26)$$

Le Tableau III présente les performances des algorithmes d'apprentissage obtenues à partir des vitesses des rotors et des indices RoCoTE des réseaux IEEE 39 -bus, IEEE 68 -bus et IEEE 39 +68 bus.

Tableau III : Performance des algorithmes de machine learning pour la DSP.

Test Systems	Software	Learning Algorithms	Signal processing	Accuracy (%)	Reliability (%)	Security (%)
IEEE 39 bus	Matlab	10- fold SVM	Speed	100	100	100
			RoCoTE	99.45	99.06	100
	Rattle	Random Forest	Speed	99.59	99.76	99.35
			RoCoTE	99.86	99.76	100
			Speed	99.72	100	99.35
AdaBoost	RoCoTE	99.86	99.76	100		
	Neural Network	Speed	100	100	100	
RoCoTE	99.72	99.53	100			
Decision Trees	Speed	98.91	99.06	98.69		
RoCoTE	98.64	98.34	99.02			
IEEE 68 bus	Matlab	10- fold SVM	Speed	99.66	99.50	99.76
			RoCoTE	92.78	91.77	92.51
	Rattle	Random Forest	Speed	99.90	99.75	100
			RoCoTE	98.60	97.61	99.20
			Speed	99.85	99.75	99.92
AdaBoost	RoCoTE	98.50	97.35	99.20		
	Neural Network	Speed	99.75	99.63	99.84	
RoCoTE	97.83	97.22	98.14			
Decision Trees	Speed	98.22	96.57	99.20		
RoCoTE	92.98	84.90	96.81			
IEEE 39 and 68-bus	Matlab	10- fold SVM	Speed	99.68	97.86	99.90
			RoCoTE	97.77	78.16	99.65
	Rattle	Random Forest	Speed	99.96	99.55	100
			RoCoTE	99.78	97.93	99.96
			Speed	99.96	99.55	100
AdaBoost	RoCoTE	99.81	98.81	99.94		
	Neural Network	Speed	99.92	99.92	99.94	
RoCoTE	99,33	95,21	99,72			
Decision Trees	Speed	98,46	83,64	99,71		
RoCoTE	97,41	60,70	99,96			

Le Tableau IV présente les *CPU time* d'extraction des attributs HDSI et le temps de réponse des algorithmes de *machine learning* (SVM, RF, DT, ANN et AdaBoost). Ces performances ont été évaluées sur un ordinateur DELL configuré grâce à un processeur Intel i7-7700HQ 4 cœurs à 2,80 GHz avec 16 Go de RAM. Bien que le *CPU Time* soit relativement important, le code réel est écrit sur MATLAB et par conséquent, il peut être rendu plus rapide à l'aide de la programmation C. De plus, puisque chaque série temporelle (ou sous-ensemble de séries temporelles) est analysée individuellement à l'aide de HCTSA, le problème du calcul de l'HDSI est naturellement parallélisable, en divisant par exemple les 10 machines présentes dans le réseau IEEE 39 bus en 10 tâches parallèles.

L'algorithme d'apprentissage AdaBoost offre le meilleur compromis entre la précision de la classification et le temps de calcul tandis que la forêt aléatoire et le SVM multiplient de manière exponentielle avec la taille du problème.

Tableau IV : CPU Time (in sec.) pour la DSP.

<i>N</i> ° of busses	HDSI	10 –fold SVM	RF	Boost	ANN	DT
IEEE 39 bus	325,17	5,28	9,80	12,95	55,31	7,45
IEEE 68 bus	880,57	14,04	70,36	34,82	113,09	13,50
IEEE 39 +68 bus	7738,67	460,01	494,77	54,28	120,94	29,67

2.6 Conclusion

Les méthodes existantes d'analyse et d'extraction d'attributs nécessitent généralement une expertise importante pour l'interprétation physique des signaux. Ce chapitre visait à développer un indice de stabilité à haute dimension, appelé HDSI, pour la surveillance et le contrôle de la stabilité à grande échelle des réseaux électriques. Cette nouvelle approche permet d'extraire et classer plus de 500 attributs décrivant chaque signal de réponse de générateur. Pour extraire les attributs de stabilité, la transformée de Fourier, la transformée en ondelettes, la méthode de périodogramme, la méthode Welch et les exposants de Lyapunov sont appliqués à chaque signal de réponse post-perturbation enregistrée lors de la simulation de défaut sur chacune des lignes des réseaux IEEE 39 -bus et IEEE 68 -bus. Ces attributs sont ensuite corrélés et normalisés de manière adjacente pour améliorer la visibilité et l'interprétation des caractéristiques cachées dans les signaux. Les attributs les plus discriminants sont regroupés en fonction de leurs pourcentages, puis classés en deux catégories de stabilité à l'aide du SVM. Les indices HDSI sont utilisés pour construire des modèles de prédiction sur Matlab et Rattle à l'aide des algorithmes d'apprentissage : SVM, DT, RF, ANN et AdaBoost. Tous les modèles se sont bien comportés, à l'exception du modèle DT standard dont la fiabilité n'est pas satisfaisante.

Chapitre 3 — Dictionnaire d'apprentissage joint pour la prédiction de la stabilité dynamique

3.1 Introduction

Ce chapitre introduit l'outil *supervised learning of overcomplete dictionary* (SLOD), utile pour analyser et prédire de la stabilité dynamique (DSP) dans les grands systèmes électriques. Pour y parvenir, plusieurs signaux post-contingence générés par des centaines de défauts (variation de charge, déclenchement des lignes, déclenchement des barres PV) ont été acquis par les unités des PMUs de tous les générateurs pour former l'ensemble de données d'apprentissage. Le développement de la SLOD s'effectue essentiellement à partir de la *Dynamic K-singular value decomposition* (D-KSVD) couplée à l'*orthogonal matching pursuit* (OMP). L'approche SLOD présente le mérite d'apprendre principalement un dictionnaire parcimonieux surcomplet joint à un classificateur. L'objectif de la SLOD est d'apprendre les indicateurs des signaux en ajoutant un terme discriminatif de l'état de stabilité (stable : "0" ou instable : "1") dans un problème d'optimisation. Cette formulation va permettre de discriminer les classes stables/instables de la base de données test. La SLOD proposée est appliquée sur de nouveaux scénarios (c'est-à-dire totalement inconnus du processus d'apprentissage) pour reconstruire et prédire la stabilité online des machines immédiatement après l'élimination des défauts.

3.2 Méthode de recherche des limites de stabilité

Les avancées des techniques d'acquisition des données ouvrent de nouvelles perspectives de développement dans l'évaluation de la stabilité dynamique basée sur la réponse des générateurs afin d'améliorer le temps de décision (après élimination des défauts) et de réduire les exigences de calcul sur des signaux tests [111]. La détection précise, la localisation temporelle et la classification des perturbations multiples en temps réel à l'aide de données de synchrophaseur permettent d'avoir une connaissance fiable et

sécurisée de l'état dynamique et des réseaux électriques [112-113]. L'apprentissage supervisé de dictionnaires surcomplets est une technique axée sur les données bien adaptée à la prédiction de stabilité dynamique, car elle nécessite peu de connaissances préalables du système ou de prétraitement des réponses des signaux entrants. La méthodologie SLOD pour la DSP a été développée à partir des données issues des réseaux IEEE 2 area 4 machines (S_1), IEEE 39 -bus (S_2), et IEEE 68 -bus (S_3). En raison de sa sévérité, plusieurs défauts triphasés terre sont simulés progressivement (par pas de 0,5 cycle) jusqu'à la limite critique sur chaque ligne et à proximité de chaque barre génératrice des trois réseaux tests (S_1, S_2, S_3). Pour chaque cas d'étude, la durée de simulation numérique est fixée à $t = 10s$ et le défaut est appliqué à $t = 1s$. Immédiatement, après chaque occurrence de défaut, les signaux de stabilité sont ré-échantillonnés (selon une fenêtre de décision cible de $125 ms$) et stockés progressivement dans la base de données d'apprentissage. L'organigramme SLOD présenté à la Figure 17 s'articule autour de deux étapes suivantes : 1) la génération et ré-échantillonnage des réponses des vitesses des rotors post-contingence et 2) entraînement et évaluation de la SLOD.

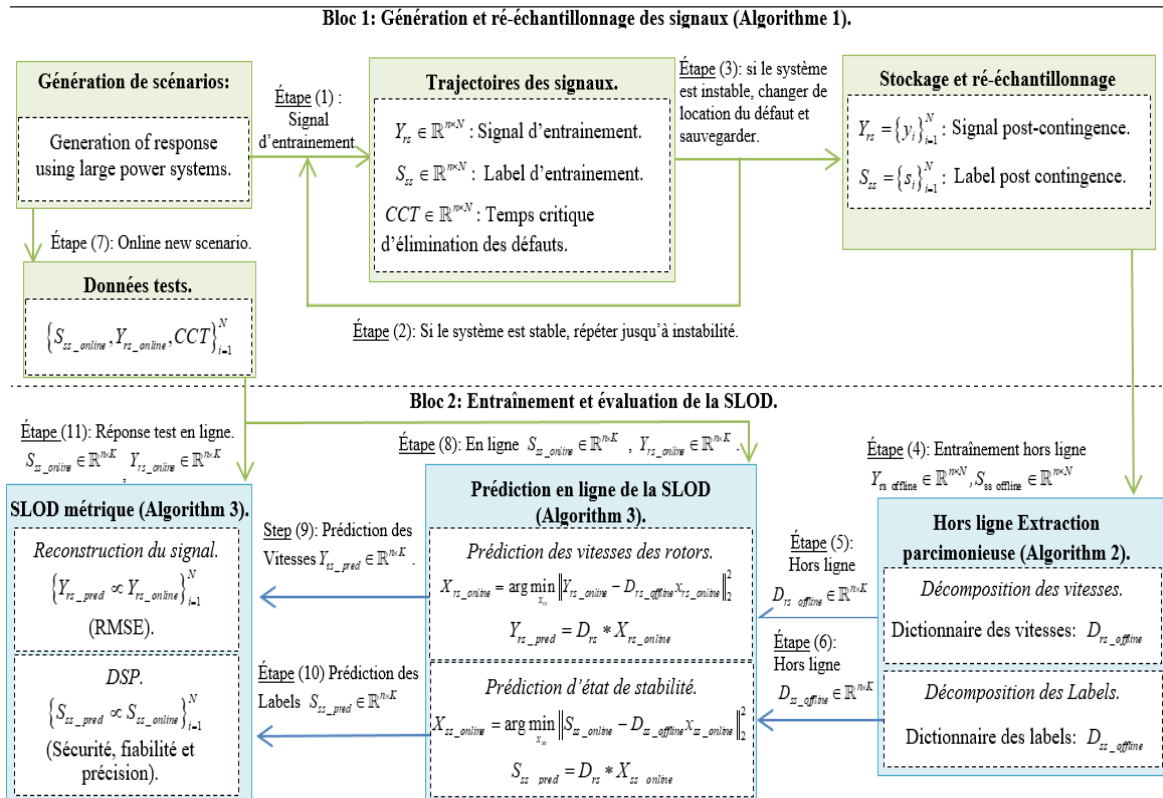


Figure 17: Organigramme pour l'apprentissage supervisé du dictionnaire surcomplet [40].

La simulation numérique de l'algorithme 1 permet de générer, de ré-échantillonner et de stocker de milliers des signaux des PMUs utiles à l'analyse prédictive de la stabilité. Pour chaque scénario de défaut x ($1 \leq x \leq n$), le solveur à pas variable évalue la dynamique des réseaux électriques sur une période spécifiée. L'ensemble de données d'apprentissage $T_{set} = \{y_i, s_i\}_{i=1}^n$ sauvegardé va permettre de concevoir un dictionnaire adaptatif joint à un classificateur. Pour les scénarios instables, la simulation numérique varie en fonction de la durée de perte synchrone. Pour diverses conditions de fonctionnement, seuls 75 échantillons (taux d'échantillonnage : 600 échantillons/s) par signal de vitesse de rotor sont utilisés pour la DSP. Chaque Y_{rs} échantillon est labellisé en fonction de l'état de stabilité $S_{ss} = \{''0'' \text{ ou } ''1''\}$ des réseaux électriques. Ce choix de troncature est motivé par le fait que la perturbation étant éliminée après les CT (critical time : $1+CT/60$ sec.) cycles, nous avons essentiellement réservé une fenêtre maximale de 125 ms pour la prise de décision après l'élimination du défaut. Le Tableau V présente la configuration des systèmes de test utilisés pour mettre en œuvre le SLOD.

Tableau V : Configuration des réseaux électriques utilisée pour l'approche SLOD

Réseaux électriques Tests	Nombre de ligne	Nombre de générateurs	Initial Bus Load (MW)		Load Change (LC)
			Low	High	Range (%)
2-area 4 machines	3	4	950	1775	
IEEE 39 -bus	30	10	217	629	10: 20 : 100
IEEE 68 -bus	66	16	9	6000	

L'algorithme 1 permet ainsi de générer et sauvegarder : 1) une base de données des vitesses des rotors, 2) une base de données des états de stabilité et 3) une base de données des temps critiques d'élimination des défauts.

Algorithm 1. Time-series Generation and Re-sampling

- 1. Input Training:** σ_c : standard deviation; μ_c : mean; $x_1 = CT / 60 + x$: sample obtained after clearing fault in a sec; $x_2 = x_1 + 0.125$: sample obtained over 125 ms after fault clearing in sec; the number of generators; β : number of simulations; e : simulation step; L : vector of loads; i : generator bus or line location; $\theta\%$: rate of load change.
 - 2. Output:** Y_{rs} : rotor speed and S_{ss} : stability status.
 - 3. Load switch:** $K = (L \pm \theta\% L)$
 - 4. For:** k in K
 - 5. For** $i = 1$: length (L)
 - 6. Setting up:** $CT_2 = \mu_c + \sigma_c$, $CT_1 = \mu_c - \sigma_c$, and e
 - 7. While** status = 0
 - 8. While** $(CT_2 - CT_1) > e$
$$CT = (CT_1 + CT_2) / 2$$
 - 9. Setting:** 1-, 2-, 3-Ph-G Fault and 3-Ph Breaker Block located on line (i) or generator bus (i).
 - 10. Activate:** a localized Fault and Breaker Block located on line (i) or generator bus (i).
 - 11. Run:** perform simulation of current contingency
 - 12. if** (status = unstable)
$$Y_{rs}(:, Ng \times (\beta - 1) + 1 : Ng \times \beta) = Y_{rs}(x_1 : x_2, 1 : Ng);$$
$$S_{ss}(:, Ng \times (\beta - 1) + 1 : Ng \times \beta) = S_{ss}(x_1 : x_2, 1 : Ng);$$
 - 13. Else:**
$$CT_2 = CT \quad // \text{CT: Critical Time}$$
$$Y_{rs}(:, 1 : Ng \times \beta) = Y_{rs}(x_1 : x_2, 1 : Ng);$$
$$S_{ss}(:, 1 : Ng \times \beta) = S_{ss}(x_1 : x_2, 1 : Ng);$$
 - 14. End;**
 $k = k + 1; \beta = \beta + 1;$
 - 15. End;**
 - 16. End;**
 - 17. End;**
 - 18. End;**
-

La Figure 18 présente un exemple des réponses des vitesses des rotors instables enregistrées lors de la recherche du temps critique d'élimination des défauts sur la barre 1 proche du générateur 1 (avec une commutation de 10 % de la charge la plus faible) du réseau IEEE 2 area 4 machines.

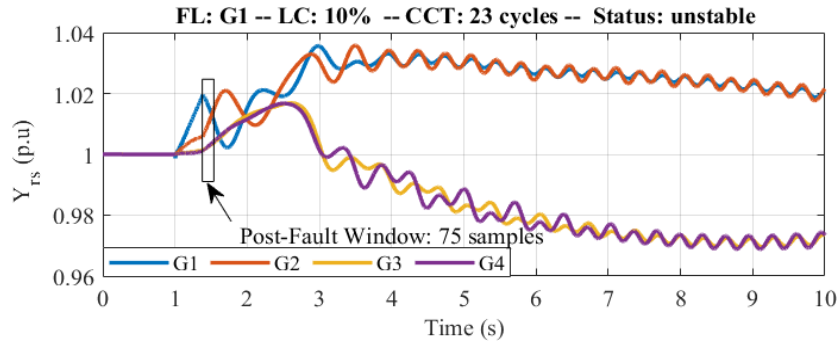


Figure 18 : Allure des vitesses des rotors instables du IEEE 2 area 4 machines [40].

La Figure 19 illustre un exemple des trajectoires des vitesses des rotors instables enregistrées lors de la recherche du temps critique d'élimination des défauts sur la barre 1 située proche du générateur 1 (avec une commutation de 10 % de la charge la plus élevée) du IEEE 39 bus.

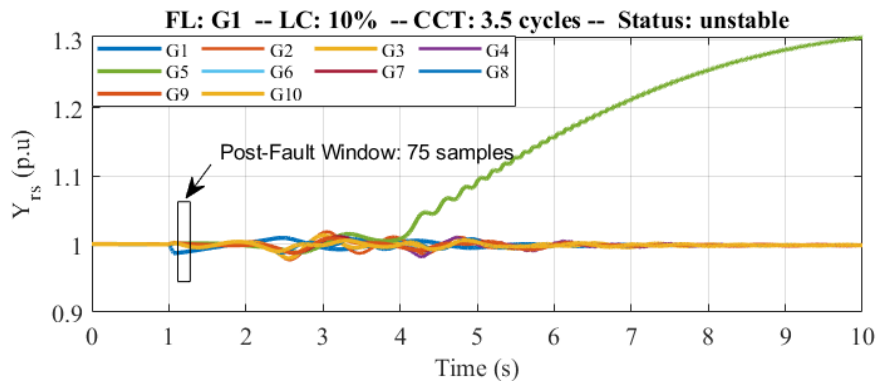


Figure 19 : Allure des vitesses des rotors instables du IEEE 39 bus [40].

La Figure 20 représente un exemple des seize profils des vitesses des rotors instables enregistrées pendant la recherche du temps critique d'élimination des défauts sur la ligne 1 du réseau IEEE 68 -bus.

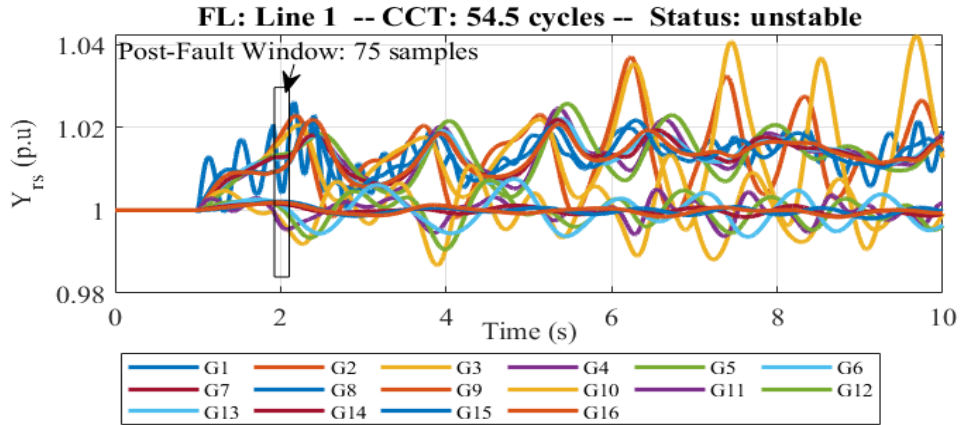
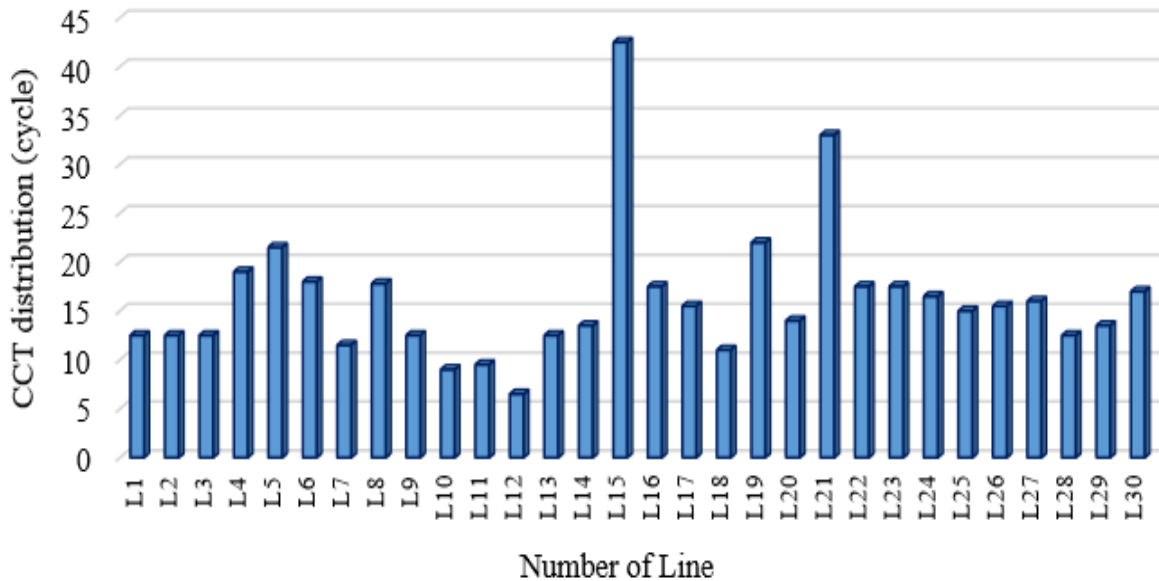
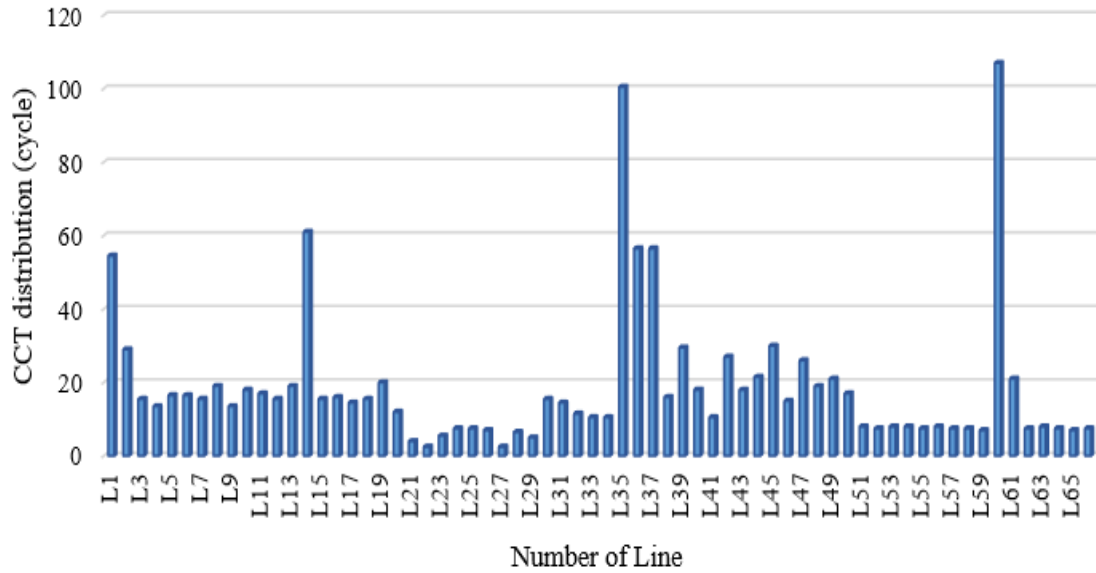


Figure 20 : Allure des vitesses des rotors instables du IEEE 68 -bus [40].

La Figure 21 illustre l’histogramme de sensibilité obtenu en simulant les défauts triphasés à la terre sur chaque ligne des réseaux IEEE 39 -bus et IEEE 68 -bus. Ce diagramme présente les lignes les plus sensibles (voir L6 sur la Figure 21(a) et L27 sur la Figure 21(b)) et résilientes (voir L15 sur la Figure 21(a) et L60 sur la Figure 21(b)) aux défauts triphasés terre.



(a): Sensibilité des lignes du réseau IEEE 39 -bus.



(b): Sensibilité des lignes du réseau IEEE 68 -bus.

Figure 21: Histogrammes des réseaux IEEE 39, -68 bus face aux défauts triphasés terre [40].

3.3 Développement d'un dictionnaire joint au classificateur

Les scénarios de séries chronologiques générés et ré échantillonnés ont été séparés en un ensemble d'entraînement pour l'apprentissage (70 %) et un ensemble test (30 %) pour l'évaluation des performances, voir le Tableau VI.

Tableau VI : Structure des données d'apprentissage de la SLOD.

Test systems	Machine Learning	Offline Training (Input size)	Features Extraction (Input size)	Online Testing (Input size)
IEEE 2-area 4 machines	Proposed SLOD	Rotor speed (75×756)	ASFI (75×300)	Rotor speed (75×324)
	SVM, DT, RF, ANN, AdaBoost CNN & LSTM	Rotor speed (75×756)	HDSI (507×756)	HDSI (512×324)
IEEE 39-bus	Proposed SLOD	Rotor speed (75×5510)	ASFI (75×2325)	Rotor speed (75×2361)
	SVM, DT, RF, ANN, AdaBoost CNN & LSTM	Rotor speed (75×5510)	HDSI (491×5510)	HDSI (464×2361)
IEEE 68 -bus	Proposed SLOD	Rotor speed (75×20216)	ASFI (75×8700)	Rotor speed (75×8664)
	SVM, DT, RF, ANN, AdaBoost CNN & LSTM	Rotor speed (75×20216)	HDSI (470×20216)	HDSI (472×8664)

3.3.1 Extraction des attributs parcimonieux pour la DSP

Considérons $Y_{rs} \in \mathbb{R}^{n \times N}$ (N représente le nombre de signaux enregistrés après la perturbation) utilisés pour la DSP, le dictionnaire d'atomes parcimonieux $D_{rs} \in \mathbb{R}^{n \times K}$ ($K \succ n$, nombre de colonnes supérieur au nombre de lignes) est développé en résolvant le problème non déterministe (28) :

$$\langle D_{rs}, X_{rs} \rangle = \arg \min_{D_{rs}, X_{rs}} \left(\|Y_{rs} - D_{rs} X_{rs}\|_2^2 \right) \quad \text{subject to: } \forall_i \|x_{irs}\|_0 \leq T \quad (27)$$

Où : $X_{rs} \in \mathbb{R}^{K \times N}$, $\|Y_{rs} - D_{rs} X_{rs}\|_2^2 \in \mathbb{R}^{n \times N}$, T et $\|\cdot\|_0$ désignent respectivement le support parcimonieux, l'erreur de reconstruction, la contrainte de parcimonie et le nombre d'élément non nul dans le vecteur. Le dictionnaire D_{rs} est implémenté à partir de la *K-singular value decomposition* (K-SVD) couplée à l'*orthogonal matching pursuit* (OMP) qui est un algorithme introduit par [36] pour l'apprentissage des structures discriminantes.

L'extraction linéaire des attributs par la K-SVD et l'OMP génère une infinité de solutions, qui n'ont pas toujours de signification physique. Pour déterminer l'unique solution optimale pour la *dynamic stability prediction* (DSP), il est nécessaire de définir et d'intégrer un paramètre de régularisation [28] dans la fonction objective. Ce terme de régularisation $J_{rs}(x_{rs})$ permet ainsi de guider la décomposition parcimonieuse vers l'unique jeu d'atomes physiquement interprétable. Celui-ci est défini par l'Eq (29) :

$$\min_{x_{rs}} [J_{rs}(x_{rs})] = \min_{x_{rs}} \left[\frac{1}{2} \|B_{rs} x_{rs}\|_2^2 \right] \quad \text{subject to: } Y_{rs} = D_{rs} X_{rs} \quad (28)$$

Où : $B_{rs} \in \mathbb{R}^{n \times K}$ est la matrice qui garantit l'unicité des indicateurs parcimonieux extraits.

La solution mathématique de l'Eq (29) est définie sous la contrainte λ_{rs} caractérisée par le

Lagrangien $L(x_{rs}, \lambda_{rs})$:

$$L(x_{rs}, \lambda_{rs}) = \frac{1}{2} \|B_{rs} x_{rs}\|_2^2 + \lambda_{rs}^T (D_{rs} x_{rs} - Y_{rs}) \quad (29)$$

En annulant le gradient de l'Eq (30), nous obtenons plusieurs supports parcimonieux transitoires définis par l'Eq (32) :

$$\nabla L(x_{rs}, \lambda_{rs}) = B_{rs}^T B_{rs} x_{rs} + D_{rs}^T \lambda_{rs} = 0 \quad (30)$$

$$x_{rs} = -\left(B_{rs}^T B_{rs}\right)^{-1} D_{rs}^T \lambda_{rs} \quad (31)$$

$$Y_{rs} = D_{rs} x_{rs} = -D_{rs} \left(B_{rs}^T B_{rs}\right)^{-1} D_{rs}^T \lambda_{rs} \quad (32)$$

La contrainte λ_{rs} permettant d'extraire les indicateurs parcimonieux uniques de la base de données d'entraînement est définie par :

$$\lambda_{rs} = -\left(D_{rs} \left(B_{rs}^T B_{rs}\right)^{-1} D_{rs}^T\right)^{-1} Y_{rs} \quad (33)$$

En raison de λ_{rs} , l'unique support parcimonieux optimal X_{opt} capable d'extraire les attributs de stabilité immédiatement après l'élimination des défauts est défini par [28] :

$$X_{opt} = \left(B_{rs}^T B_{rs}\right)^{-1} D_{rs}^T \left(D_{rs} \left(B_{rs}^T B_{rs}\right)^{-1} D_{rs}^T\right)^{-1} Y_{rs} \quad (34)$$

L'algorithme 2 résume l'approche SLOD développée pour extraire les indices de stabilité parcimonieux de la base de données des signaux des vitesses des rotors.

Algorithme 2 : Attributs adaptatifs parcimonieux.

1. Input : $Y_{rs} \in \mathbb{R}^{n \times N}$: rotor speed ; $S_{ss} \in \mathbb{R}^{n \times N}$: stability status; $D_{0_{rs}}$: Initial rotor speed; $D_{0_{ss}}$: stability status dictionaries; T : sparsity index; L : number of iterations.

2. Output: D_{rs} : learned rotor speed dictionary; D_{ss} : learned stability status dictionary
 X_{rs_opt} : optimal learned rotor speed sparse coding; X_{ss_opt} : optimal learned stability status sparse coding.

3. Initialization: $D_{rs} = D_{0_{rs}}$; $D_{ss} = D_{0_{ss}}$;

4. for $n = 1, \dots, L$ **do**

$$\forall i: \begin{cases} x_{irs} = \arg \min_{x_{rs}} \|Y_{irs} - D_{rs} x_{rs}\|_2^2 \text{ subject to } \|x_{rs}\|_0 \leq T \\ x_{iss} = \arg \min_{x_{ss}} \|S_{iss} - D_{ss} x_{ss}\|_2^2 \text{ subject to } \|x_{ss}\|_0 \leq T \end{cases}$$

5. for $j = 1, \dots, K$ **do**

$$d_{rj} = 0; d_{sj} = 0;$$

6. Decomposition Error:

$$E_{rs} = Y_{rs} - \sum_{l \neq j} d_{rl} x_{rs}^l ; E_{ss} = S_{ss} - \sum_{l \neq j} d_{sl} x_{ss}^l ;$$

7. Update Dictionaries :

$$\{d_{rs}, x_{rs}\} = \arg \min_{d_{rs}, x_{rs}} \|E_{rs} - d_{rj} x_{rs}^j\|_2^2, \text{ subject to } \|d_{rs}\|_2 = 1$$

$$\{d_{ss}, x_{ss}\} = \arg \min_{d_{ss}, x_{ss}} \|E_{ss} - d_{sj} x_{ss}^j\|_2^2, \text{ subject to } \|d_{ss}\|_2 = 1$$

$$E_{rs} = U_{rs} \Gamma_{rs} V_{rs}^T ; E_{ss} = U_{ss} \Gamma_{ss} V_{ss}^T ;$$

$$d_{rj} = d_{rs} ; d_{sj} = d_{ss} ; X_{rj} = x_{rs}^T ; X_{sj} = x_{ss}^T ;$$

8. End ($j = 1 : K$) ;

9. End ($n = 1 : L$) ;

Où : V , U , Γ , $d_{rs} = u_1$ et $x_{rs} = \sigma_1 v_1$ désignent respectivement les vecteurs propres de droites, les vecteurs propres de gauches, les valeurs singulières, les premiers vecteurs propres gauches et non nuls de la k^{th} ligne dans X_{rs} . Cette méthode itérative et gloutonne consomme un temps de calcul important, pour extraire les attributs de stabilité parcimonieux. Toutefois, elle présente une performance (précision, fiabilité et sécurité) satisfaisante sur la base de données test.

3.3.2 Reconstruction et classification des réponses des générateurs

Le dictionnaire d'atomes des vitesses des rotors D_{rs} développé à partir des signaux de la base de données d'entraînement est utilisé pour reconstruire et prédire la stabilité transitoire des nouveaux scénarios, immédiatement après l'élimination du défaut. Ainsi, à partir du support X_{rs_online} d'un nouveau scénario et du dictionnaire appris D_{rs} , il est possible de reconstruire signal original, Eq (36).

$$Y_{rs_pred} \simeq \sum d_{rs} x_{rs_online} \simeq D_{rs} X_{rs_online} \quad (35)$$

Le support X_{ss_online} est extrait en utilisant le dictionnaire supervisé D_{ss} des états de stabilité appris de la base de données d'entraînement et ayant pour label les signaux des vitesses des rotors.

$$\langle D_{ss}, X_{ss_online} \rangle = \arg \min_{D_{ss}, X_{ss_online}} \left(\|S_{ss_online} - D_{ss} X_{ss_online}\|_2^2 \right) \quad \text{subject to: } \forall_i \|x_{iss_online}\|_0 \leq T \quad (36)$$

Où : $\|S_{ss_online} - D_{ss} X_{ss_online}\|_2^2 \in \mathbb{R}^{n \times N}$ et T désignent l'erreur quadratique moyenne (RMSE) et la contrainte de parcimonie. Le support X_{ss_online} de l'état de stabilité est obtenu en résolvant :

$$x_{iss_online}^k = \arg \min_{x_{ss}} \|S_{ss_online} - D_{ss}^{k-1} x_{ss_online}\|_2^2$$

$$\text{subject to } \|x_{iss_online}\|_0 \leq T \text{ for } i = 1, 2, \dots, N \quad (37)$$

En utilisant la même procédure de sélection définie des Eq (29) à (35), la prédiction de stabilité transitoire S_{ss_pred} est calculée à partir du support extrait X_{ss_online} et du dictionnaire des vitesses des rotors D_{rs} appris de la base de données d'entraînement.

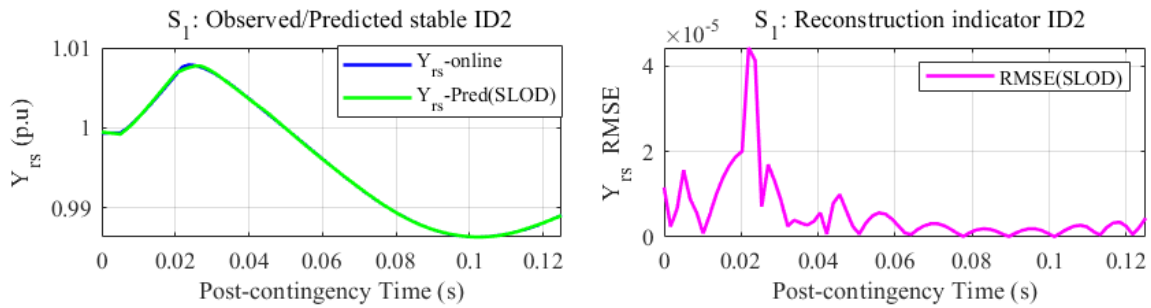
$$S_{ss_pred} \simeq \sum d_{rs} x_{ss_online} \simeq D_{rs} X_{ss_online} \quad (38)$$

Étant donné que la classification de stabilité est fortement déséquilibrée [114], des métriques (telles que la précision, la fiabilité, la sécurité et le RMSE) sont utilisées pour évaluer les performances de généralisation du SLOD immédiatement après l'occurrence d'une perturbation. L'algorithme 3 permet de décrire le processus de calcul des paramètres de la matrice de confusion utiles pour la prédiction de nouveau scénario.

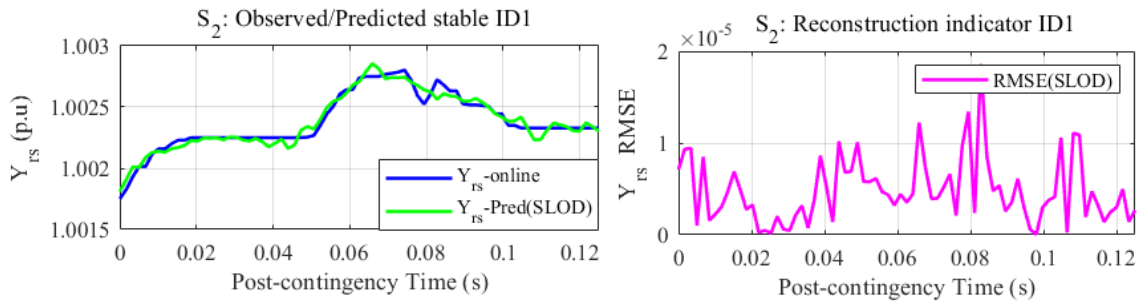
Algorithme 3: Évaluation des métriques pour la DSP.

- 1. Input:** S_{ss_online} : online stability status; D_{rs} : supervised rotor speed dictionary;
 S_{ss_pred} : predicted stability status.
 - 2. Output:** TP : True positive; TN : True negative;
 FP : True positive; FN : True negative.
 - 3. For** $i = 1 : size(S_{ss_online}, 1)$
 - 4. For** $j = 1 : size(S_{ss_pred}, 2)$
 - 5. if** $\{S_{ss_online}(i, j) = 1 \ \& \ S_{ss_pred}(i, j) \simeq 1\}$
 $TP = TP + 1$
 - 6. Else if** $\{S_{ss_online}(i, j) = 0 \ \& \ S_{ss_pred}(i, j) \simeq 0\}$
 $TN = TN + 1$
 - 7. Else if** $\{S_{ss_online}(i, j) = 0 \ \& \ S_{ss_pred}(i, j) \simeq 1\}$
 $FP = FP + 1$
 - 8. Else if** $\{S_{ss_online}(i, j) = 1 \ \& \ S_{ss_pred}(i, j) \simeq 0\}$
 $FN = FN + 1$
 - 9. End; End; End;**
-

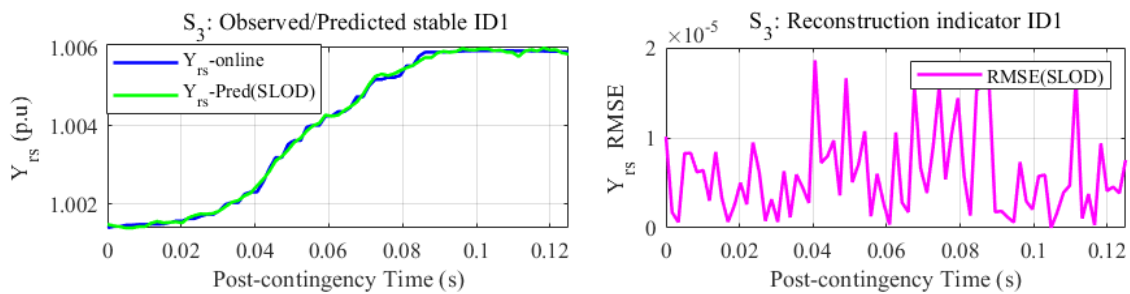
À titre illustratif, les dictionnaires d'atomes extraits de la base de données d'entraînement des bases de données d'entraînement S_1 , S_2 et S_3 sont utilisés pour reconstruire les vitesses des rotors et par conséquent analyser la stabilité des réponses tests. Afin de bien observer ces performances de reconstruction, les RMSE entre la prédiction Y_{rs_pred} et la réponse observée Y_{rs_online} . Les Figures 22(a), 22(b) & 22(c) présentent les vitesses des rotors stables observées (en bleu) /prédites (en vert) et les RMSE (en magenta) associées, des systèmes S_1 (a), S_2 (b) et S_3 (c).



(a) Décomposition d'une vitesse de rotor stable issue du système S_1 .



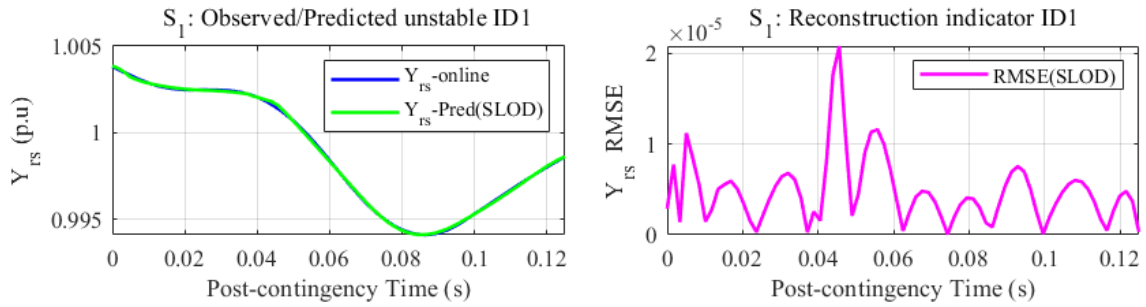
(b) Décomposition d'une vitesse de rotor stable issue du système S_2 .



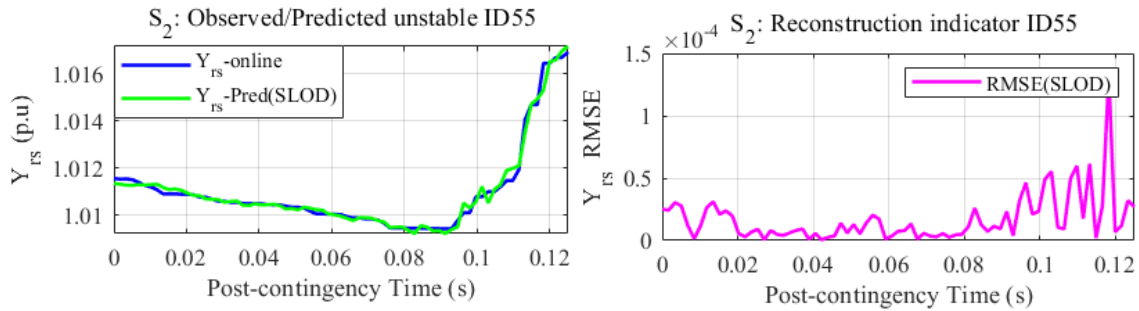
(c) Décomposition d'une vitesse de rotor stable issue du système S_3 .

Figure 22: Décompositions sur des dictionnaires stables et les RMSE associées [40].

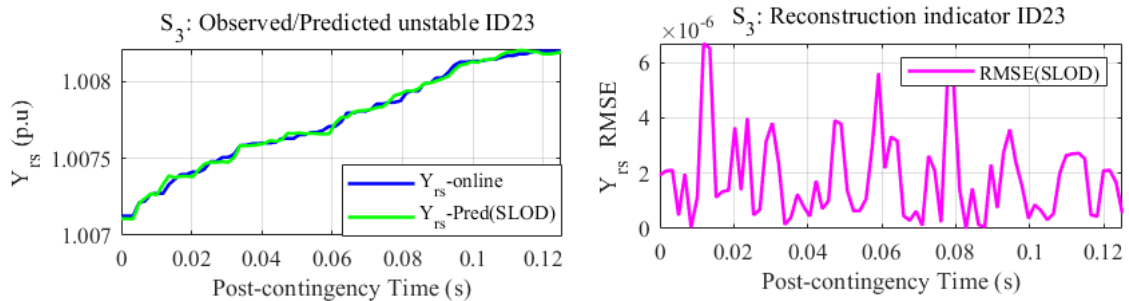
De même, les dictionnaires d'atomes extraits de la base de données d'entraînement S_1 , S_2 et S_3 sont utilisés pour reconstruire les vitesses des rotors instables. Comme le montrent les Figures 23(a), 23(b) & 23(c) les vitesses des rotors instables observées (en bleu $Y_{rs} Test$)/prédites (en vert $Y_{rs} SLOD$) et les RMSE associées (en magenta) aux systèmes S_1 (a), S_2 (b) et S_3 (c) pour la reconstruction des réponses des machines.



(a) Décomposition d'une vitesse de rotor instable issue du système S_1 .



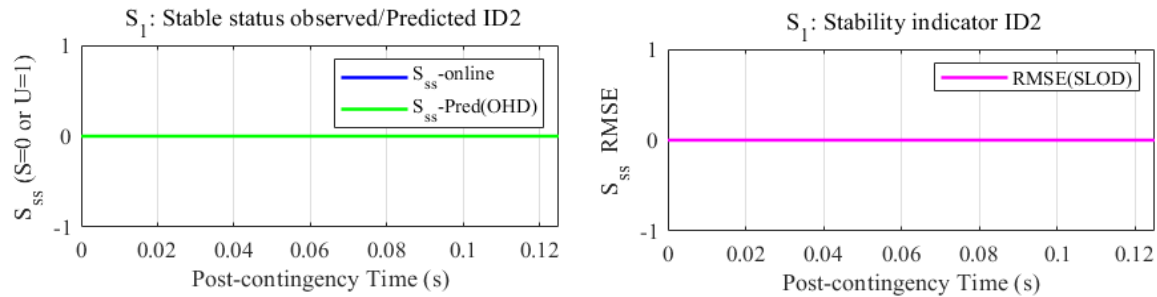
(b) Décomposition d'une vitesse de rotor instable issue du système S_2 .



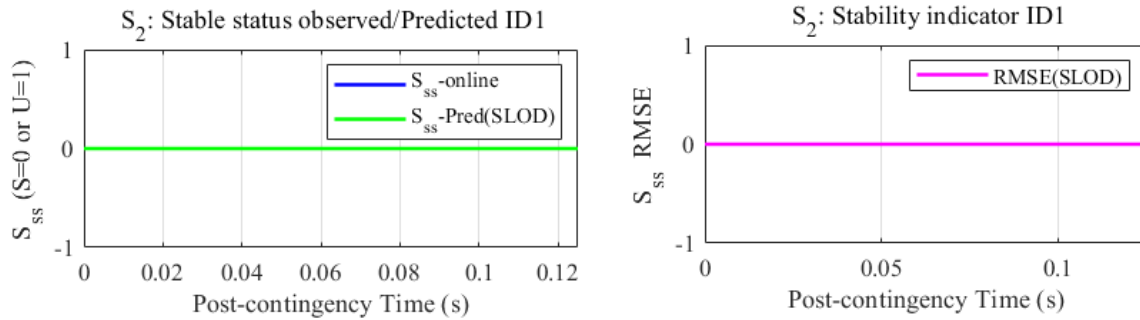
(c) Décomposition d'une vitesse de rotor instable issue du système S_3 .

Figure 23: Décompositions sur des dictionnaires instables et les RMSE associées (en magenta) [40].

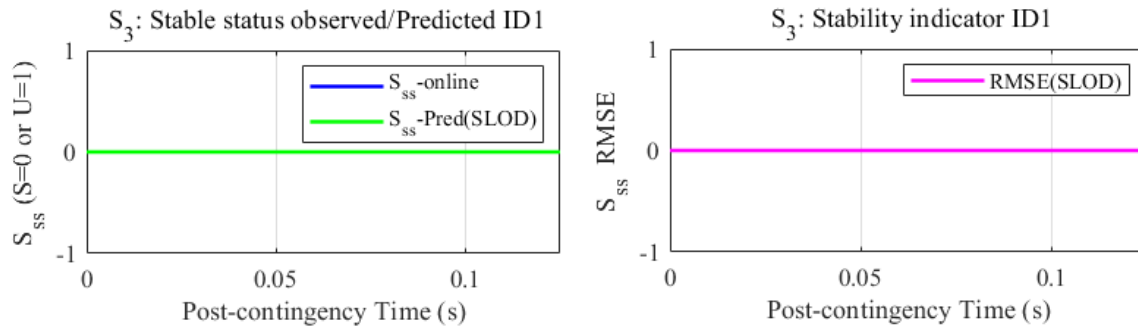
Ces caractéristiques parcimonieuses adaptatives extraites des bases de données d'entraînement permettent de reconstruire les signaux des vitesses des rotors stables et instables avec un taux d'erreur $\sim 10^{-4}$. Les travaux de [74] ont prouvé que les signaux de même classe (stable ou instable) ont un support parcimonieux similaire X_{rs} ou X_{ss} . En se basant sur ce principe, les supports parcimonieux de la base de données des états de stabilité X_{ss} sont extraits et projetés sur le dictionnaire des vitesses des rotors D_{rs} .



(a) Classification de l'état de stabilité d'une réponse stable issu du système S_1 .



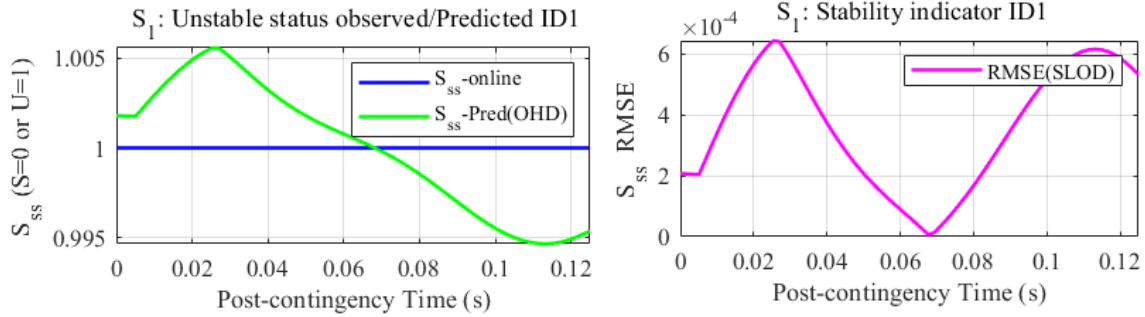
(b) Classification de l'état de stabilité d'une réponse stable issu du système S_2 .



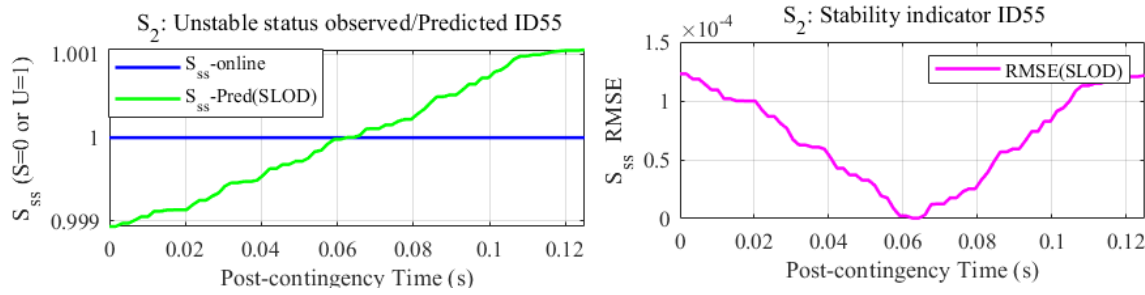
(c) Classification de l'état de stabilité d'une réponse stable issu du système S_3 .

Figure 24: Classifications sur des dictionnaires stables et les RMSE associées (en magenta) [40].

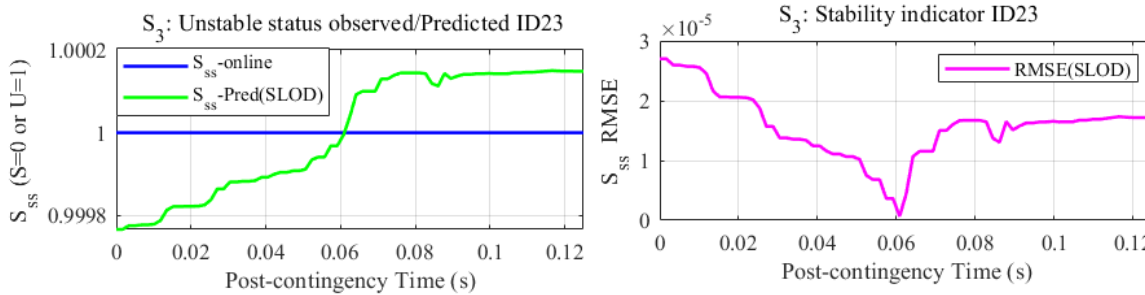
Les Figures 24(a), 24(b) & 24(c) présentent les états de stabilité stable observés (en bleu $Y_{rs} Test$)/prédites (en vert $Y_{rs} SLOD$) et les RMSE associées (en magenta) aux systèmes S_1 (a), S_2 (b) et S_3 (c). Les Figures 25(a), 25(b) & 25(c) présentent les états de stabilité instable observés (bleu)/prédits (vert) et les RMSE associées (en magenta) aux systèmes S_1 (a), S_2 (b) et S_3 (c).



(a) Classification de l'état de stabilité d'une réponse instable issu du système S_1 .



(b) Classification de l'état de stabilité d'une réponse instable issu du système S_2 .



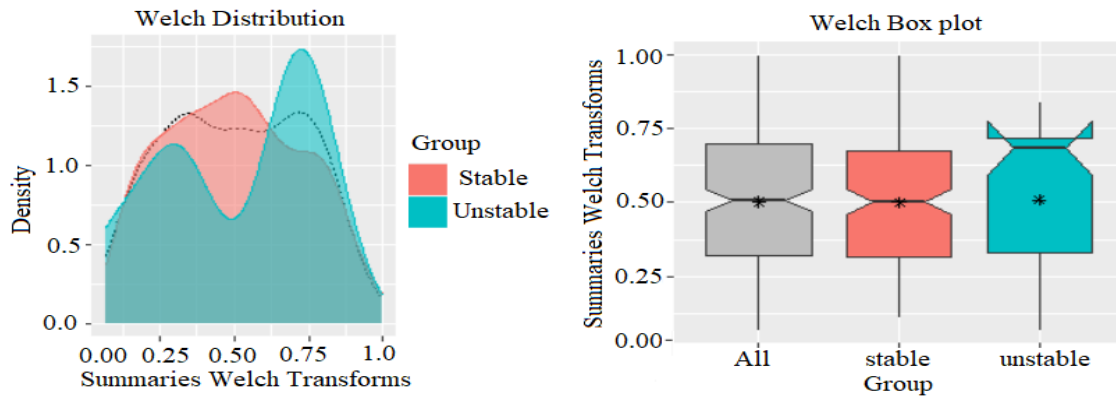
(c) Classification de l'état de stabilité d'une réponse instable issu du système S_3 .

Figure 25: Classifications sur des dictionnaires instables et les RMSE associées [40].

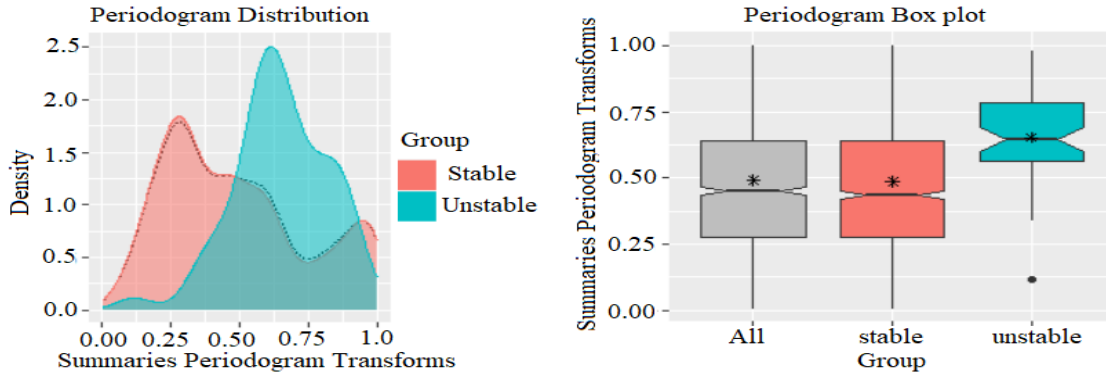
On constate ainsi que la projection des supports parcimonieux X_{ss} sur le dictionnaire des vitesses des rotors D_{rs} prédit les états de stabilité des signaux instables avec une marge d'erreur $\sim 10^{-3}$.

3.4 Comparaison du SLOD et des algorithmes de machine learning

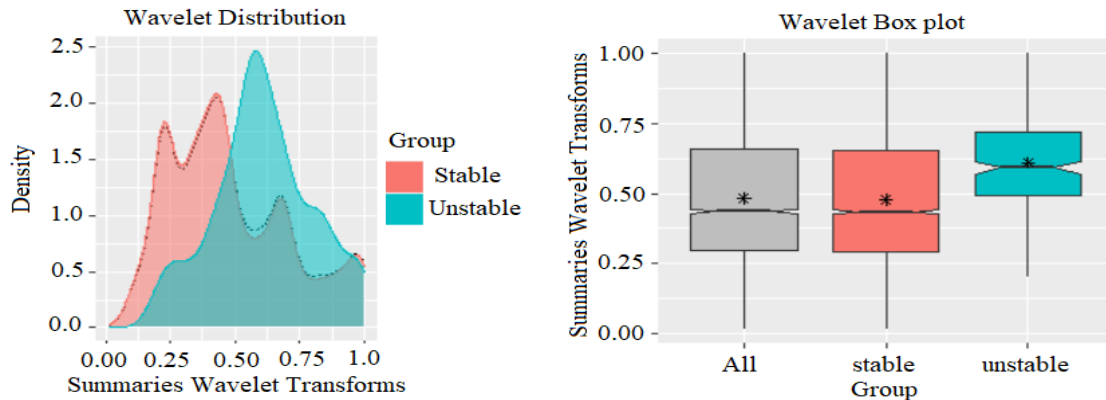
Les performances (*accuracy, reliability and reliability*) de prédiction du SLOD sont comparées à ceux des *10 - fold support vector machine (SVM), artificial neural network (ANN), decision trees (DT), random forest (RF), adaptive boosting (AdaBoost), convolutional neural network (CNN) and Long short-term memory (LSTM)* mis en œuvre en utilisant les indices de stabilité à haute dimension (HDSI) décrivant les signaux de réponse du générateur [96], immédiatement après l'élimination du défaut. Les Figures 26(a), 26(b) & 26(c) présentent les distributions de la densité et les boîtes à moustaches des attributs de Welch, de périodogramme et d'ondelettes, extraites des bases de données S_1 , S_2 et S_3 .



(a) Distribution et boîte à moustache de l'attribut de Welch (IEEE 2 area 4 machines).



(b) Distribution et boîte à moustache de l'attribut issu du périodogramme (IEEE 39 -bus).



(c) Distribution et boîte à moustache de l'attribut issu des ondelettes (IEEE 39 -bus).

Figure 26: Attributs de stabilité de Welch, du périodogramme et des ondelettes [40].

Les distributions des deux spectres stables/instables issues de la méthode de Welch, de la méthode du périodogramme et de la transformation en ondelette donnent un histogramme clair montrant que les valeurs extrêmes correspondent aux cas instables des réseaux S_1 , S_2 et S_3 . La mise en œuvre des algorithmes de *machine learning* basée sur les réponses des vitesses des rotors appelle essentiellement les fonctions d'apprentissage statistique de MATLAB pour le DSP test. L'ensemble des données du réseau IEEE 68 – bus est sélectionné pour illustrer la matrice de confusion (CM) des approches d'apprentissage supervisé, voir Figure 27. L'approche SLOD prédit correctement les $TP = 8,341$ signaux stables (sur 8 341 cas) et les $TN = 323$ signaux instables (sur 323 cas) et fait $FP + FN = 0$ fausse classification, voir Figure 27(a). L'approche SVM détecte avec précision les $TP = 7,799$ signaux stables (plus de 8 341 cas) et les $TN = 267$ signaux

instables (plus de 323 cas) et fait $FP+FN=598$ fausses classifications, Figure 27(b). L'approche DT sélectionne normalement les $TP=8,308$ signaux stables (sur 8 341 cas) et les $TN=233$ signaux instables (sur 323 cas) et effectue $FP+FN=123$ fausses classifications, voir Figure 27(c). L'approche RF prédit strictement les $TP=8,336$ signaux stables (plus de 8 341 cas) et les $TN=283$ signaux instables (plus de 323 cas) et fait $FP+FN=45$ fausses classifications, Figure 27(d). L'approche ANN prédit de manière absolue les $TP=8,397$ signaux stables (plus de 8 341 cas) et les $TN=225$ signaux instables (plus de 323 cas) et fait $FP+FN=42$ fausses classifications, Figure 27(e). L'approche AdaBoost discrimine les $TP=8,340$ signaux stables (sur 8 341 cas) et les $TN=242$ signaux instables (sur 323 cas) et fait $FP+FN=82$ fausses classifications, Figure 27(f).

CM_{SLOD} of S_3 test system

Actual class	stable	8341 96.3%	0 0.0%	100% 0.0%
	unstable	0 0.0%	323 3.7%	100% 0.0%
		100% 0.0%	100% 0.0%	100% 0.0%
		stable	unstable	
		Predicted class		

(a) SLOD on the S_3 test system.

CM_{SVM} of S_3 test system

Actual class	stable	7799 90.0%	56 0.6%	99.3% 0.7%
	unstable	542 6.3%	267 3.1%	33.0% 67.0%
		93.5% 6.5%	82.7% 17.3%	93.1% 6.9%
		stable	unstable	
		Predicted class		

(b) SVM on the S_3 test system.

CM_{DT} of S_3 test system

Actual class	stable	8308 95.9%	90 1.0%	98.9% 1.1%
	unstable	33 0.4%	233 2.7%	87.6% 12.4%
		99.6% 0.4%	72.1% 27.9%	98.6% 1.4%
		stable	unstable	
		Predicted class		

(c) DT on the S_3 test system.

CM_{RF} of S_3 test system

Actual class	stable	8336 96.2%	40 0.5%	99.5% 0.5%
	unstable	5 0.1%	283 3.3%	98.3% 1.7%
		99.9% 0.1%	87.6% 12.4%	99.5% 0.5%
		stable	unstable	
		Predicted class		

(d) RF on the S_3 test system.

CM_{ANN} of S_3 test system

Actual class	stable	8397 96.9%	25 0.3%	99.7% 0.3%
	unstable	17 0.2%	225 2.6%	93.0% 7.0%
		stable	unstable	
		99.8% 0.2%	90.0% 10.0%	99.5% 0.5%
		Predicted class		

(e) ANN on the S_3 test system

$CM_{AdaBoost}$ of S_3 test system

Actual class	stable	8340 96.3%	81 0.9%	99.0% 1.0%
	unstable	1 0.0%	242 2.8%	99.6% 0.4%
		stable	unstable	
		100.0% 0.0%	74.9% 25.1%	99.1% 0.9%
		Predicted class		

(f) AdaBoost on the S_3 test system

CM_{CNN} of S_3 test system

Actual class	stable	8324 96.1%	68 0.8%	99.2% 0.8%
	unstable	17 0.2%	255 2.9%	93.8% 6.3%
		stable	unstable	
		99.8% 0.2%	78.9% 21.1%	99.0% 1.0%
		Predicted class		

g) CNN on the S_3 test system.

CM_{LSTM} of S_3 test system

Actual class	stable	8324 96.1%	61 0.7%	99.3% 0.7%
	unstable	17 0.2%	262 3.0%	93.9% 6.1%
		stable	unstable	
		99.8% 0.2%	81.1% 18.9%	99.1% 0.9%
		Predicted class		

(h) LSTM on the S_3 test system.

Figure 27: Matrices de confusion des HDSI issues des vitesses des réseaux tests [40].

L'approche CNN détecte avec précision les $TP = 8,324$ signaux stables (plus de 8 341 cas) et les $TN = 255$ signaux instables (plus de 323 cas) et $FP + FN = 85$ fausses classifications, Figure 27(g). Le LSTM répertorie strictement les $TP = 8,324$ signaux stables (plus de 8 341 cas) et les $TN = 262$ signaux instables (plus de 323 cas) et $FP + FN = 78$ fausses classifications, Figure 27(h). Le Tableau VII résume les performances de la SLOD et des cinq algorithmes de machine learning (SVM, DT, RF, ANN, AdaBoost, CNN et LSTM) mis en œuvre à partir de la HDSI extraits de la base de données des vitesses des rotors.

Tableau VII : Comparaison de la SLOD et des algorithmes de machine learning.

Test systems	Tool	Learning Approaches	Accuracy (%)	Reliability (%)	Security (%)
IEEE 2-area 4 machines	Matlab Software	Proposed SLOD	99.99	99.99	100
		SVM	99.07	95.16	100
		DT	92.90	72.54	96.40
		RF	99.38	96.82	100
		ANN	96.29	91.66	97.22
		AdaBoost	98.14	91.66	99.61
		CNN	99.07	95.16	100
		LSTM	98.45	91.66	100
IEEE 39 -bus	Matlab Software	Proposed SLOD	99.99	99.99	100
		SVM	99.15	74.07	99.73
		DT	98.98	71.69	99.60
		RF	99.66	88.52	99.95
		ANN	97.96	74.07	98.49
		AdaBoost	99.15	64.00	99.91
		CNN	99.32	88.52	99.60
		LSTM	99.78	93.75	99.95
IEEE 68 -bus	Matlab Software	Proposed SLOD	99.99	99.99	100
		SVM	93.09	79.02	93.05
		DT	98.58	61.37	99.60
		RF	99.48	85.86	99.94
		ANN	99.00	65.41	99.96
		AdaBoost	99.05	66.52	99.98
		CNN	99.01	73.33	99.79
		LSTM	99.09	76.71	99.79

Le Tableau VIII récapitule tous les *CPU Time* de l'analyse de la base de données d'entraînement et de la réponse test. La SLOD fournit une prédiction presque parfaite sur tous les réseaux S_1 , S_2 et S_3 . De plus, les performances de fiabilité de la SLOD sont très élevées par rapport aux algorithmes de *machine learning*. Les performances de calcul de la SLOD ont été évaluées sur un ordinateur DELL configuré avec le processeur Intel i7-7700HQ à 4 cœurs fonctionnant à 2,80 GHz avec 16 Go de RAM. Bien que le *CPU time* soit relativement important, le code réel est en langage de script Matlab et, par conséquent, il peut être rendu plus rapide à l'aide de la programmation C.

Tableau VIII : CPU time (in sec.) du SLOD et des algorithmes de machine learning.

Test systems	Tool	Learning Approaches	Input Dataset	Offline Processing Time/s	Online Response Time/s
IEEE 2-area 4 machines	Matlab Software	Proposed SLOD	ASFI	120.70	0.125
		SVM		9.62	4.40
		DT		1.70	0.09
		RF		3.17	0.60
		ANN	HDSI	11.48	0.29
		AdaBoost		5.78	1.29
		CNN		100.92	1.40
		LSTM		205	1.26
IEEE 39 -bus	Matlab Software	Proposed SLOD	ASFI	42617.29	1.21
		SVM		137.14	20.09
		DT		0.57	0.04
		RF		1.70	0.34
		ANN	HDSI	45516.07	1.51
		AdaBoost		5238.78	702.57
		CNN		638	2.01
		LSTM		1358.82	3.50
IEEE 68 -bus	Matlab Software	Proposed SLOD	ASFI	409744.06	6.59
		SVM		4852.96	832.62
		DT		1.59	0.12
		RF		3.43	1.78
		ANN	HDSI	80089.93	1.12
		AdaBoost		11815.26	5672.12
		CNN		2106.60	3.78
		LSTM		110501.37	160.67

Le processus d'entraînement de SLOD recalcule à chaque nouvelle itération, la représentation parcimonieuse de tous les signaux déjà calculés précédemment. C'est la principale raison du temps de calcul important. Pour les bases de données volumineuses, la SLOD nécessite de bonnes performances de calcul pour sa mise en œuvre. De plus, la décomposition est effectuée échantillon par échantillon, et ce jusqu'à obtention du dictionnaire d'apprentissage. Cette lourde consommation du temps d'entraînement peut être réduite en parallélisant la base de données d'entraînement à l'aide de GPU comme dans la *deep learning*. Cela permettra d'apprendre plusieurs sous-dictionnaires (selon la puissance de calcul de l'ordinateur) simultanément, qui seront par la suite concaténés pour obtenir des indices parcimonieux adaptatifs (AFSI) de la base de données d'entraînement. Cette méthode d'approximation permettra de diviser par n (qui est fonction des cœurs de l'ordinateur) le temps de calcul pendant le processus d'entraînement.

3.5 Conclusion

Ce chapitre visait à développer un dictionnaire d'apprentissage prometteur nommé SLOD capable d'améliorer la stabilité des schémas de protection sur une large zone [115]. Ce dictionnaire surcomplet robuste peut reconstruire les signaux des vitesses des rotors et prédire l'état de stabilité de chaque réponse du générateur obtenu à partir de l'algorithme de recherche du temps critique d'élimination des défauts. Pour concevoir un dictionnaire d'apprentissage pour la classification de la stabilité, l'algorithme *K-Singular value decomposition* (K-SVD) couplé à l'*orthogonal matching pursuit* (OMP) a été mis en œuvre sur la base des signaux des réponses post-perturbation obtenue par simulation des défauts triphasés terre sur chaque ligne et proche des barres génératrices. En plus de sa prédiction basée sur une courte fenêtre de données, la SLOD présente de bonnes performances (99.9%) en termes de fiabilité et de temps calcul par rapport à : *10-fold support vector machine (SVM)*, *artificial neural network (ANN)*, *decision trees (DT)*, *random forest (RF)*, *and adaptive boosting (AdaBoost)*, *convolutional neural network (CNN)* and *long short-term memory (LSTM)* mis en œuvre sur la base des indices HDSI extraits des données des réseaux IEEE 2 area 4 machines, IEEE 39 -bus et IEEE 68 -bus.

Chapitre 4 — Apprentissage séparé des dictionnaires pour la prédiction de la stabilité dynamique

4.1 Introduction

L'apprentissage supervisé fixe et adaptatif des dictionnaires est proposé dans ce chapitre pour évaluer la stabilité transitoire des grands réseaux électriques immédiatement après l'élimination des défauts. Ces structures fixes (simples et hybrides) sont développées sur la base du dictionnaire d'impulsion (ID), de la transformée de Haar discrète (DHT), de la transformée en cosinus discrète (DCT), de la transformée en sinus discrète (DST) et de la transformée en ondelettes discrète (DWT) pour l'extraction des indicateurs parcimonieux existants dans les bases de données [116-119]. De même, les structures adaptatives basées sur l'algorithme *K-Singular value decomposition* (K-SVD) couplé à *l'orthogonal matching pursuit* (OMP) sont implémentées pour extraire les attributs annonciateurs d'instabilité. Pour mettre en œuvre ces dictionnaires, la base de données d'entraînement est subdivisée en deux : 1) scénarios des vitesses des rotors stables et 2) scénarios des vitesses des rotors instables. L'erreur quadratique moyenne (RMSE) est calculée pour évaluer le degré d'appartenance des nouveaux scénarios d'apprentissage.

Ce chapitre met en lumière les performances des dictionnaires (c'est-à-dire séparés des classificateurs) simples, hybrides et adaptatifs pour évaluer la stabilité qui n'a pas encore été abordée dans la littérature sur les réseaux intelligents. Afin de déterminer quelle approche d'apprentissage de dictionnaire est la meilleure, selon des métriques telles que la précision, la fiabilité et la sécurité, en plus des exigences de calcul en apprentissage et test, il est question d'implémenter vingt dictionnaires d'apprentissage supervisé stables et instables, pour la reconstruction parcimonieuse et la classification des réponses des signaux de stabilité sur la base de séries temporelles. L'architecture d'apprentissage des dictionnaires supervisés simples, hybrides et adaptatifs proposée pour reconstruire et

classer les signaux de stabilité transitoire immédiatement après l'élimination des défauts est présentée dans la Figure 28.

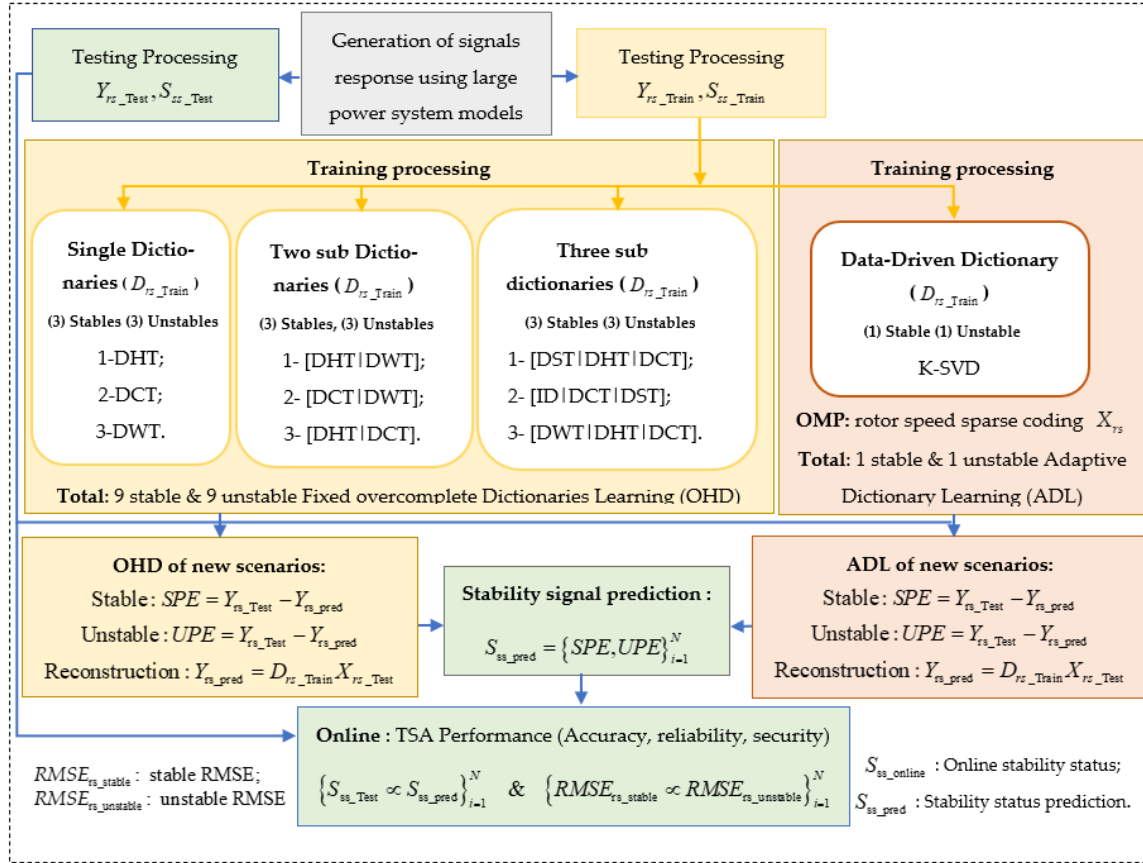


Figure 28: Organigramme pour d'apprentissage des dictionnaires surcomplets [35].

4.2 Dictionnaire fixe et adaptatif stable/instable

Considérons $Y_{rs} = [y_1, \dots, y_N] \in \mathbb{R}^{n \times N}$ les vitesses des rotors post-contingence, qui contient N signaux d'entrée de dimension n . La décomposition parcimonieuse de Y_{rs} sur un dictionnaire d'atome $D_{rs} \in \mathbb{R}^{n \times K}$ ($K > n$, nombre de colonnes supérieur au nombre de lignes) est calculée par :

$$Y_{rs} = \sum_{i=1}^K D_{irs} X_{irs} + \varepsilon_{irs} \quad (39)$$

L'*orthogonal matching pursuit (OMP)* est une méthode itérative capable de mettre en exergue les supports parcimonieux stables/instables utiles pour l'apprentissage des dictionnaires surcomplets [120].

4.2.1 Dictionnaire d'apprentissage fixe et simple

En se basant sur des expériences de prévision des PQDs et des séries chronologiques, des dictionnaires fixes ont été développés pour reconstruire les vitesses des rotors et prédire la stabilité transitoire des réponses des générateurs. Pour extraire des signaux de stabilité les indicateurs fixes parcimonieux et redondants, cinq méthodes simples sont explorées : 1) la transformée en cosinus discrète (DCT), 2) la transformée en sinus discrète (DST), 3) la transformée de Haar discrète (DHT), 4) la transformée en ondelette discrète (DWT) et 5) le dictionnaire d'impulsion (DI). Chacune de ces méthodes servira à implémenter un dictionnaire de cas stables et un dictionnaire de cas instables.

4.2.1.1 Transformée en Cosinus Discrète

La transformée en cosinus discrète (DCT) permet d'exprimer une série temporelle comme une somme de fonctions cosinus oscillant à différentes fréquences introduites par l'occurrence du défaut. Le noyau de projection DCT est défini par :

$$DCT = D_i = \sqrt{\frac{2}{n}} \left[\xi_i \cos \left(\frac{\pi(2j+1)i}{2n} \right) \right] \quad (40)$$

$$i, j = 0, 1, 2, 3, \dots, K - 1$$

Où : $\xi_i = 1/\sqrt{2}$ for $i=0$, par ailleurs $\xi_i = 1$. La DCT met en exergue des coefficients de décorrélation optimaux des signaux de stabilité, tout en regroupant l'énergie contenue dans les signaux en coefficients basse fréquence [121].

4.2.1.2 Transformée en Sinus Discrète

La transformée en sinus discrète (DST) dérivée de la DCT, permet d'exprimer le signal comme une somme de sinusoides avec différentes fréquences et amplitudes. Le noyau de projection DST est défini par :

$$DST = D_{rs} = \sqrt{\frac{2}{n}} \left[\xi_i \sin \left(\frac{\pi(2j+1)(i+1)}{2n} \right) \right] \quad (41)$$

$$i, j = 0, 1, 2, 3, \dots, K-1$$

où : $\xi_i = 1/\sqrt{2}$ pour $i = n-1$, par ailleurs $\xi_i = 1$. La DCT et la DST sont des opérations presque similaires à la transformée de Fourier discrète (DFT). La seule différence réside dans les noyaux de projection, qui donnent des coefficients réels pour la DCT, des coefficients imaginaires pour la DST et des coefficients complexes pour le DFT [122].

4.2.1.3 Transformée de Haar Discrète

La transformée de Haar discrète (DHT) est une forme d'onde rectangulaire orthogonale, qui permet d'échantillonner des points dans un sous-intervalle de $t \in [0,1)$ et où l'amplitude peut différer d'une fonction à l'autre comme suit [123]:

$$DHT \equiv D_{rs} = \frac{1}{\sqrt{N}} \begin{cases} 2^{m/2}; & \frac{k-1}{2^m} \leq t \leq \frac{k-1/2}{2^m} \\ -2^{m/2}; & \frac{k-1/2}{2^m} \leq t \leq \frac{k}{2^m} \\ 0; & \text{sinon } [0,1) \end{cases} \quad (42)$$

$$r = 2^m + k - 1; \quad t = n/N; \quad 0 \leq r, N \leq N - 1$$

$$DHT \equiv D_{rs} = \frac{1}{\sqrt{N}} \begin{cases} 2^{m/2}, & \frac{k-1}{2^m} \leq t < \frac{k-(1/2)}{2^m} \\ -2^{m/2}, & \frac{k-(1/2)}{2^m} \leq t < \frac{k}{2^m} \\ 0, & \text{otherwise in } [0,1) \end{cases} \quad (43)$$

$$r = 2^m + k - 1, \quad t = n/N, \quad 0 \leq r, n \leq N - 1$$

Où : m et k représentent la décomposition entière de l'indice r . La fonction de Haar temps-fréquence est unitaire et invariante par décalage.

4.2.1.4 Transformée en Ondelette Discrète

La transformée en ondelettes discrète (DWT) permet d'extraire dans un signal de stabilité, les attributs fixes à différents niveaux échelles (niveau 2, niveau 4 et niveau 6). Ses ondelettes (Daubechies et Symlets, pour notre cas d'étude) sont généralement

dénomérées par le nombre de coefficients α_k non nuls. Cette fonction normalisée et orthogonale est définie par l'ondelette mère $\Psi_{m,n}$ [124]:

$$DWT = D_{rs} = \frac{1}{\sqrt{a_0^m}} \sum_k \Psi_{m,n} \left(\frac{k - nb_0 a_0^m}{a_0^m} \right) \quad (44)$$

Où : $a = a_0^m$ et $b = nb_0 a_0^m$ sont le facteur d'échelle, le facteur de décalage et les n, m entiers positifs.

4.2.1.5 Dictionnaire d'impulsion

Le dictionnaire d'impulsions (ID) permet d'exprimer le signal comme une combinaison linéaire des vecteurs de base de Dirac représentant la réponse des pics de fréquence introduit par les défauts [11]. Il est défini par :

$$ID = D_{rs} = \begin{bmatrix} 1 & 0 & \dots & \dots & 0 \\ 0 & 1 & \dots & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & 1 & \dots \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}_{K \times K} \quad (45)$$

4.2.2 Dictionnaires fixes d'atome hybride à deux concaténations

Trois concaténations de deux sous-dictionnaires ([DHT|DWT], [DCT|DWT], [DHT|DCT]) ont été développées à partir des DCT, DHT et DWT pour l'extraction des attributs de stabilité parcimonieux et améliorer les performances des dictionnaires uniques. Cette concaténation permettra de mettre en exergue les attributs de stabilité par extraction simultanée des: 1) coefficients réels basses fréquences, 2) coefficients dans un sous-intervalle de $t \in [0,1)$ et 3) coefficients discrets par échelle.

4.2.2.1 Dictionnaire hybride de Haar-Cosinus

La transformée de Haar discrète et la transformée en cosinus discrète ont été associées pour extraire de manière simultanée dans le signal de stabilité, les formes d'onde rectangulaire dans un sous-intervalle de $t \in [0,1)$ et les coefficients réels basses fréquences.

$$Y_{rs_{n \times N}} = D_{rs} X_{K \times N} = \left[DHT | DCT \right]_{n \times K} X_{K \times N} \quad (46)$$

4.2.2.2 Dictionnaire hybride en Ondelette-Haar

La transformée en ondelette discrète et la transformée Haar discrète ont été associées pour extraire de manière simultanée dans le signal de stabilité, un ensemble de coefficient discret par échelle et les formes d'onde rectangulaire dans un sous-intervalle de $t \in [0,1)$.

$$Y_{rs_{n \times N}} = D_{rs} X_{K \times N} = \left[DWT | DHT \right]_{n \times K} X_{K \times N} \quad (47)$$

4.2.2.3 Dictionnaire hybride en Cosinus-Ondelette

La transformée en cosinus discrète et la transformée en ondelette discrète ont été associées pour extraire de manière simultanée dans le signal de stabilité, les coefficients réels basses fréquences et un ensemble de coefficient discret par échelle.

$$Y_{rs_{n \times N}} = D_{rs} X_{K \times N} = \left[DCT | DWT \right]_{n \times K} X_{K \times N} \quad (48)$$

4.2.3 Dictionnaires fixes d'atome hybride à trois concaténations

Pour améliorer les performances des structures simples et hybrides à deux concaténations, trois sous dictionnaires ([ID|DCT|DST], [DST|DHT|DCT] et [DWT|DHT|DCT]) sont développés à partir d'un arrangement de DCT, DST, ID, DHT et DWT.

4.2.3.1 Dictionnaire hybride de Dirac-Cosinus-Sinus

Le dictionnaire d'impulsion, la transformée en cosinus discrète et la transformée sinus discrets ont été implémentés pour extraire de manière simultanée dans le signal de stabilité, les pics de fréquences, les coefficients réels et les coefficients imaginaires basses fréquences.

$$Y_{rs_{n \times N}} = D_{rs} X_{K \times N} = \left[ID | DCT | DST \right]_{n \times K} X_{K \times N} \quad (49)$$

4.2.3.2 Dictionnaire hybride de Sinus-Haar-Cosinus

La transformée en sinus discrète, la transformée de Haar discrète et la transformée en cosinus discrète ont été implémentées pour extraire de manière simultanée dans le signal de stabilité, les coefficients imaginaires basses fréquences, des représentations dans un sous intervalle unitaire, invariante par décalage et les coefficients réels basses fréquences.

$$Y_{rs_{n \times N}} = D_{rs} X_{K \times N} = \left[DST | DHT | DCT \right]_{n \times K} X_{K \times N} \quad (50)$$

4.2.3.3 Dictionnaire hybride en Ondelette-Haar-Sinus

La transformée en ondelette discrète, la transformée de Haar discrète et la transformée en sinus discrète ont été implémentées pour extraire de manière simultanée dans le signal de stabilité, un ensemble de coefficients discrets par échelle, des représentations dans un sous intervalle unitaire, invariante par décalage, les coefficients imaginaires basses fréquences.

$$Y_{rs_{n \times N}} = D_{rs} X_{K \times N} = \left[DWT | DHT | DST \right]_{n \times K} X_{K \times N} \quad (51)$$

2.4.4 Dictionnaires adaptatifs d'atomes

La K-SVD est une généralisation de l'algorithme de k -mean clustering introduit par [36] pour l'apprentissage de dictionnaire d'atome. Pendant le processus d'apprentissage de la K-SVD, le dictionnaire $D_{rs} = [d_1 \dots d_K] \in \mathbb{R}^{n \times K}$ ($K \succ n$, dictionnaire surcomplet) est fixé, et le meilleur support parcimonieux $X_{rs} = [x_1 \dots x_N] \in \mathbb{R}^{K \times N}$ est trouvé, sous contrainte de parcimonie T . L'approche K-SVD est évaluée sur la base de la résolution d'un problème d'optimisation défini par :

$$\langle D_{rs}, X_{rs} \rangle = \arg \min_{D_{rs}, X_{rs}} \|Y_{rs} - D_{rs} X_{rs}\|_2^2 \quad \text{subject to } \forall_i \|x_i\|_0 \leq T \quad (52)$$

L'algorithme OMP est couplé à la K-SVD pour faciliter la convergence de la solution de l'Eq (52). Étant donné D_{rs} , le support parcimonieux $X_{rs} \in \mathbb{R}^{K \times N}$, est déterminée par :

$$x_{irs} = \arg \min_{x_{irs}} \|Y_{rs} - D_{rs} X_{rs}\|_2^2 \quad \text{subject to } \forall i, \|x_i\|_0 \leq T \quad (53)$$

Où : $D_{rs} \in \mathbb{R}^{n \times K}$ et T désignent respectivement les dictionnaires surcomplets et la contrainte de parcimonie. La mise à jour de chaque atome d_i entraîne nécessairement celle des entrées non nulles dans le vecteur ligne $x_i^T \neq 0$. Le dictionnaire apprend en résolvant le problème de minimisation suivant (Cf. Algorithme 2):

$$\min_{d_i, x_i} \|E_i - d_i x_i^T\|_2^2 \quad \text{subject to } \|d\|_2^2 = 1 \quad (54)$$

Où : $E_i = Y_{rs} - \sum_{j \neq i} d_j x_j^T$ désigne la matrice d'erreur d'approximation, qui peut être facilement décomposée en $U\Delta V^T$ [125].

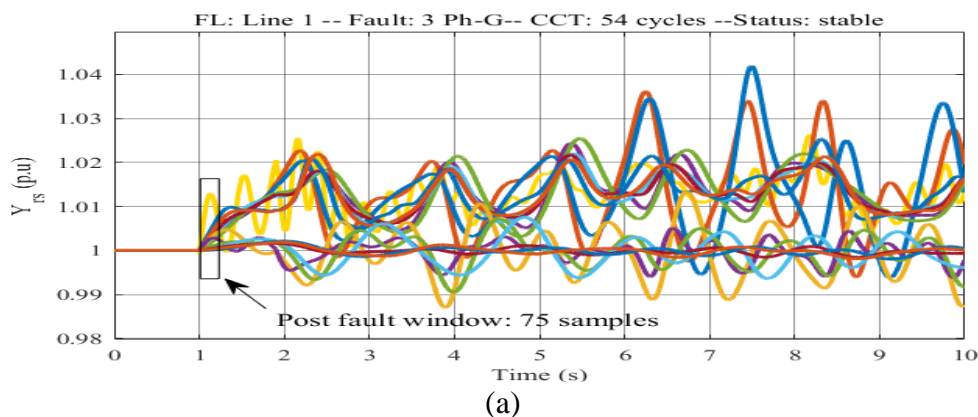
4.3 Reconstruction/classification sur des dictionnaires fixes et adaptatifs

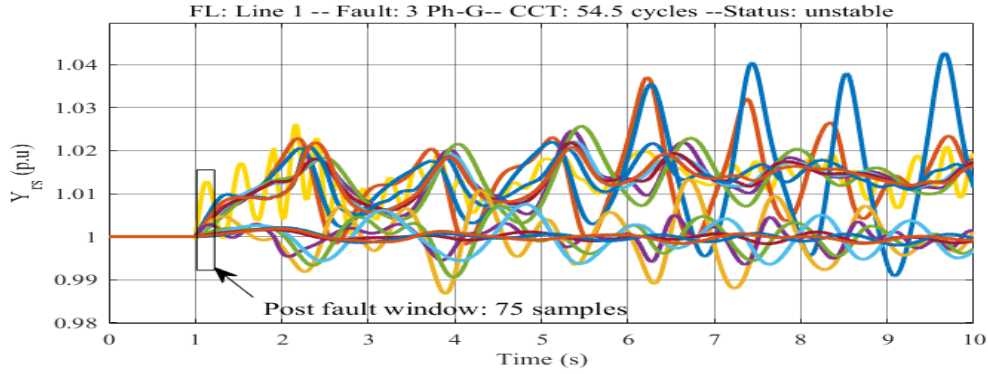
Les décompositions sur des dictionnaires supervisés simples ([DCT], [DHT], [DWT]), hybrides à deux concaténations ([DHT|DCT], [DWT|DCT], [DHT|DWT]) et hybrides à trois concaténations ([DST|DHT|DCT], [ID|DCT|DST], [DWT|DHT|DCT]) sont utilisées pour reconstruire les réponses des générateurs et prédire la stabilité transitoire des machines du réseau IEEE 68 -bus. Ces signaux générés sont séparés en deux : 1) base de données des scénarios stable et 2) base de données des scénarios instable. Chacune des bases de données va permettre de construire un dictionnaire identique. Par exemple le dictionnaire DCT stable sera construit uniquement à partir des scénarios stables et le dictionnaire DCT instable sera développé uniquement sur la base des scénarios instables. Pour évaluer la stabilité de la base de données test, chacun de ces signaux sera projeté sur ces dictionnaires de même nature. Par hypothèse, le dictionnaire (stable ou instable) qui présentera une faible valeur RMSE, imposera son état de stabilité au signal test [35].

4.4 Évaluation des performances des dictionnaires d'atome joint et séparé

4.4.1 Descriptif des données d'apprentissage

Le réseau IEEE 68 -bus est utilisé pour simuler une diversité de défaut, de sévérité variable par pas de 0,5 cycle jusqu'à la limite critique de stabilité. Chaque défaut est appliqué à $t = 1$ s sur une ligne ou barre génératrice, et cela pour une durée de simulation numérique de 10 s. Par cette simulation, 75×28880 signaux des vitesses des rotors sont générées et réparties en 75×20216 pour le processus d'apprentissage et les 75×8664 pour le test. Pour chaque simulation, uniquement 75 échantillons sont retenus (taux d'échantillonnage : 600 échantillons/sec) par vitesse de rotor pour l'apprentissage des dictionnaires, ce qui signifie une fenêtre de temps de 125 ms pour la prise de décision (7,5 cycles de fondamental). La Figure 29 présente un exemple de vitesse stable (a) et instable (b) enregistré pendant la recherche du temps d'élimination des défauts critiques sur la ligne 1 du réseau IEEE 68 -bus. Les réponses des générateurs obtenues pour l'apprentissage sont subdivisées en ensemble d'apprentissages stables (c'est-à-dire : $75 \times 19,477$) et ensemble d'apprentissages instables (c'est-à-dire : 75×739). Les cas instables ne représentent donc que 3,8 % du total des cas, ce qui signifie que l'ensemble de données est fortement biaisé vers les cas stables.





(b)

Figure 29: Allure des vitesses des rotors stables (a) et instables (b) du IEEE 68 -bus [35].

Le Tableau IX présente la configuration du jeu de données utilisée pour reconstruire et classer la base de données de test (c'est-à-dire : contenant 8,664 signaux : 8,341 scénarios stables et 323 scénarios instables soit 3,72 % du total des cas) basée sur l'apprentissage supervisé de dictionnaires surcomplets.

Tableau IX : Configuration des signaux utilisés pour la décomposition parcimonieuse.

Sparse Signal Decomposition	Offline Features Extraction	Online Reconstruction and TSA
Adaptive Dictionary Learning (ADL)	Rotor speed (For D_{rs} and X_{rs} computed)	Prediction error (For $RMSE_{rs}$ computed)
Fixed Dictionaries Learning (1,2,3H OHD)	Rotor speed (For D_{rs} and X_{rs} computed)	Prediction error (For $RMSE_{rs}$ computed)

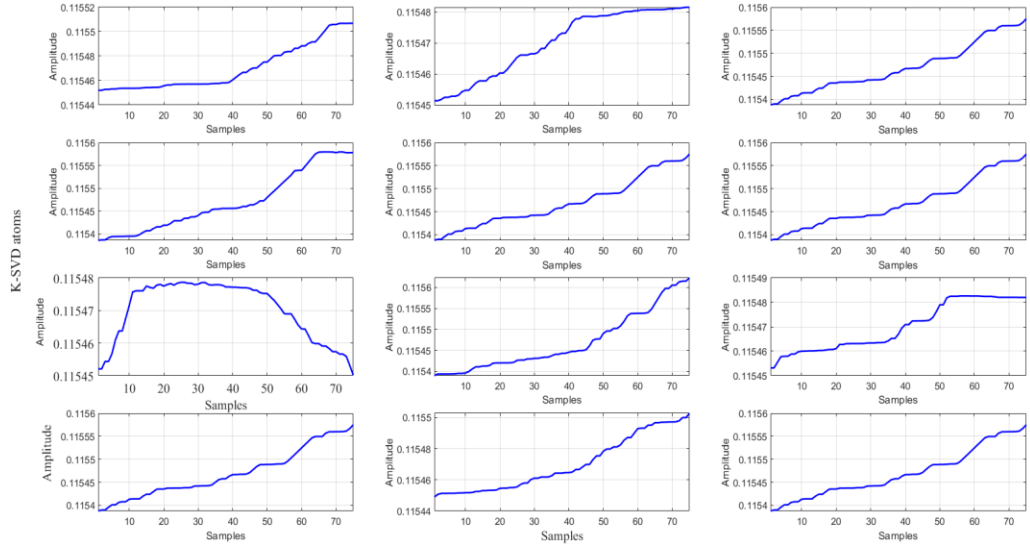
4.4.2 Extraction des atomes parcimonieux fixes et adaptatifs

Le nombre d'attributs fixes et hybrides (c'est-à-dire : 25 indicateurs) extraits par signal dépend du nombre d'itérations défini par l'algorithme *orthogonal matching pursuit* [126]. Cependant, l'apprentissage sur un dictionnaire adaptatif (K-SVD) permet d'extraire 75 attributs par réponse du générateur. Le Tableau X résume les décompositions parcimonieuses des signaux développées en apprentissage et utilisées pour le DSP test.

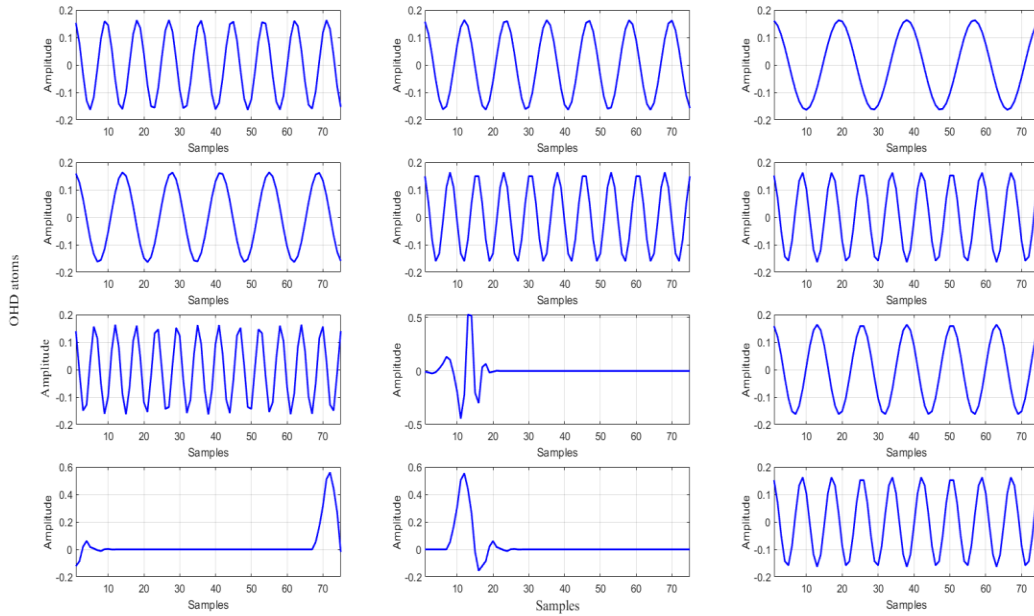
Tableau X : Configuration des dictionnaires d'apprentissage supervisés.

Configuration	Learning Approaches	Dictionaries / Feature Size	Sparse Expansion Coefficient Size
Adaptive dictionary (ADL)	K-SVD	75×20216	8700×20216
	DHT		75×1617280
Single dictionary (SDL)	DCT	25×20216	75×1516200
	DWT		75×1556632
	[DHT DWT]		75×3133480
Two sub-dictionaries (2H-OHD)	[DWT DCT]	25×20216	75×3153696
	[DHT DCT]		75×2304624
	[DST DHT DCT]		75×3800608
Three sub-dictionaries (3H-OHD)	[ID DCT DST] [11]	25×20216	75×4589032
	[DWT DHT DCT]		75×3820824

Les dictionnaires K-SVD et OHD stables sont initialisés et entraînés avec des signaux stables en entrée, tandis que les dictionnaires K-SVD et OHD instables sont entraînés avec des réponses instables uniquement. Pour une parcimonie fixe $T = 10$ (c'est-à-dire : matrice de nombres qui comprend de nombreux zéros ou valeurs qui ne seront pas significatifs), les coefficients sont calculés à l'aide de l'algorithme OMP et le maximum d'itérations défini égal à 90. À partir du support parcimonieux extrait de la base de données test, les dictionnaires K-SVD & OHD stables et instables extraits de la base de données d'entraînement permettront de reconstruire les réponses des machines et de prédire stabilité transitoire du réseau électrique test. Des exemples d'atomes K-SVD (a) et OHD (b) obtenus pendant le processus d'apprentissage des réponses des vitesses des rotors sont présentés dans la Figure 30:



(a)



(b)

Figure 30: Illustration des atomes K-SVD (a) et OHD (b) [35].

4.4.3 Reconstruction des signaux à partir des atomes fixes et adaptatifs

Chaque signal contenu dans la base de données test est projeté sur les dictionnaires stables et instables appris séparément. La Figure 31 présente des décompositions parcimonieuses de deux signaux de test ID 1 (stable en bleu) et ID 23 (instable en rouge) choisis aléatoirement dans la base de données test. En se basant sur les trajectoires des

signaux tests et sur ceux prédits, on constate que les atomes K-SVD généralisent bien sur les transitoires ID 1 (stable) et ID 23 (instable).

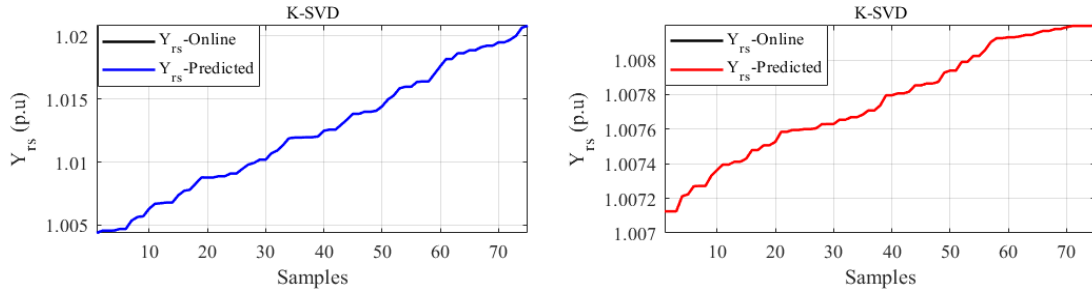


Figure 31 : Reconstruction d'un signal stable et instable sur le dictionnaire K-SVD [35].

Les atomes OHD, tels que les dictionnaires simples (DCT, DWT, DHT), les dictionnaires à deux concaténations ([DHT|DCT], [DWT|DCT], [DHT|DWT]) et les dictionnaires à trois concaténations ([DST|DHT|DCT], [ID|DCT|DST], [DWT|DHT|DCT]) sont utilisés pour reconstruire les vitesses des rotors stable (ID 1 en couleur bleue) et instable (ID 23 en couleur rouge) conformément aux réponses tests (couleur noire), voir Fig. 32.

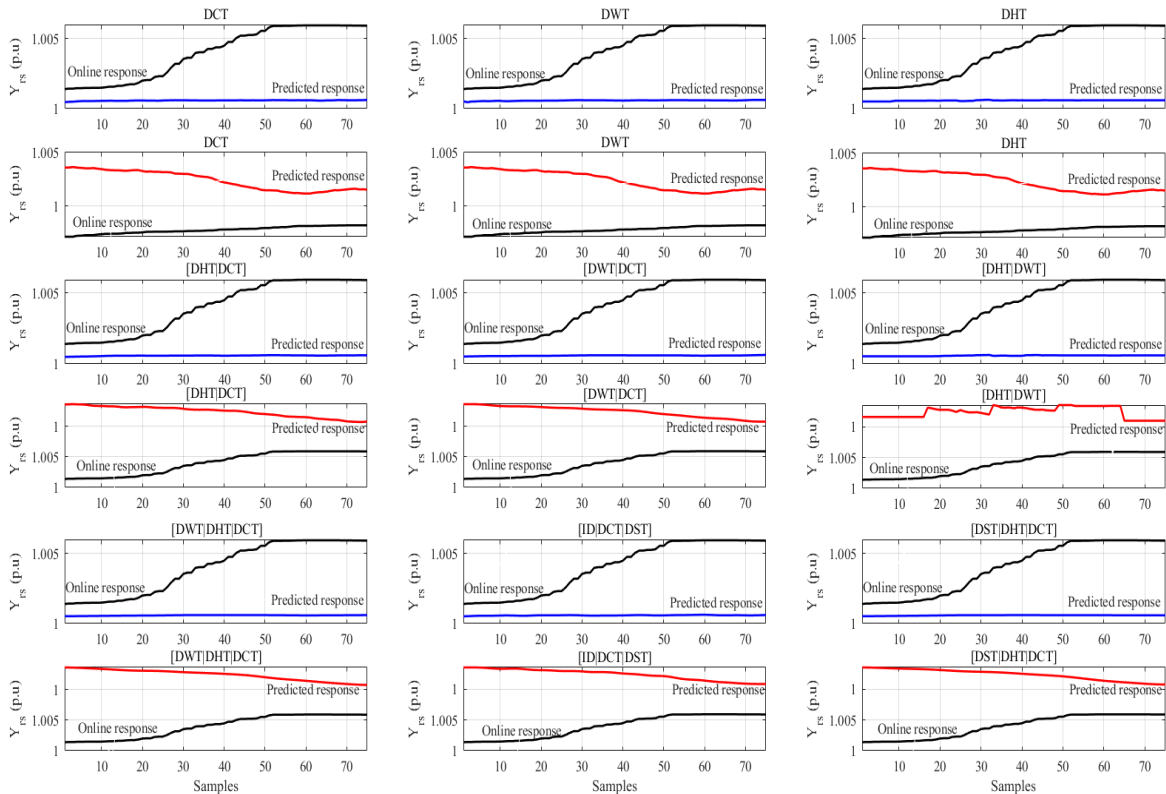


Figure 32 Reconstruction d'un signal stable et instable sur les dictionnaires OHD [35].

4.4.4 Prédiction de la stabilité à partir des atomes fixes et adaptatifs

Deux signaux ID 1 & ID 23 sont choisis aléatoirement dans la base de données test pour illustrer le processus de prédiction parcimonieuse sur : SDL, 2H-OHD, 3H-OHD & K-SVD. La projection de ces deux signaux sur des dictionnaires appris stables/instables permet d'évaluer l'état de stabilité test en se basant sur le principe de rapprochement des labels d'apprentissage. Par hypothèse, nous admettons que les dictionnaires stables ont la capacité de bien reconstruire les signaux tests stables, et par conséquent auront une RMSE faible. De même, les dictionnaires instables ont la capacité de bien reconstruire les signaux tests instables, et par conséquent auront une RMSE faible.

4.4.4.1 Prédiction de la stabilité à partir des dictionnaires simples et adaptatifs

La Figure 33 présente les RMSE obtenues en projetant les signaux ID 1 (stable, de couleur bleue) et ID 23 (instable, de couleur rouge) sur SDL (stables et instables) et K-SVD (stables et instables). On constate ainsi que les projections de tous les signaux stables sur les dictionnaires (stable [DCT], stable [DHT], stable [DWT], stable [KSVD]) ont des RMSE faibles, car l'histogramme bleu (qui correspond à une réponse stable) est inférieur à l'histogramme rouge (qui correspond à une réponse instable).

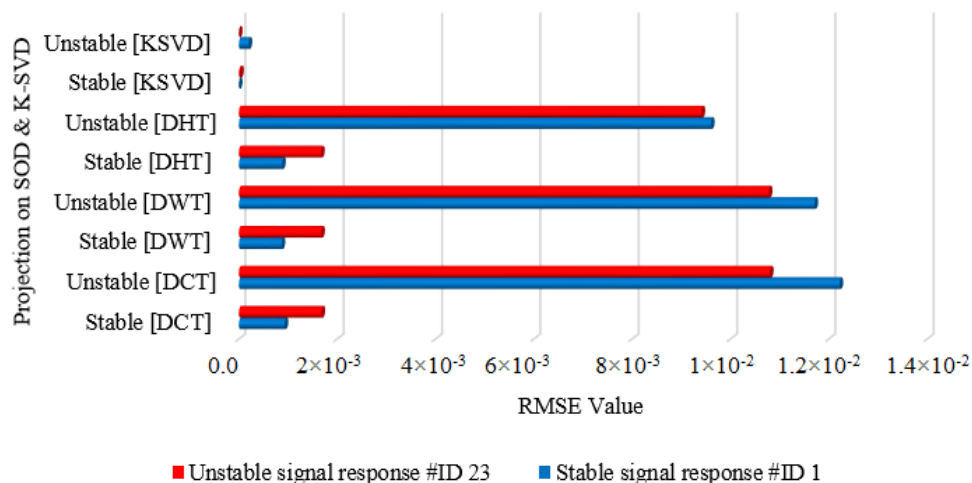


Figure 33: Projection d'un signal stable ID 1 et instable ID 23 sur SDL et ADL [35].

De même les projections de tous les signaux instables sur les dictionnaires (instable [DCT], instable [DHT], instable [DWT], instable [KSVD]) ont des RMSE faibles, car

l'histogramme rouge (qui correspond à une réponse instable) est inférieur à l'histogramme bleu (qui correspond à une réponse stable). Selon SDL, le dictionnaire DWT est le meilleur descripteur des vitesses des rotors stables, tandis que pour les scénarios instables, le dictionnaire DHT est indiqué pour la prédiction.

4.4.4.2 Prédiction sur des dictionnaires fixes à deux concaténations et adaptatifs

La Figure 34 présente les RMSE obtenues en projetant les signaux ID 1 (stable, de couleur bleue) et ID 23 (instable, de couleur rouge) sur 2H-OHD (stables et instables) et K-SVD (stables et instables). On constate ainsi que les projections de tous les signaux stables sur les dictionnaires (stable [DHT|DCT], stable [DWT|DCT], stable [DHT|DWT], stable [KSVD]) ont des RMSE faibles, car l'histogramme bleu (qui correspond à une réponse stable) est inférieur à l'histogramme rouge (qui correspond à une réponse instable). De même les projections de tous les signaux instables sur les dictionnaires (instable [DHT|DCT], instable [DWT|DCT], instable [DHT|DWT], instable [KSVD]) ont des RMSE faibles, car l'histogramme rouge (qui correspond à une réponse instable) est inférieur à l'histogramme bleu (qui correspond à une réponse stable).

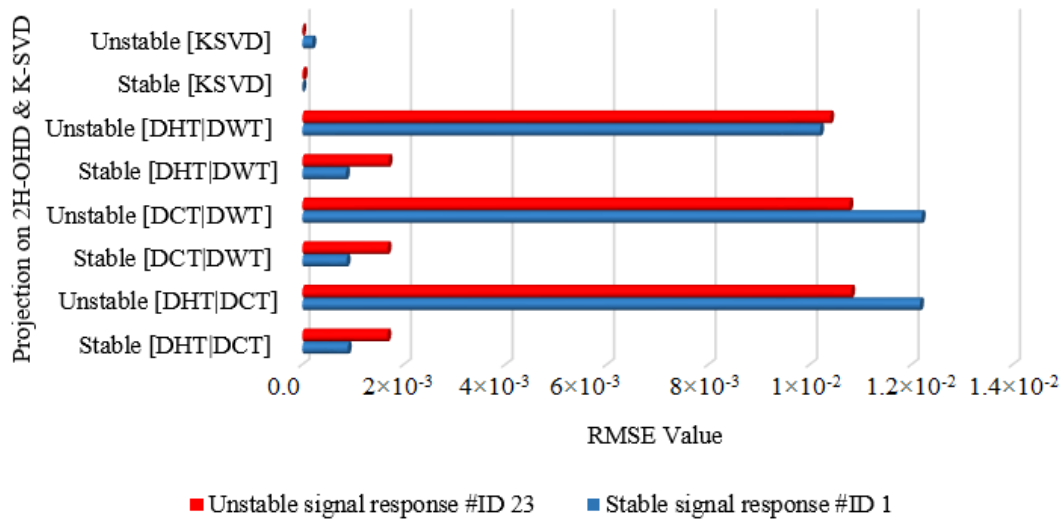


Figure 34: Projection d'un signal stable ID 1 et instable ID 23 sur 2H-OHD et ADL [35].

Selon 2H-OHD, le dictionnaire [DHT|DWT] est le meilleur descripteur des vitesses des rotors stables, tandis que pour les scénarios instables, le dictionnaire [DWT|DCT] est indiqué. Toutefois le dictionnaire instable [DHT|DWT] commet une erreur de

classification sur le signal instable ID 23, car il le détecte comme stable, pourtant celui-ci est instable.

4.4.4.3 Prédiction sur des dictionnaires fixes à trois concaténations et adaptatifs

La Figure 35 présente les RMSE obtenues en projetant les signaux ID 1 (stable de couleur bleue) et ID 23 (instable de couleur rouge) sur 3H-OHD (stables et instables) et K-SVD (stables et instables). On constate ainsi que les projections de tous les signaux stables sur les dictionnaires (stable [DST|DHT|DCT], stable [ID|DCT|DST], stable [DWT|DHT|DCT], stable [KSVD]) ont des RMSE faibles, car l’histogramme bleu (qui correspond à une réponse stable) est inférieur à l’histogramme rouge (qui correspond à une réponse instable). De même les projections de tous les signaux instables sur les dictionnaires (instable [DST|DHT|DCT], instable [ID|DCT|DST], instable [DWT|DHT|DCT], instable [K-SVD]) ont des RMSE faibles, car l’histogramme rouge (qui correspond à une réponse instable) est inférieur à l’histogramme bleu (qui correspond à une réponse stable).

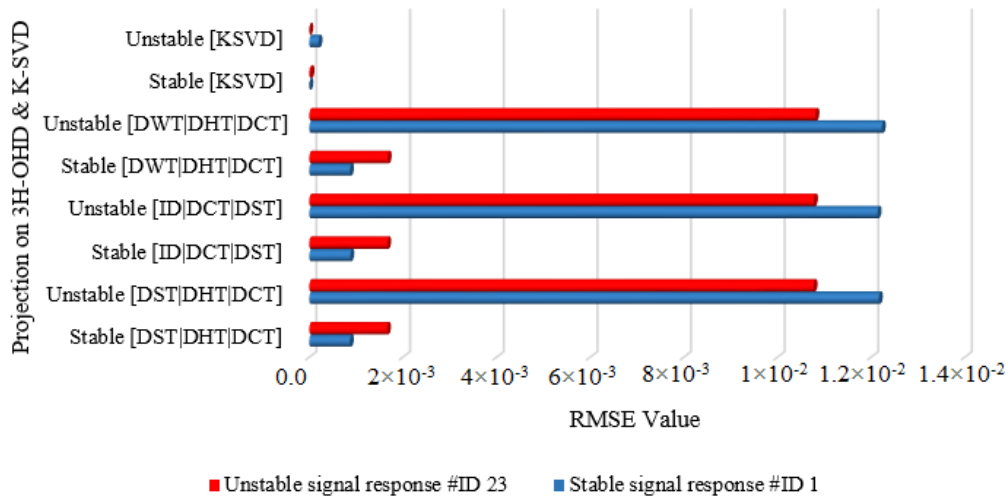


Figure 35: Projection d’un signal stable ID 1 et instable ID 23 sur 3H-OHD et ADL [35].

Selon 3H-OHD, le dictionnaire [DST|DHT|DCT] est le meilleur descripteur des vitesses des rotors stables et instables. La Figure 36 présente les degrés de stabilité des signaux ID 1 et ID 23, développés sur la base des dictionnaires stables et instables. Pour chaque signal, le ratio du RMSE instable et du RMSE stable permet de calculer ces degrés

de stabilité. Par conséquent, un petit RMSE entraîne un degré élevé de stabilité, ce qui signifie une probabilité plus élevée que le cas soit stable, tandis qu'une petite valeur signifie que le cas est probablement classé comme instable. Pour le cas stable ID 1, la K-SVD donne un très grand degré de stabilité par rapport à [DCT], [DHT], [DWT], [DHT|DCT], [DWT|DCT], [DHT|DWT], [DST |DHT|DCT], [ID|DCT|DST] et [DWT|DHT|DCT]. De même, la K-SVD donne un très faible degré de stabilité pour le signal ID 23, ce qui confirme une bonne séparabilité de stabilité du dictionnaire basé sur K-SVD.

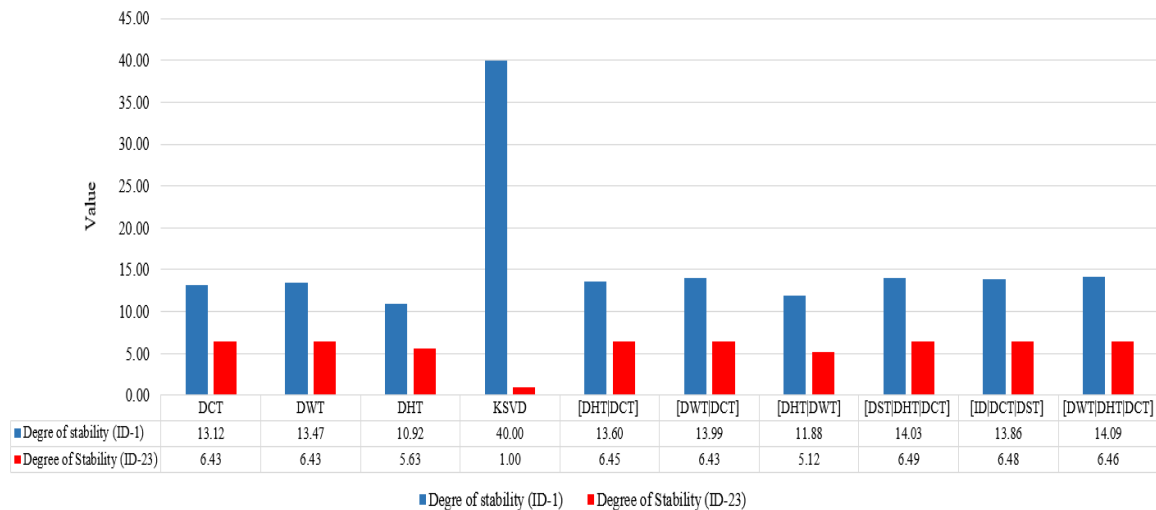


Figure 36: Degrés de stabilité des signaux tests ID 1 et ID 23 [35].

4.4.4.4 Évaluation des métriques des dictionnaires fixes et adaptatifs

Les métriques (à savoir : précision, fiabilité et sécurité) sont utilisées pour évaluer les performances d'apprentissage des dictionnaires surcomplets fixes et adaptatifs. La précision permet de définir la capacité des dictionnaires appris à classer correctement les 8341 scénarios stables et 323 scénarios instables de la base de données de test. La fiabilité permet d'évaluer les performances des dictionnaires appris à prédire les scénarios instables. La sécurité permet de définir la capacité des dictionnaires appris à prédire les scénarios stables. Ces trois métriques sont globalement évaluées en projetant sur chaque dictionnaire stable et instable, les nouveaux scénarios. Le Tableau XI résume l'efficacité de la TSA test de chaque OHD et ADL développé à partir des deux bases (stable et instable) de données d'entraînement totalement séparées. De plus, les performances des dictionnaires OHD et

KSVD sont comparées à ceux de l’approche SLOD développée dans [127], à partir d’une base de données d’entraînement contenant les scénarios stable et instable.

Tableau XI : Métrique de comparaison des dictionnaires d’apprentissage supervisés.

Configuration	Learning approaches	Accuracy	Reliability	Security
Adaptive dictionary (ADL) (With a separate training dataset)	K-SVD	93.42	81.68	93.35
SLOD (With a joint training dataset)	TK-SVD	99.99	99.99	99.99
Single dictionary (SDL)	DCT	94.76	93.05	94.52
	DHT	95.79	94.10	95.66
	DWT	94.96	93.40	94.74
Two sub-dictionaries (2H-OHD)	[DHT DWT]	95.41	93.75	95.24
	[DWT DCT]	94.79	93.05	94.56
Three sub-dictionaries (3H-OHD)	[DHT DCT]	94,82	93,05	94,59
	[DST DHT DCT]	94,88	93,05	94,66
	[ID DCT DST] [17]	94,88	93,05	94,66
	[DWT DHT DCT]	94,81	93,05	94,58

4.4.4.5 Temps d’apprentissage des dictionnaires fixes et adaptatifs

Les performances de calcul de l’apprentissage des dictionnaires fixes et adaptatifs ont été évaluées sur un ordinateur DELL configuré avec le processeur Intel i7-7700HQ à 4 cœurs fonctionnant à 2,80 GHz avec 16 Go de RAM. Le Tableau XII résume le temps de calcul du processeur pour l’analyse de la base de données d’entraînement et la réponse test. Les dictionnaires adaptatifs surcomplets proposés prennent plus de temps de calcul du processus d’entraînement (c’est-à-dire : 298 475,89 s soit 82.90 h) pour l’extraction de caractéristiques parcimonieuses. Cependant, les fonctionnalités de la K-SVD permettent une prédiction de stabilité transitoire test très rapide (c’est-à-dire : temps de calcul de 4,45 s). Comparé au meilleur algorithme d’apprentissage de dictionnaire, le SLOD (DK-SVD), qui a un temps d’entraînement de 409 744,06 s (113,81 h) et un temps de lecture test de 6,59 s, l’approche ADL (K-SVD) est plus rapide.

Tableau XII : Temps CPU (en sec.) des dictionnaires d'apprentissage supervisés.

Configuration	Learning Approaches	Offline Processing Time/s	Online Processing Time/s
Adaptive dictionary (ADL) (with a separate training dataset)	K-SVD	298,475.89	4.45
SLOD (with a joint training dataset)	DKSVD	409,744.06	6.59
Single dictionary (S-OHD)	DHT	16,518.44	6061.06
	DCT	42,290.29	17,547.32
	DWT	45,882.93	17,366.59
Two sub-dictionaries (2H-OHD)	[DHT DWT]	31,535.18	13,854.68
	[DWT DCT]	52,812.70	18,519.29
	[DHT DCT]	55 830,41	20 278,90
Three sub-dictionaries (3H-OHD)	[DST DHT DCT]	33 949,11	14 489,51
	[ID DCT DST] [17]	28 617,70	12 320,68
	[DWT DHT DCT]	40 062,59	16 521,75

Ces différentes méthodes sont comparées à la Figure 37 en utilisant les données de la Table XI. L'essentiel est que la TSA basée sur le dictionnaire joint permet de mieux reconstruire et prédire la stabilité transitoire.

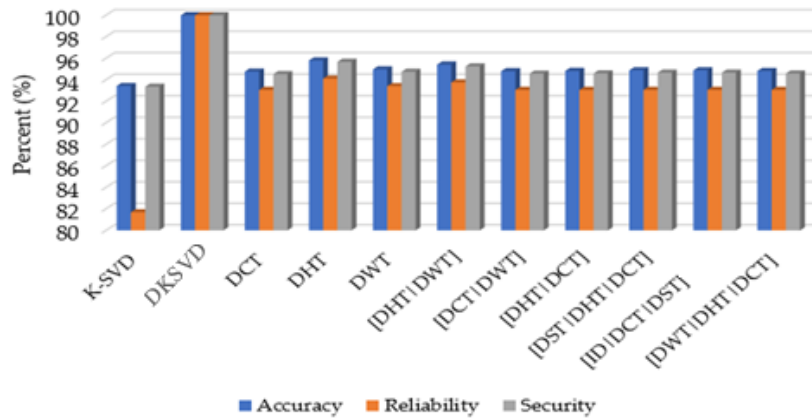


Figure 37: Comparaison des performances de classification des dictionnaires [35].

Le DHT s'est avéré être le meilleur dictionnaire fixe en termes de fiabilité et de sécurité, bien que la différence ne soit pas énorme entre tous les types de dictionnaires fixes. Étant

donné que la DHT ne prend qu'environ 50 % du temps de calcul du 3H-OHD de [11] à la fois dans l'apprentissage supervisé et la prédiction, elle est nettement plus prometteuse et mérite des recherches plus approfondies dans le contexte de la TSA. De plus, étant donné la courte fenêtre de prédiction (75 ms) et les classes de données très asymétriques (la classe instable ne représentant généralement que 3 à 10 % de l'ensemble de données total), la classification basée sur la reconstruction utilisant la K-SVD semble inadaptée et plutôt, l'algorithme d'apprentissage supervisé basé sur DK-SVD de [40] devrait être davantage étudié et comparé à la classification basée sur la reconstruction utilisant un dictionnaire DHT fixe. Ces deux options ont émergé comme les meilleures méthodes de prédiction DSP basées sur un dictionnaire de la présente étude.

4.5 Conclusion

Ce chapitre visait à reconstruire et à prédire la stabilité transitoire basée sur l'apprentissage de dictionnaires surcomplets fixes et adaptatifs. Concrètement, cette étude développe et met en œuvre un nouvel arrangement d'atomes capable d'améliorer la surveillance et le contrôle de la stabilité à grande échelle en utilisant une approche de reconstruction de signal à double dictionnaire. À cette fin, des structures fixes et adaptatives ont été étudiées sur la base des dictionnaires DCT, DST, DHT et DWT pour la reconstruction parcimonieuse des vitesses des rotors. Ensuite, l'état de stabilité est déterminé sur la base des indices RMSE des vitesses des rotors en utilisant chaque dictionnaire stable/instable. Le rapport RMSE de ces deux prédicteurs séparés est proposé comme mesure de la stabilité (ou de l'instabilité) du signal post-perturbation observé sur une courte fenêtre de données de 7,5 cycles. Plusieurs concaténations de deux (c'est-à-dire : [DHT|DCT], [DWT|DCT] & [DHT|DWT]) et trois (c'est-à-dire : [DST|DHT|DCT], [ID|DCT|DST] & [DWT|DHT |DCT]) des sous-dictionnaires fixes sont utilisés pour effectuer une décomposition parcimonieuse similaire sur les doubles dictionnaires. Alternativement, l'approche adaptative K-SVD est mise en œuvre à l'aide de doubles dictionnaires appris à partir de l'ensemble des données des réponses des générateurs stables/instables. Alors que la K-SVD adaptative a démontré une meilleure performance de reconstruction, en termes d'erreur de prédiction, sa performance de fiabilité n'était pas satisfaisante en raison de l'ensemble de données fortement asymétriques dérivé du système de test IEEE 68 bus (3,8 % des cas stables seulement).

Conclusion Générale et Perspectives

Cette thèse présente une nouvelle approche d'extraction des attributs capable de prédire la stabilité parcimonieuse des réseaux électriques. Pour mener à bien ce travail, une méthode simple et rapide de génération (en un clic) massive des bases de données a été développée sur Simscape power systems à partir des modèles IEEE 2-area 4 machines, IEEE 39 et 68-bus. Cette base de données est principalement utilisée pour 1) extraire les nouveaux indicateurs de stabilité à haute dimension (HDSI) pour la *machine learning*, 2) extraire les attributs de stabilité adaptatifs joint au classificateur et 3) extraire les attributs adaptatifs et fixes séparés aux classificateurs. Immédiatement après l'élimination des défauts, les indices de stabilité à haute dimension (HDSI) ont été implémentés pour mettre en exergue plus de 500 attributs physiquement interprétables par vitesse de rotor et par taux de changement d'énergie transitoire (RoCoTE). Ces indicateurs sont utilisés pour implémenter les algorithmes : *support vector machine (SVM)*, *decision trees (DT)*, *random forest (RF)*, *artificial neural network (ANN)* et *adaptive boosting (AdaBoost)* sur Matlab et Rstudio. Alors que la plupart des méthodes fonctionnent de manière similaire, avec une précision d'environ 100 % sur les cas test utilisant le même ensemble d'attributs basés sur HDSI (non parcimonieux), le classificateur RF avec l'importance de l'indice Gini associée, permet un classement explicite et une interprétation des attributs, ce qui donne la priorité aux caractéristiques du domaine fréquentiel par rapport au domaine temporel.

L'algorithme K-SVD couplé à l'OMP est utilisé pour extraire les attributs parcimonieux de la base de données des vitesses des rotors labellisés. Le dictionnaire d'atomes des vitesses des rotors est implémenté pour reconstruire les réponses des générateurs synchrones de la base de données test. Le dictionnaire d'atomes des états de stabilité extraits de la base de données d'entraînement développé a permis d'introduire un terme discriminant des classes stables/instables dans le dictionnaire des vitesses des rotors pour la DSP test. Il ressort de cette étude que l'approche SLOD donne des performances très satisfaisantes dans tous les cas (c'est-à-dire, ~99,99% de précision, ~99,99% de fiabilité et 100% de sécurité) par rapport aux algorithmes SVM, ANN, DT, RF, AdaBoost, CNN, LSTM qui utilisent les indices de stabilité à haute dimension (HDSI) basés sur des

séries chronologiques des vitesses des rotors comme entrée pour le DSP. Les dictionnaires simples (K-SVD, DCT, DHT et DWT) hybrides à deux concaténations ([DHT|DCT], [DWT|DCT] et [DHT|DWT]) et hybrides à trois concaténations ([DST|DHT|DCT], [ID|DCT|DST] et [DWT|DHT|DCT]) ont été implémentés pour évaluer l'impact de l'apprentissage séparé (stable ou instable) et des atomes prédéfinis sur la DSP test. Pour y parvenir, la base de données d'entraînement est subdivisée en deux : 1) un ensemble constitué des scénarios stables et 2) un ensemble constitué des scénarios instables. Chacune de ces bases de données a été utilisée pour développer des dictionnaires séparés des classificateurs. La projection des nouveaux scénarios sur chacun de ces dictionnaires a permis de définir le degré d'appartenance des classes stables/instables des bases de données tests. L'ensemble, de cette étude a prouvé que les dictionnaires fixes [DWT], [DHT|DWT] et [DST|DHT|DCT] sont de meilleurs descripteurs de stabilité par rapport aux K-SVD adaptatifs [DHT], [DWT], [ID|DCT|DST] et [DWT|DHT|DCT] mais aucune de ces méthodes n'a pu égaler les bonnes performances de la méthode rapportée dans [40].

Nous évaluerons plus avant les performances du modèle SLOD sur une base de données combinée constituée de la vitesse du rotor des réseaux de test IEEE 39 et 118 bus pour accéder à sa capacité de généralisation dans l'esprit de [35]. Nous prévoyons également d'évaluer le nouveau schéma sur un simulateur en temps réel OPAL-RT e-megasim pour démontrer que l'évaluation de la stabilité basée sur l'approche SLOD peut être effectuée en temps réel avec des données des PMUs bruitées, en moins de 125 *ms*, ce qui est beaucoup moins que les 200 *ms* habituellement spécifiées dans les exigences spéciales de protection basées sur l'intervention d'Hydro-Québec [128]. De plus, nous prévoyons améliorer le temps d'apprentissage de l'approche SLOD notamment en effectuant la décomposition par pattern d'échantillons réguliers. Plusieurs nouveaux dictionnaires de stabilité couplés à la *global matching pursuit* (GMP) seront implémentés sur python à partir des algorithmes: *recursive least squares dictionary learning* (RLS-DLA), *online dictionary learning* (ODL), *method of optimized directions* (MOD), la *sparse modeling software* (SPAMS).

Références Bibliographiques

- [1] <https://www.iea.org/reports/world-energy-outlook-2021> (consulter le 17/01/2022)
- [2] Zimran Rafique, Haris M. Khalid, S.M. Muyeen, Innocent Kamwa, "Bibliographic review on power system oscillations damping: An era of conventional grids and renewable energy integration," *International Journal of Electrical Power & Energy Systems*, Volume 136, 2022,107556.
- [3] Suprême, Hussein (2017). Détection en temps réel de l'instabilité transitoire des réseaux électriques avec les mesures synchronisées de phaseurs. *Thèse de doctorat en génie électrique*, Montréal, École de technologie supérieure.
- [4] Yanzhen Zhou, Qinglai Guo, Hongbin Sun, Zhihong Yu, Junyong Wu, Liangliang Hao, "A novel data-driven approach for transient stability prediction of power systems considering the operational variability," *International Journal of Electrical Power & Energy Systems*, Volume 107, 2019, Pages 379-394.
- [5] Zhao, Junbo, et al., Power system dynamic state and parameter estimation-transition to power electronics-dominated clean energy systems: *IEEE task force on power system dynamic state and parameter estimation* (IEEE Power and Energy Society Resource Center), IEEE PES, 90pp, 2021.
- [6] <https://aemo.com.au/-/media/files/major-publications/qed/2021/q3-report.pdf?la=en&hash=F7831B51290237F9033B5D22E52EF4C1> (consulter le 17/01/2022)
- [7] Chakravorti, N. R. Nayak, R. Bisoi, P. K. Dash, and L. Tripathy, "A new robust kernel ridge regression classifier for islanding and power quality disturbances in a multi distributed generation based microgrid," *Renew. Energy Focus*, vol. 28, pp. 78–99, Mar. 2019.
- [8] Z. Ma, C. Zhang and C. Qian, "Review of Machine Learning in Power System," *2019 IEEE Innovative Smart Grid Technologies - Asia (ISGT Asia)*, Chengdu, China, 2019, pp. 3401-3406.

- [9] W. Wang et al., "Multiple event detection and recognition through sparse unmixing for high-resolution situational awareness in power grid," *IEEE Trans. Smart Grid*, vol. 5, no. 4, pp. 1654–1664, Jul. 2014.
- [10] Zhang, Shitu, Zhixun Zhu, and Yang Li. 2021. "A Critical Review of Data-Driven Transient Stability Assessment of Power Systems: Principles, Prospects and Challenges" *Energies* 14, no. 21: 7238.
- [11] M. S. Manikandan, S. R. Samantaray, and I. Kamwa, "Detection and classification of power quality disturbances using sparse signal decomposition on hybrid dictionaries," *IEEE Transactions on Instrumentation and Measurement*, vol. 64, no. 1, pp. 27–38, Jan. 2015.
- [12] M. Babakmehr, H. Sartipizadeh, and M. G. Simões, "Compressive informative sparse representation-based power quality events classification," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 2, pp. 909–921, Feb. 2020.
- [13] Aanand Kumbhar, Pravin G. Dhawale, Shobha Kumbhar, Uday Patil, Pravin Magdum, "A comprehensive review: Machine learning and its application in integrated power system," *Energy Reports*, Volume 7, 2021, Pages 5467-5474.
- [14] A. Alimi, K. Ouahada and A. M. Abu-Mahfouz, "A Review of Machine Learning Approaches to Power System Security and Stability," in *IEEE Access*, vol. 8, pp. 113512-113531, June 2020.
- [15] F. Luo *et al.*, "Advanced Pattern Discovery-based Fuzzy Classification Method for Power System Dynamic Security Assessment," in *IEEE Transactions on Industrial Informatics*, vol. 11, no. 2, pp. 416-426, April 2015.
- [16] D. R. Ostojic, "Spectral monitoring of power system dynamic performances," *IEEE Trans. Power Systems*, vol. 8, no. 2, pp. 445-451, May 1993.
- [17] I. Kamwa, R. Grondin, and L. Loud, "Time-varying contingency screening for dynamic security assessment using intelligent-systems techniques," *IEEE Trans. Power Systems*, vol. 16, no. 3, pp. 526-536, Aug. 2001.
- [18] R. Yadav, A. K. Pradhan, and I. Kamwa, "A spectrum similarity approach for identifying coherency change patterns in power system due to variability in renewable generation," *IEEE Trans. Power Systems*, vol.34, no.5, pp. 3769 – 3779, Sept. 2019.

- [19] S. M. Rovnyak, D. W. Longbottom, D. C. Vasquez, and M. N. Nilchi, "Response-based event detection for one shot wide-area stability controls," *Monitoring and Control using Synchrophasors in Power Systems with Renewables*, eds. I. Kamwa, C. Lu, L. Zhu, The Institution of Engineering and Technology Press, July 2020, Chap 8, pp.1-24.
- [20] V. Malbasa, C. Zheng, P. Chen, T. Popovic and M. Kezunovic, "Voltage Stability Prediction Using Active Machine Learning," in *IEEE Transactions on Smart Grid*, vol. 8, no. 6, pp. 3117-3124, Nov. 2017.
- [21] T. Amraee and S. Ranjbar, "Transient Instability Prediction Using Decision Tree Technique," in *IEEE Trans. on Power Syst.*, vol. 28, no. 3, pp. 3028-3037, Aug. 2013.
- [22] B. Li, J. Wu, L. Hao, M. Shao, R. Zhang and W. Zhao, "Anti-Jitter and Refined Power System Transient Stability Assessment Based on Long-Short Term Memory Network," in *IEEE Access*, vol. 8, pp. 35231-35244, Feb. 2020.
- [23] Y. Xu, Z. Y. Dong, J. H. Zhao, P. Zhang, and K. P. Wong, "A reliable intelligent system for real-time dynamic security assessment of power systems," *IEEE Trans. Power Syst.*, vol. 27, no. 3, pp. 1253–1263, Aug. 2012.
- [24] A. D. Rajapakse, F. Gomez, K. Nanayakkara, P. A. Crossley and V. V. Terzija, "Rotor Angle Instability Prediction Using Post-Disturbance Voltage Trajectories," in *IEEE Trans. on Power Syst.*, vol. 25, no. 2, pp. 947-956, May 2010.
- [25] I. Kamwa, S. R. Samantaray, and G. Joos, "On the accuracy versus transparency trade-off of data-mining models for fast-response PMU-based catastrophe predictors," *IEEE Trans. Smart Grid*, 3 (1), pp. 152 – 161, Mar. 2012.
- [26] J. James, D. J. Hill, A. Y. Lam, J. Gu, and V. O. Li, "Intelligent time adaptive transient stability assessment system," *IEEE Transactions on Power Systems*, vol. 33, no. 1, pp. 1049–1058, Jan. 2018. (Ref 42)
- [27] B. D. Fulcher and N. S. Jones, "Hctsa: A computational framework for automated time-series phenotyping using massive feature extraction," *Cell Syst*, vol 5, issue 5, pp 527-531, Nov. 2017.
- [28] Michael Elad, *Sparse and redundant representations: From theory to application in signal and image processing*. New York, Springer-Verlag, 2010, pp. 1-376.

- [29] W. Li, J. Tang, J. Ma, and Y. Liu, "Online detection of start time and location for hypocenter in north american power grid," *IEEE Trans. Smart Grid*, vol. 1, no. 3, pp. 253–260, 2010.
- [30] R. Rubinstein, A. M. Bruckstein, and M. Elad, "Dictionaries for sparse representation modeling," in *Proceedings of the IEEE*, vol. 98, no. 6, pp. 1045–1057, April 2010.
- [31] Karl Skretting (2002). Sparse signal representation using overlapping frames [Thèse de doctorat]. Stavanger College University.
- [32] R. Hooshmand and A. Enshae, "Detection and classification of single and combined power quality disturbances using fuzzy systems oriented by particle swarm optimization algorithm," *Electric Power Syst. Res.*, vol. 80, no. 12, pp. 1552–1561, 2010.
- [33] L. Cheng, L. Wang, and F. Gao, "Power system fault classification method based on sparse representation and random dimensionality reduction projection," *IEEE Power & Energy Society General Meeting*, pp. 1–5, Oct. 2015.
- [34] Q. Zhao, J. Dong, T. Xia, and Y. Liu, "Detection of the start of frequency excursion in wide-area measurements," in *Proc. IEEE Power Energy Soc. General Meeting*, 2008, pp. 1–6.
- [35] Dabou, Raoult T., Innocent Kamwa, Jacques Tagoudjeu, and Francis C. Mugombozi. 2021. "Sparse Signal Reconstruction on Fixed and Adaptive Supervised Dictionary Learning for Transient Stability Assessment" *Energies* 14, no. 23: 7995.
- [36] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, Nov. 2006.
- [37] A. M. Bruckstein, D. L. Donoho, and M. Elad, B. From sparse solutions of systems of equations to sparse modeling of signals and images, *SIAM Review*, vol. 51, no. 1, pp. 34–81, 2009.
- [38] W. James Murdoch, Chandan Singh, Karl Kumbier, Reza Abbasi-Asl, Bin Yu, "Definitions, methods, and applications in interpretable machine learning," *Proceedings of the National Academy of Sciences*, Oct 2019, 116 (44) 22071-22080.

- [39] I. Kamwa, S. R. Samantaray, and G. Joos, "Catastrophe predictors from ensemble decision-tree learning of wide-area severity indices," *IEEE Trans. Smart Grid*, vol. 1, no. 2, pp. 144-158, Sept. 2010.
- [40] Dabou, R.T., I. Kamwa, J. Tagoudjeu, and C. F. Mugombozi. Supervised learning of overcomplete dictionaries for on-line response based transient stability prediction. *IEEE Transactions on Power Systems* (In first revision 03/01/2022).
- [41] W. Medjroubi and C. Matke, "SciGIRD: an Open Source Reference Model of European Transmission Networks for Scientific Analysis," 2015. [Online]. Available: <https://www.researchgate.net/publication/310773975>. (consulter: Octobre 2021).
- [42] J. Zhao *et al.*, "Roles of Dynamic State Estimation in Power System Modeling, Monitoring and Operation," in *IEEE Transactions on Power Systems*, vol. 36, no. 3, pp. 2462-2472, May 2021.
- [43] A. Moeini, I. Kamwa, P. Brunelle and G. Sybille, "Open data IEEE test systems implemented in SimPowerSystems for education and research in power grid dynamics and control," *2015 50th International Universities Power Engineering Conference (UPEC)*, pp. 1–6, 2015.
- [44] Y. Chao, Z. Xiangjun, and X. Yunfeng, "Improved algorithm for fault location in distribution network with distributed generations," in *Proc. Int. Conf. Intelligent Computation Technology and Automation*, Oct. 2008, vol. 2, pp. 893–896.
- [45] R. T. Dabou and I. Kamwa, "Rapid design method for generating power system stability databases in sps for machine learning," *2020 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, 2020, pp. 1–6.
- [46] S. M. Brahma, "Fault Location in Power Distribution System With Penetration of Distributed Generation," in *IEEE Transactions on Power Delivery*, vol. 26, no. 3, pp. 1545-1553, July 2011, doi: 10.1109/TPWRD.2011.2106146.
- [47] Roh, Y Heo, G and Whang, S E "A Survey on Data Collection for Machine Learning: a Big Data - AI Integration Perspective," *IEEE Trans. On Knowledge and Data Engineering*, October 2019.
- [48] H. I. Fawaz *et al.*, "Deep learning for time series classification: a review," *Data Min Knowl Disc* 33, 917–963 (2019).

- [49] S. K. Azman, Y. J. Isbeih, M. S. El Moursi, and K. Elbassioni, "A unified online deep learning prediction model for small signal and transient stability," *IEEE Trans. Power Systems*, June 2020.
- [50] Miraftabzadeh, S.M.; Longo, M.; Foiadelli, F.; Pasetti, M.; Igual, R. Advances in the application of machine learning techniques for power system analytics: A Survey. *Energies* 2021, *14*, 4776.
- [51] Labeled, I.; Labeled, D. Extreme learning machine-based alleviation for overloaded power system. *IET Gener. Transm. Distrib.* 2019, *13*, 5058–5070.
- [52] H. Liao, J. V. Milanović, M. Rodrigues and A. Shenfield, "Voltage sag estimation in sparsely monitored power systems based on deep learning and system area mapping," in *IEEE Transactions on Power Delivery*, vol. 33, no. 6, pp. 3162–3172, Dec. 2018.
- [53] B. Farsi, M. Amayri, N. Bouguila and U. Eicker, "On short-term load forecasting using machine learning techniques and a novel parallel deep lstm-cnn approach," in *IEEE Access*, vol. 9, pp. 31191-31212, 2021.
- [54] S. Dutta and T. J. Overbye, "Feature extraction and visualization of power system transient stability results," in *IEEE Transactions on Power Systems*, vol. 29, no. 2, pp. 966–973, March 2014.
- [55] C. Zhang, Y. Li, Z. Yu and F. Tian, "Feature selection of power system transient stability assessment based on random forest and recursive feature elimination," *2016 IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC)*, 2016, pp. 1264–1268.
- [56] M. He, J. Zhang and V. Vittal, "Robust online dynamic security assessment using adaptive ensemble decision-tree learning," in *IEEE Transactions on Power Systems*, vol. 28, no. 4, pp. 4089–4098, Nov. 2013.
- [57] G. Ravikumar and S. A. Khaparde, "Taxonomy of PMU data based catastrophic indicators for power system stability assessment," in *IEEE Systems Journal*, vol. 12, no. 1, pp. 452–464, March 2018.
- [58] Y. Zhou, H. Sun, Q. Guo, B. Xu, J. Wu and L. Hao, "Data driven method for transient stability prediction of power systems considering incomplete

measurements," *2017 IEEE Conference on Energy Internet and Energy System Integration (EI2)*, 2017, pp. 1-6.

[59] Q. Zhu, J. Dang, and J. Chen, "A method for power system transient stability assessment based on deep belief networks," *Chin. Soc. Elect. Eng.*, vol. 38, no. 3, pp. 735–743, 2018.

[60] R. Zhang, Y. Xu, Z. Y. Dong, and K. P. Wong, "Post-disturbance transient stability assessment of power systems by a self-adaptive intelligent system," *IET Gener., Transmiss. Distrib.*, vol. 9, no. 3, pp. 296–305, 2015.

[61] L. Zhu, D. J. Hill and C. Lu, "Data-Driven Fast Transient Stability Assessment Using (Fault-on + 2) Generator Trajectories," *2019 IEEE Power & Energy Society General Meeting (PESGM)*, pp. 1-5, Aug. 2019.

[62] Y. Zhang, Y. Xu, R. Zhang and Z. Y. Dong, "A Missing-Data Tolerant Method for Data-Driven Short-Term Voltage Stability Assessment of Power Systems," in *IEEE Transactions on Smart Grid*, vol. 10, no. 5, pp. 5663-5674, Sept. 2019.

[63] Y. Luo, et al. "Data-driven short-term voltage stability assessment based on spatial-temporal graph convolutional network," *International Journal of Electrical Power & Energy Systems* 130, Mar. 2021.

[64] F. Aboytes and R. Ramirez, "Transient stability assessment in longitudinal power systems using artificial neural networks," *IEEE Transactions on Power Systems*, vol. 11, no. 4, pp. 2003–2010, Nov. 1996.

[65] N. Amjady and S. F. Majedi, "Transient stability prediction by a hybrid intelligent system," *IEEE Transactions on Power Systems*, vol. 22, no. 3, pp. 1275–1283, Aug. 2007.

[66] B. Tan, J. Yang, X. Pan, J. Li, P. Xie, and C. Zeng, "Representational learning approach for power system transient stability assessment based on convolutional neural network," *JEng.*, vol. 2017, no. 13, pp. 1847–1850, Jan. 2017.

[67] L. Zheng, W. Hu, K. Hou, X. Xu, and G. Shao, "Real-time transient stability assessment based on deep recurrent neural network," in *Proc. IEEE Innovative Smart Grid Technol.-Asia*, pp. 1–5, Dec. 2017.

- [68] Q. Zhu *et al.*, “A deep end-to-end model for transient stability assessment with pmu data,” *IEEE Access*, vol. 6, pp. 65 474–65 487, Dec. 2018.
- [69] R. Yan, G. Geng, Q. Jiang, and Y. Li, “Fast transient stability batch assessment using cascaded convolutional neural networks,” *IEEE Transactions on Power Systems*, vol. 34, no. 4, pp. 2802–2813, Jul. 2019.
- [70] L. Liu, Y. Li, Y. Cao, F. Liu, W. Wang, and J. Zuo, “Transient rotor angle stability prediction based on deep belief network and long short-term memory network,” *IFAC-Papers On Line*, vol. 52, no. 4, pp. 176–181, Sep. 2019.
- [71] B. Tan, J. Yang, Y. Tang, S. Jiang, P. Xie, and W. Yuan, “A deep imbalanced learning framework for transient stability assessment of power system,” *IEEE Access*, vol. 7, pp. 81 759–81 769, June 2019.
- [72] Z. Jiang, Z. Lin and L. S. Davis, “Label Consistent K-SVD: Learning a discriminative dictionary for recognition,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 35, no. 11, pp. 2651–2664, Nov. 2013.
- [73] R. Rubinstein, T. Peleg and M. Elad, “Analysis K-SVD: A Dictionary-Learning algorithm for the analysis sparse model,” in *IEEE Trans. on Signal Process.*, vol. 61, no. 3, pp. 661–677, 2013.
- [74] Z. Jiang, Z. Lin and L. S. Davis, “Learning a discriminative dictionary for sparse coding via label consistent K-SVD,” *CVPR 2011*, Colorado Springs, CO, USA, pp. 1697–1704, 2011.
- [75] D. L. Donoho and M. Elad, “Optimally sparse representation in general (non-orthogonal) dictionaries via l_1 minimization,” *Proc. Nat. Acad. Sci.*, vol. 100, no. 5, pp. 2197–2202, 2003.
- [76] T. T. Cai and L. Wang, “Orthogonal matching pursuit for sparse signal recovery with noise,” *IEEE Trans. on Information Theory*, vol. 57, no. 7, pp. 4680–4688, July 2011.
- [77] Z. Zhang *et al.*, “Jointly learning structured analysis discriminative dictionary and analysis multiclass classifier,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 8, pp. 3798–3814, Aug. 2018.

- [78] M. A. S. Masoum, S. Jamali, and N. Ghaffarzadeh, "Detection and classification of power quality disturbances using discrete wavelet transform and wavelet networks," *IET Sci. Meas. Technol.*, vol. 4, no. 4, pp. 193–205, Jul. 2010.
- [79] M. Babakmehr, M. Majidi, and M. G. Simoes, "Compressive sensing for power system data analysis," in *Big Data Application in Power Systems*. Amsterdam, The Netherlands: Elsevier, 2018, pp. 159–178, ch. 8.
- [80] Z. Zhang, F. Li, T. W. S. Chow, L. Zhang and S. Yan, "Sparse codes auto-extractor for classification: a joint embedding and dictionary learning framework for representation," *IEEE Trans. on Signal Process.*, vol. 64, no. 14, pp. 3790–3805, 2016.
- [81] W. M. Jiang, Z. Zhang, F. Z. Li, L. Zhang, M. B. Zhao, and X. H. Jin, "Joint label consistent dictionary learning and adaptive label prediction for semi-supervised machine fault classification," *IEEE Trans. Ind. Inform.*, vol. 12, no. 1, pp. 248–256, Feb. 2016.
- [82] L. Cheng, C. You and L. Chen, "Identification of Power Line Outages Based on PMU Measurements and Sparse Overcomplete Representation," *2016 IEEE 17th International Conference on Information Reuse and Integration (IRI)*, 2016, pp. 343–349.
- [83] Y. Song, W. Wang, Z. Zhang, H. Qi and Y. Liu, "Multiple event detection and recognition for large-scale power systems through cluster-based sparse coding," *IEEE Trans. on Power Systems*, vol. 32, no. 6, pp. 4199–4210, Nov. 2017.
- [84] H. Zhu and G. B. Giannakis, "Sparse overcomplete representations for efficient identification of power line outages," *IEEE Trans. Power Syst.*, vol. 27, no. 4, pp. 2215–2224, Nov. 2012.
- [85] Y. Qin, J. Zou, B. Tang, Y. Wang and H. Chen, "Transient Feature Extraction by the Improved Orthogonal Matching Pursuit and K-SVD Algorithm With Adaptive Transient Dictionary," in *IEEE Trans. on Ind. Inform.*, vol. 16, no. 1, 215–227, 2020.
- [86] L. Zheng *et al.*, "Deep belief network based nonlinear representation learning for transient stability assessment," *IEEE Power & Energy Society General Meeting*, pp. 1–5, 2017.

- [87] P. Kathirvel, M. S. Manikandan, P. Maya and K. P. Soman, "Detection of power quality disturbances with overcomplete dictionary matrix and ℓ_1 -norm minimization," *International Conference on Power and Energy Systems*, Chennai, 2011.
- [88] D. Cai, K. Li, S. He, Y. Li and Y. Luo, "A highly accurate and fast power quality disturbances classification based on dictionary learning sparse decomposition," *Transactions of the Institute of Measurement and Control.*, pp. 145–155, 2019.
- [89] T. X. Zhu, "Detection and characterization of oscillatory transients using matching pursuits with a damped sinusoidal dictionary," in *IEEE Transactions on Power Delivery*, vol. 22, no. 2, pp. 1093–1099, April 2007.
- [90] C. liu and J. Liu, "Research on power quality signals reconstruction method based on K-SVD dictionary learning," *2020 39th Chinese Control Conference (CCC)*, pp. 2930–2934, 2020.
- [91] J. Zhao, J. Qi, Z. Huang, A. P. S. Meliopoulos, A. Gomez-Exposito, M. Netto, L. Mili, A. Abur, V. Terzija, I. Kamwa, B. Pal, and A. K. Singh, "Power system dynamic state estimation: Motivations, definitions, methodologies, and future work," *IEEE Trans. Power Syst.*, vol. 34, no. 4, pp. 3188_3198, Jul. 2019.
- [92] B. D Fulcher, M. A. Little, and N. S. Jones "Highly comparative time-series analysis: the empirical structure of time series and their methods," *Journal Royal Society Interface* 10, Mar. 2013.
- [93] P. Kundur, *Power System Stability and Control*. New York, NY, USA: McGraw-Hill, 1994.
- [94] A. P. Meliopoulos et al., "Dynamic state estimation based protection: Status and promise," *IEEE Trans. Power Del.*, vol. 32, no. 1, pp. 320–330, Feb. 2017
- [95] A. Abur and A. G. Exposito. *Power System State Estimation: Theory and Application*. New York, NY, USA: Marcel Dekker, 2004.
- [96] R. T. Dabou, I. Kamwa, C. Y. Chung and C. F. Mugombozi, "Time series-analysis based engineering of high-dimensional wide-area stability indices for machine learning," in *IEEE Access*, vol. 9, pp. 104927-104939, 2021.

- [97] M. Pavella, D. Ernst, and D. Ruiz-Vega, *Transient stability of power systems: A unified approach to assessment and control*. Boston, MA: Springer, 2000, pp. 1-256.
- [98] A. Paul, I. Kamwa, and G. Joos, "PMU Signals responses-based ras for instability mitigation through on-the fly identification and shedding of the run-away generators," *IEEE Trans. Power Systems*, vol. 35, no. 3, pp. 1707–1717, May 2020.
- [99] S. Dasgupta, M. Paramasivam, U. Vaidya and V. Ajjarapu, "Real-Time monitoring of short-Term voltage stability using PMU Data," *IEEE Trans. Power Systems*, vol. 28, no. 4, pp. 3702–3711, Nov. 2013.
- [100] P. Fahim and A. Karimpour, "A Method Based on Fast Fourier Transform for Online Supervising of Power System and Control Structure Design," *7th Iran Wind Energy Conference (IWEC2021)*, pp. 1-6, 2021.
- [101] S. A. Fulop and K. Fitz, "Algorithms for computing the time-corrected instantaneous frequency (reassigned) spectrogram, with applications," *Journal of the Acoustical Society of America*. vol. 119, pp. 360–371, Jan. 2006.
- [102] K. Barbé, R. Pintelon, and J. Schoukens, "Welch method revisited: nonparametric power spectrum estimation via circular overlap," *IEEE Trans. Signal Processing*, vol. 58, no. 2, pp. 553–565, Feb. 2010.
- [103] L. Wei and H. Put, "Application of wavelet transformation for voltage stability in distributed power system network," *Chinese Control and Decision Conference*, Yantai, Shandong, pp. 116–119, Aug. 2008.
- [104] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. New York, NY, USA: Springer-Verlag, 2008.
- [105] F. R. Gomez, A. D. Rajapakse, U. D. Annakkage, and I. T. Fernando, "Support vector machine-based algorithm for post-fault transient stability status prediction using synchronized measurements," *IEEE Trans. Power Systems*, vol. 26, no. 3, pp. 1474–1483, Aug. 2011.
- [106] V. Krishnan, J. D. McCalley, S. Henry and S. Issad, "Efficient Database Generation for Decision Tree Based Power System Security Assessment," in *IEEE Transactions on Power Systems*, vol. 26, no. 4, pp. 2319-2327, Nov. 2011.

- [107] B. H. Menze *et al.*, "A comparison of random forest and its Gini importance with standard chemometric methods for the feature selection and classification of spectral data," *BMC Bioinformatics* 10,213, July 2009.
- [108] Breiman L: Random forests. *J Mach Learn* 2001, 45: 5–32. 10.1023/A:1010933404324
- [109] A. Gupta, G. Gurralla and P. S. Sastry, "An Online Power System Stability Monitoring System Using Convolutional Neural Networks," in *IEEE Transactions on Power Systems*, vol. 34, no. 2, pp. 864-872, March 2019.
- [110] L. Shaowen and C. Yong, "A Kind of Improved AdaBoost Algorithm," *2014 7th International Conference on Intelligent Computation Technology and Automation*, pp. 16–18, 2014.
- [111] T. Miki, R. Sugino, and Y. Kono, "The efficient generation method of critical fault clearing time functions for power system transient stability assessment," *IEEE Transactions on Computers*, vol. 2, no 1, pp. 116–121, Jan. 2003.
- [112] T. Miki, D.Okitsu, E.Takashima, Y.Abe and M.Tano, "Power System Transient Stability Assessment Using Critical Fault Clearing Time Functions," *IEEE/PES Transmission and Distribution Conference and Exhibition 2002 : Asia Pasific*, Vol.3, 2002, pp. 1514-1517.
- [113] Liu, Y.; Fan, R.; Terzija, V. Power system restoration: a literature review from 2006 to 2016. *Journal of Modern Power Systems and Clean Energy*, 2016, 4, 332–341.
- [114] Caiafa, Cesar F., Jordi Solé-Casals, Pere Marti-Puig, Sun Zhe, and Toshihisa Tanaka. 2020. "Decomposition Methods for Machine Learning with Small, Incomplete or Noisy Datasets" *Applied Sciences* 10, no. 23: 8481.
- [115] I. Kamwa, S. R. Samantaray, and G. Joos, "Development of rule-based classifiers for rapid stability assessment of wide-area post-disturbance records," *IEEE Transactions on Power System*, vol. 24, no. 1, pp. 258–270, Feb. 2009.
- [116] V. Raghavan, K. M. M. Prabhu and P. C. W. Sommen, "An analysis of real-Fourier domain-based adaptive algorithms implemented with the Hartley transform using cosine-sine symmetries," in *IEEE Transactions on Signal Processing*, vol. 53, no. 2, pp. 622-629, Feb. 2005.

- [117] H. V. Soenson and C. S. Burrus, "Efficient computation of the DFT with only a subset of input or output points," *IEEE Trans. Signal Processing*, vol. 41, no. 3, pp. 1184–1199, Mar. 1993.
- [118] S. Aslan, H. Badem, D. Karaboga, A. Basturk and T. Ozcan, "Accelerating Discrete Haar Wavelet Transform on GPU cluster," *2015 9th International Conference on Electrical and Electronics Engineering (ELECO)*, 2015, pp. 1237-1240,
- [119] F. Arb, S. M. Daud and S. Z. Hashim, "Discrete Wavelet Transform Domain Techniques," *2013 International Conference on Informatics and Creative Multimedia*, 2013, pp. 340-345.
- [120] S. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Trans. Signal Process.*, vol. 41, no. 12, pp. 3397–3415, 1993.
- [121] Y. A. Reznik, "Relationship between DCT-II, DCT-VI, and DST-VII transforms," *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 5642-5646, 2013.
- [122] R. I. iego, J. Barros, M. de Apráiz and J. Prieto, "A simple Discrete Fourier Transform-based synchronization method for power quality measurement," *2013 IEEE Grenoble Conference*, 2013, pp. 1-6.
- [123] J. A R. Macias and A. G. Exposito, "Efficient computation of the running discrete Haar transform," in *IEEE Transactions on Power Delivery*, vol. 21, no. 1, pp. 504-505, Jan. 2006.
- [124] Ma, B.; Zou, X. "Discrete wavelet transform for signal processing in weight-in-motion system," *International Conference on Electrical and Control Engineering*, 2010, 4668–4671.
- [125] J. A. Tropp, "Topics in sparse approximation," *Ph.D. dissertation*, University of Texas at Austin, Austin, 2004.
- [126] A. J. Weinstein and M. B. Wakin, "Online search Orthogonal Matching Pursuit," *2012 IEEE Statistical Signal Processing Workshop (SSP)*, 2012, pp. 584-587
- [127] J. A. Trpp, "Greed is Good: Algorithmic Results for Sparse Approximation," *IEEE Trans. on Info. Theory*, vol. 50, no. 10, pp. 2231–2242, Oct. 2004.

- [128] A. Heniche, I. Kamwa, and M. Dobrescu, “Hydro-Québec’s defense plan: Present and future,” *2013 IEEE Power & Energy Society General Meeting*, pp. 1–5, July 2013.

Annexes

A.1 Données des réseaux IEEE 2 Area 4 Machines, IEEE 39 – bus et IEEE 68 –bus

A.1.1 Données IEEE 2 Area 4 Machines

BUS DATA FOLLOWS										9 ITEMS							
1	Bus1	16.5kV	1	1	3	1.040	0.0	0.0	0.0	0.0	16.5	1.040	0.0	0.0	0.0	0	
2	Bus2	18kV	1	1	2	1.025	0.0	0.0	0.0	163.0	0.0	18.0	1.025	99.0	-99.0	0.0	0
3	Bus3	13.8kV	1	1	2	1.025	0.0	0.0	0.0	85.0	0.0	13.8	1.025	99.0	-99.0	0.0	0
4	Bus4	230kV	1	1	0	1.000	0.0	0.0	0.0	0.0	230.0	0.000	0.0	0.0	0.0	0	
5	Bus5	230kV	1	1	0	1.000	0.0	125.0	50.0	0.0	0.0	230.0	0.000	0.0	0.0	0.0	0
6	Bus6	230kV	1	1	0	1.000	0.0	90.0	30.0	0.0	0.0	230.0	0.000	0.0	0.0	0.0	0
7	Bus7	230kV	1	1	0	1.000	0.0	0.0	0.0	0.0	230.0	0.000	0.0	0.0	0.0	0	
8	Bus8	230kV	1	1	0	1.000	0.0	100.0	35.0	0.0	0.0	230.0	0.000	0.0	0.0	0.0	0
9	Bus9	230kV	1	1	0	1.000	0.0	0.0	0.0	0.0	230.0	0.000	0.0	0.0	0.0	0	

BRANCH DATA FOLLOWS										9 ITEMS							
1	4	1	1	1	0	0.0000	0.0576	0.0000	0	0	0	0.0	0.0	0.0	0.0	0.0	0.0
2	7	1	1	1	0	0.0000	0.0625	0.0000	0	0	0	0.0	0.0	0.0	0.0	0.0	0.0
3	9	1	1	1	0	0.0000	0.0586	0.0000	0	0	0	0.0	0.0	0.0	0.0	0.0	0.0
4	5	1	1	1	0	0.0100	0.0850	0.1760	0	0	0	0.0	0.0	0.0	0.0	0.0	0.0
4	6	1	1	1	0	0.0170	0.0920	0.1580	0	0	0	0.0	0.0	0.0	0.0	0.0	0.0
5	7	1	1	1	0	0.0320	0.1610	0.3060	0	0	0	0.0	0.0	0.0	0.0	0.0	0.0
6	9	1	1	1	0	0.0390	0.1700	0.3580	0	0	0	0.0	0.0	0.0	0.0	0.0	0.0
7	8	1	1	1	0	0.0085	0.0720	0.1490	0	0	0	0.0	0.0	0.0	0.0	0.0	0.0
8	9	1	1	1	0	0.0119	0.1008	0.2090	0	0	0	0.0	0.0	0.0	0.0	0.0	0.0

A.1.2 Données IEEE 39 -bus

- 1 %% Line data Format (line)
- 2 % All values are given on the same system base MVA
- 3 % 1: From bus
- 4 % 2: To bus
- 5 % 3: Resistance(pu)
- 6 % 4: Reactance(pu)
- 7 % 5: Charge (pu)
- 8 % 6: Transformer Tap Amplitude
- 9 % 7: base MVA

10 % 8: Nomonal Voltage(KV)

11 % 1 2 3 4 5 6 7 8

12 line=[...

13				1	2	0.0035	0.0411	0.6987	0	100	345
14				1	39	0.001	0.025	0.75	0	100	345
15				2	3	0.0013	0.0151	0.2572	0	100	345
16				2	25	0.007	0.0086	0.146	0	100	345
17				2	30	0	0.0181	0	1.025	100	22
18				3	4	0.0013	0.0213	0.2214	0	100	345
19				3	18	0.0011	0.0133	0.2138	0	100	345
20				4	5	0.0008	0.0128	0.1342	0	100	345
21				4	14	0.0008	0.0129	0.1382	0	100	345
22				5	8	0.0008	0.0112	0.1476	0	100	345
23				6	5	0.0002	0.0026	0.0434	0	100	345
24				6	7	0.0006	0.0092	0.113	0	100	345
25				6	11	0.0007	0.0082	0.1389	0	100	345
26				7	8	0.0004	0.0046	0.078	0	100	345
27				8	9	0.0023	0.0363	0.3804	0	100	345
28				9	39	0.001	0.025	1.2	0	100	345
29				10	11	0.0004	0.0043	0.0729	0	100	345
30				10	13	0.0004	0.0043	0.0729	0	100	345
31				10	32	0	0.02	0	1.07	100	22
32				12	11	0.0016	0.0435	0	1.006	100	345
33				12	13	0.0016	0.0435	0	1.006	100	345
34				13	14	0.0009	0.0101	0.1723	0	100	345
35				14	15	0.0018	0.0217	0.366	0	100	345
36				15	16	0.0009	0.0094	0.171	0	100	345
37				16	17	0.0007	0.0089	0.1342	0	100	345
38				16	19	0.0016	0.0195	0.304	0	100	345
39				16	21	0.0008	0.0135	0.2548	0	100	345
40				16	24	0.0003	0.0059	0.068	0	100	345
41				17	18	0.0007	0.0082	0.1319	0	100	345
42				17	27	0.0013	0.0173	0.3216	0	100	345
43				19	33	0.0007	0.0142	0	1.07	100	22
44				19	20	0.0007	0.0138	0	1.06	100	345
45				20	34	0.0009	0.018	0	1.009	100	22
46				21	22	0.0008	0.014	0.2565	0	100	345
47				22	23	0.0006	0.0096	0.1846	0	100	345
48				22	35	0	0.0143	0	1.025	100	22
49				23	24	0.0022	0.035	0.361	0	100	345
50				23	36	0.0005	0.0272	0	1	100	22
51				25	26	0.0032	0.0323	0.513	0	100	345
52				25	37	0.0006	0.0232	0	1.025	100	22
53				26	27	0.0014	0.0147	0.2396	0	100	345
54				26	28	0.0043	0.0474	0.7802	0	100	345

```

55          26  29  0.0057  0.0625  1.029  0  100  345
56          28  29  0.0014  0.0151  0.249  0  100  345
57          29  38  0.0008  0.0156  0  1.025  100  22
58          31   6   0  0.025  0  1  100  22];
59 %% Machine Data Format (mac_con)
60 % 1. Machine Number
61 % 2. Bus Number
62 % 3. Base MVA
63 % 4. Leakage Reactance x_l(pu)
64 % 5. Resistance r_a(pu)
65 % 6. d-axis synchronous reactance x_d(pu)
66 % 7. d-axis transient reactance x'_d(pu)
67 % 8. d-axis subtransient reactance x''_d(pu)
68 % 9. d-axis open-circuit time constant T'_do(sec),
69 % 10. d-axis open-circuit subtransient time constant T''_do(sec)
70 % 11. q-axis synchronous reactance x_q(pu)
71 % 12. q-axis transient reactance x'_q(pu)
72 % 13. q-axis subtransient reactance x''_q(pu)
73 % 14. q-axis open-circuit time constant T'_qo(sec)
74 % 15. q-axis open circuit subtransient time constant % T''_qo(sec)
75 % 16. inertia constant H(sec)
76 % 17. damping coefficient d_o(pu)
77 % 18. damping coefficient d_l(pu)
78 % 19. bus number
79 % Note: all the following machines use sub-transient model
    % 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
80 18 19
81 mac_con=[
    1 39 1000.0 0.350 0.0010 0.200 0.060 0.01 7.000 0.003 0.190 0.080 0.03 0.700 0.005
82 50.00 0.000 0.00 39 ;
    2 31 1000.0 0.350 0.0270 2.950 0.697 0.01 6.560 0.003 2.820 1.7 0.03 1.500 0.005
83 3.030 0.000 0.00 31 ;
    3 32 1000.0 0.304 .00386 2.495 0.531 0.01 5.700 0.003 2.370 0.876 0.03 1.500 0.005
84 3.580 0.000 0.00 32 ;
    4 33 1000.0 0.295 .00222 2.620 0.436 0.01 5.690 0.003 2.580 1.66 0.03 1.500 0.005
85 2.860 0.000 0.00 33 ;
    5 34 1000.0 0.540 0.0014 6.700 1.320 0.01 5.400 0.003 6.200 1.66 0.03 0.440 0.005
86 2.600 0.000 0.00 34 ;
    6 35 1000.0 0.224 0.0615 2.540 0.500 0.01 7.300 0.003 2.410 0.814 0.03 0.400 0.005
87 3.480 0.000 0.00 35 ;
    7 36 1000.0 0.322 .00268 2.950 0.490 0.01 5.660 0.003 2.920 1.86 0.03 1.500 0.005
88 2.640 0.000 0.00 36 ;
    8 37 1000.0 0.280 .00686 2.900 0.570 0.01 6.700 0.003 2.800 0.911 0.03 0.410 0.005
89 2.430 0.000 0.00 37 ;
    9 38 1000.0 0.298 0.0030 2.106 0.570 0.01 4.790 0.003 2.050 0.587 0.03 1.960 0.005
90 3.450 0.000 0.00 38 ;
   10 30 1000.0 0.125 0.0014 1.000 0.310 0.01 10.20 0.003 0.690 0.08 0.03 1.500 0.005
91 4.200 0.000 0.00 30 ];
92 mac_con(:,5)=0;

```

```

p0=[1000 520.81 650 632 508 650 560 540 830 250]'./1000;% Active Power Generation of
93 PV units
94 Pn(1:10,1)=mac_con(:,3)/1000; % Nominal Power
95 %% Power System Stabilizer Format (MB)
96 % Applied power system stabilizer is MBPSS with simplified settings
97 % Note: All machines use MBPSS with same configuration
98 % 1: Global gain (G)
99 % 2: Frequency of low frequency band (FL) Hz
100 % 3: Gain of low frequency band (KL)
101 % 4: Frequency of intermediate frequency band (FI) Hz
102 % 5: Gain of intermediate frequency band (KI)
103 % 6: Frequency of high frequency band (FH) Hz
104 % 7: Gain of high frequency band (KH)
105 % 1 2 3 4 5 6 7
106 MB=[1 0.2 30 1.25 40 12 160];
107 %% Excitation System format (AVR_Data)
% All machines use IEEE type 1 synchronous machine voltage regulator combined to an
108 exciter
109 % 1. Low pass filter time constant (Tr) sec
110 % 2. Regulator gain (Ka)
111 % 3. regulator time constant (Ta) sec
112 % 4. Lead-lag compensator time constant (Tb) sec
113 % 5. Lead-lag compensator time constant (Tc) sec
114 % 6. Terminal voltage (pu)
115 % 7. Lower limit for regulator output (Emin)
116 % 8. Upper limit for regulator output (Emax)
117 % 1 2 3 4 5 6 7 8
118 AVR_Data=[...
119 0.01 200 0.015 10 1 1.03 -5 5
120 0.01 200 0.015 10 1 1.03 -5 5
121 0.01 200 0.015 10 1 1.03 -5 5
122 0.01 200 0.015 10 1 1.03 -5 5
123 0.01 200 0.015 10 1 1.03 -5 5
124 0.01 200 0.015 10 1 1.03 -5 5
125 0.01 200 0.015 10 1 1.03 -5 5
126 0.01 200 0.015 10 1 1.03 -5 5
127 0.01 200 0.015 10 1 1.03 -5 5
128 0.01 200 0.015 10 1 1.03 -5 5];
129 C0=829.7e-9;
130 L0=3.220e-3;
131 R0=1;
132 Ns=120*60/(2); %Nominal speed of synchronous machines
133 s=10;
134 % PSSModel=1;% 1:No pSS //1:MB
135 %% Bus data (Bus)

```



```

136 % 1. Bus number
137 % 2. Nominal phase-phase voltage
138 % 1 2
139 Bus = [ ...
140 1 345
141 2 345
142 3 345
143 4 345
144 5 345
145 6 345
146 7 345
147 8 345
148 9 345
149 10 345
150 11 345
151 12 230
152 13 345
153 14 345
154 15 345
155 16 345
156 17 345
157 18 345
158 19 345
159 20 345
160 21 345
161 22 345
162 23 345
163 24 345
164 25 345
165 26 345
166 27 345
167 28 345
168 29 345
169 30 22
170 31 22
171 32 22
172 33 22
173 34 22
174 35 22
175 36 22
176 37 22
177 38 22
178 39 345];
179 zbase=(line(:,8).^2)./line(:,7);
180 line(:,3)=line(:,3).*zbase;

```

```

181 line(:,4)=line(:,4).*zbase/(120*pi);
182 line(:,5)=line(:,5)./zbase/(120*pi);
183 PSSMODEL=1;

```

A.1.3 Données IEEE 68 -bus

```

1 clear
2 % clc
3 %% Machine
4 % Machine data format
5 % 1. machine number,
6 % 2. bus number,
7 % 3. base mva,
8 % 4. leakage reactance x_l(pu),
9 % 5. resistance r_a(pu),
10 % 6. d-axis synchronous reactance x_d(pu),
11 % 7. d-axis transient reactance x'_d(pu),
12 % 8. d-axis subtransient reactance x''_d(pu),
13 % 9. d-axis open-circuit time constant T'_do(sec),
14 % 10. d-axis open-circuit subtransient time constant T''_do(sec),
15 % 11. q-axis synchronous reactance x_q(pu),
16 % 12. q-axis transient reactance x'_q(pu),
17 % 13. q-axis subtransient reactance x''_q(pu),
18 % 14. q-axis open-circuit time constant T'_qo(sec),
19 % 15. q-axis open circuit subtransient time constant T''_qo(sec),
20 % 16. inertia constant H(sec),
21 % 17. damping coefficient d_o(pu),
22 % 18. damping coefficient d_l(pu),
23 % 19. bus number
24 % note: all the following machines use sub-transient model
25 %      3   4   5   6   7   8   9   10  11  12  13  14  15  16
26 %      Xl  Rl  Xd  X'd  X''d  T'_do  T''_do  Xq  X'q  X''q  T'_qo  T''_qo  H
27 mac_con=[
28  1 53 100 0.0125 0 0.1 0.0310 0.0250 10.2 0.05 0.0690 0.0280 0.0250 1.5 0.0350 2.3333 0 0 53
29  2 54 100 0.0350 0 0.2950 0.0697 0.05 6.56 0.05 0.2820 0.0600 0.05 1.5 0.0350 4.9494 0 0 54
30  3 55 100 0.0304 0 0.2495 0.0531 0.0450 5.7 0.05 0.2370 0.0500 0.0450 1.5 0.0350 4.9623 0 0 55
31  4 56 100 0.0295 0 0.2620 0.0436 0.0350 5.69 0.05 0.2580 0.0400 0.0350 1.5 0.0350 4.1629 0 0 56
32  5 57 100 0.0270 0 0.3300 0.0660 0.0500 5.4 0.05 0.3100 0.0600 0.05 0.44 0.0350 4.7667 0 0 57
33  6 58 100 0.0224 0 0.2540 0.05 0.0400 7.3 0.05 0.2410 0.0450 0.0400 0.4 0.0350 4.9107 0 0 58
34  7 59 100 0.0322 0 0.2950 0.0490 0.0400 5.66 0.05 0.2920 0.0450 0.0400 1.5 0.0350 4.3267 0 0 59
35  8 60 100 0.0280 0 0.2900 0.0570 0.0450 6.7 0.05 0.2800 0.0500 0.0450 0.41 0.0350 3.9150 0 0 60
36  9 61 100 0.0298 0 0.2106 0.0570 0.0450 4.79 0.05 0.2050 0.0500 0.0450 1.96 0.0350 4.0365 0 0 61
37 10 62 100 0.0199 0 0.1690 0.0457 0.0400 10.2 0.05 0.1150 0.0450 0.0400 1.5 0.0350 2.9106 0 0 62
38 11 63 100 0.0103 0 0.1280 0.0180 0.0120 5.4 0.05 0.1230 0.0150 0.0120 1.5 0.0350 2.0053 0 0 63

```

```

39 12 64 100 0.0220 0 0.1010 0.0310 0.0250 7.3 0.05 0.0950 0.0280 0.0250 1.5 0.0350 5.1791 0 0 64
40 13 65 100 0.0030 0 0.0296 0.0055 0.0040 5.66 0.05 0.0286 0.0050 0.0040 1.5 0.0350 4.0782 0 0 65
41 14 66 100 0.0017 0 0.0180 0.0029 0.0023 6.7 0.05 0.0173 0.0025 0.0023 1.5 0.0350 3 0 0 66
42 15 67 100 0.0017 0 0.0180 0.0029 0.0023 4.79 0.05 0.0173 0.0025 0.0023 1.5 0.0350 3 0 0 67
43 16 68 100 0.0041 0 0.0356 0.0071 0.0055 10.2 0.05 0.0334 0.0060 0.0055 1.5 0.0350 4.4500 0 0 68];
44
45 Mbase=[ 300 800 1040 1040 910 900 800 800 1000 1200 1600 1900 12000 10000 10000 11000];
46 p0= [ 250 545 650 632 505.2 700 560 540 800 500 1000 1350 3591 1785 1000 4000]'/1000;%
47 Pn(1:16,1)=Mbase'./1000; % Nominal Power
48 mac_con(:,4:8)=mac_con(:,4:8).*Mbase'./mac_con(:,3);
49 mac_con(:,11:13)=mac_con(:,11:13).*Mbase'./mac_con(:,3);
50 %% Line
51 % From_bus To_bus resistance reactance line_charging tap_ratio Gen_number
52 line = [
53 1 2 0.0035 0.0411 0.6987 0 0. ;%1
54 1 30 0.0008 0.0074 0.48 0 0. ;%2
55 2 3 0.0013 0.0151 0.2572 0 0. ;%3
56 2 25 0.007 0.0086 0.146 0 0. ;%4
57 2 53 0. 0.0181 0. 1.025 1; %5 G1
58 3 4 0.0013 0.0213 0.2214 0. 0. ;%6
59 3 18 0.0011 0.0133 0.2138 0. 0. ;%7
60 4 5 0.0008 0.0128 0.1342 0. 0. ;%8
61 4 14 0.0008 0.0129 0.1382 0. 0. ;%9
62 5 6 0.0002 0.0026 0.0434 0. 0. ;%10
63 5 8 0.0008 0.0112 0.1476 0. 0. ;%11
64 6 7 0.0006 0.0092 0.1130 0. 0. ;%12
65 6 11 0.0007 0.0082 0.1389 0. 0. ;%13
66 6 54 0. 0.0250 0. 1.07 2; %14 G2
67 7 8 0.0004 0.0046 0.178 0. 0. ;%15
68 8 9 0.0023 0.0363 0.3804 0. 0. ;%16
69 9 30 0.0019 0.0183 0.29 0. 0. ;%17
70 10 11 0.0004 0.0043 0.0729 0. 0. ;%18
71 10 13 0.0004 0.0043 0.0729 0. 0. ;%19
72 10 55 0. 0.02 0. 1.07 3; %20 G3
73 12 11 0.0016 0.0435 0.0 1.06 0. ;%21*
74 12 13 0.0016 0.0435 0.0 1.06 0. ;%22*
75 13 14 0.0009 0.0101 0.1723 0. 0. ;%23
76 14 15 0.0018 0.0217 0.366 0. 0. ;%24
77 15 16 0.0009 0.0094 0.171 0. 0. ;%25
78 16 17 0.0007 0.0089 0.1342 0. 0. ;%26
79 16 19 0.0016 0.0195 0.3040 0. 0. ;%27
80 16 21 0.0008 0.0135 0.2548 0. 0. ;%28
81 16 24 0.0003 0.0059 0.0680 0. 0. ;%29
82 17 18 0.0007 0.0082 0.1319 0. 0. ;%30
83 17 27 0.0013 0.0173 0.3216 0. 0. ;%31

```

84 19 20 0.0007 0.0138 0.0 1.06 0.%;%32*
85 19 56 0.0007 0.0142 0. 1.07 4.%;%33 G4
86 20 57 0.0009 0.0180 0. 1.009 5.%;%34 G5
87 21 22 0.0008 0.0140 0.2565 0. 0.%;%35
88 22 23 0.0006 0.0096 0.1846 0. 0.%;%36
89 22 58 0. 0.0143 0. 1.025 6.%;%37 G6
90 23 24 0.0022 0.0350 0.3610 0. 0.%;%38
91 23 59 0.0005 0.0272 0. 1 7.%;%39 G7 %tap modified 0=>1
92 25 26 0.0032 0.0323 0.5310 0. 0.%;%40
93 25 60 0.0006 0.0232 0. 1.025 8.%;%41 G8
94 26 27 0.0014 0.0147 0.2396 0. 0.%;%42
95 26 28 0.0043 0.0474 0.7802 0. 0.%;%43
96 26 29 0.0057 0.0625 1.0290 0. 0.%;%44
97 28 29 0.0014 0.0151 0.2490 0. 0.%;%45
98 29 61 0.0008 0.0156 0. 1.025 9.%;%46 G9
99 9 30 0.0019 0.0183 0.29 0. 0.%;%47
100 9 36 0.0022 0.0196 0.34 0. 0.%;%48
101 9 36 0.0022 0.0196 0.34 0. 0.%;%49
102 36 37 0.0005 0.0045 0.32 0. 0.%;%50
103 34 36 0.0033 0.0111 1.45 0. 0.%;%51
104 35 34 0.0001 0.0074 0.0 0.946 0.%;%52*
105 33 34 0.0011 0.0157 0.202 0. 0.%;%53
106 32 33 0.0008 0.0099 0.168 0. 0.%;%54
107 30 31 0.0013 0.0187 0.333 0. 0.%;%55
108 30 32 0.0024 0.0288 0.488 0. 0.%;%56
109 1 31 0.0016 0.0163 0.25 0. 0.%;%57
110 31 38 0.0011 0.0147 0.247 0. 0.%;%58
111 33 38 0.0036 0.0444 0.693 0. 0.%;%59
112 38 46 0.0022 0.0284 0.43 0. 0.%;%60
113 46 49 0.0018 0.0274 0.27 0. 0.%;%61
114 1 47 0.0013 0.0188 1.31 0. 0.%;%62
115 47 48 0.0025 0.0268 0.40 0. 0.%;%63
116 47 48 0.0025 0.0268 0.40 0. 0.%;%64
117 48 40 0.0020 0.022 1.28 0. 0.%;%65
118 35 45 0.0007 0.0175 1.39 0. 0.%;%66
119 37 43 0.0005 0.0276 0.4 0 0.%;%67*
120 43 44 0.0001 0.0011 0.4 0 0.%;%68*
121 44 45 0.0025 0.073 0.4 0 0.%;%69*
122 39 44 0.000 0.0411 0.4 0 0.%;%70*
123 39 45 0.000 0.0839 0.4 0 0.%;%71*
124 45 51 0.0004 0.0105 0.72 0. 0.%;%72
125 50 52 0.0012 0.0288 2.06 0. 0.%;%73
126 50 51 0.0009 0.0221 1.62 0. 0.%;%74
127 49 52 0.0076 0.1141 1.16 0. 0.%;%75
128 52 42 0.0040 0.0600 2.25 0. 0.%;%76

```

129 42 41 0.0040 0.0600 2.25 0. 0.;%77
130 41 40 0.0060 0.0840 3.15 0. 0.;%78
131 31 62 0. 0.026 0. 1.04 10;%79 G10
132 32 63 0. 0.013 0. 1.04 11;%80 G11
133 36 64 0. 0.0075 0. 1.04 12;%81 G12
134 37 65 0. 0.0033 0. 1.04 13;%82 G13
135 41 66 0. 0.0015 0. 1. 14;%83 G14
136 42 67 0. 0.0015 0. 1. 15;%84 G15
137 52 68 0. 0.0030 0. 1. 16;%85 G16
138 1 27 0.032 0.32 0.41 0 0.];%86
139 zbase=(345^2)/100;
140 for i=1:length(line(:,1))
141     if line(i,6)==0
142         line(i,3)=line(i,3).*zbase/1;
143         line(i,4)=line(i,4).*zbase/(2*pi*60)/1;
144         line(i,5)=line(i,5)./zbase/(2*pi*60);
145     end
146     if line(i,6)~=0
147         if line(i,7)~=0 % Trans connected to a Gen
148             line(i,3)=line(i,3)*Mbase(line(i,7))/100;
149             line(i,4)=line(i,4)*Mbase(line(i,7))/100;
150             if line(i,3)==0; line(i,3)=1e-6; end % modifying R of Trans
151         end
152         if line(i,7)==0 % Trans not connected to a Gen
153             line(i,3)=line(i,3)*10;
154             line(i,4)=line(i,4)*10;
155             if line(i,3)==0; line(i,3)=1e-6; end % modifying R of Trans
156         end
157     end
158 end
159
160 %% PSS
161 PSSMODEL=1;
162 MB=[1 .2 30 1.25 40 12 160];
163 %-----
164 %% AVR
165 % Tr Ka Ta Tb Tc Vt Emin Emax
166 AVR_Data=[...
167 0.01 200 0.015 10 1 1.0 -5 5
168 0.01 200 0.015 10 1 1.0 -5 5
169 0.01 200 0.015 10 1 1.0 -5 5
170 0.01 200 0.015 10 1 1.0 -5 5
171 0.01 200 0.015 10 1 1.0 -5 5
172 0.01 200 0.015 10 1 1.0 -5 5
173 0.01 200 0.015 10 1 1.0 -5 5

```

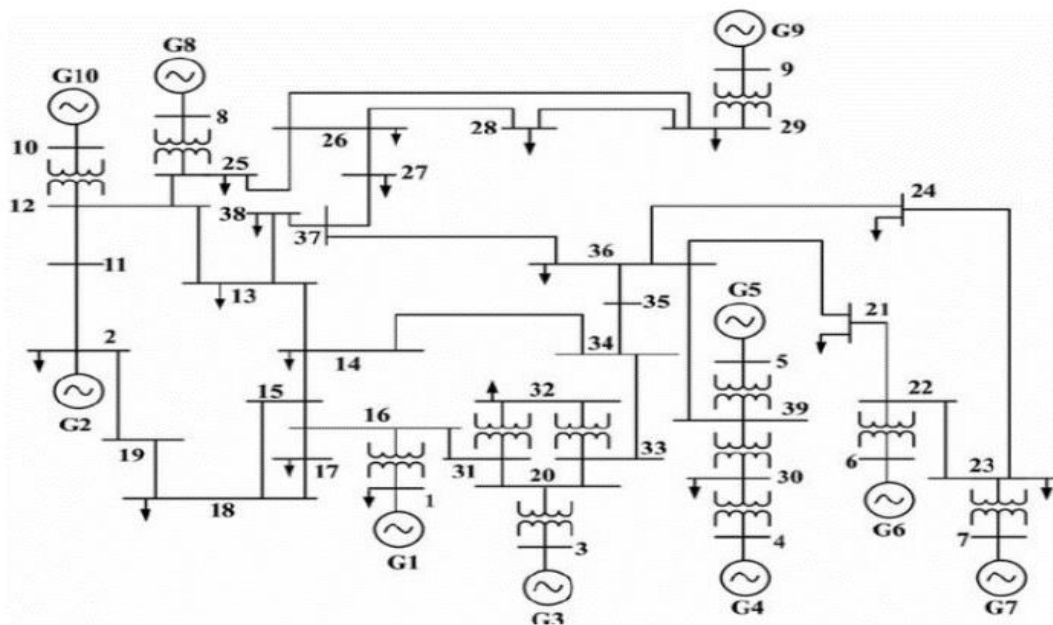
```
174 0.01 200 0.015 10 1 1.0 -5 5
175 0.01 200 0.015 10 1 1.0 -5 5
176 0.01 200 0.015 10 1 1.0 -5 5
177 0.01 200 0.015 10 1 1.0 -5 5
178 0.01 200 0.015 10 1 1.0 -5 5
179 0.01 200 0.015 10 1 1.0 -5 5
180 0.01 200 0.015 10 1 1.0 -5 5
181 0.01 200 0.015 10 1 1.0 -5 5
182 0.01 200 0.015 10 1 1.0 -5 5];
183 %%
184 Bus = [
185 1 345
186 2 345
187 3 345
188 4 345
189 5 345
190 6 345
191 7 345
192 8 345
193 9 345
194 10 345
195 11 345
196 12 230
197 13 345
198 14 345
199 15 345
200 16 345
201 17 345
202 18 345
203 19 345
204 20 345
205 21 345
206 22 345
207 23 345
208 24 345
209 25 345
210 26 345
211 27 345
212 28 345
213 29 345
214 30 345
215 31 345
216 32 345
217 33 345
218 34 345
```

```
219 35 345
220 36 345
221 37 345
222 38 345
223 39 345
224 40 345
225 41 345
226 42 345
227 43 345
228 44 345
229 45 345
230 46 345
231 47 345
232 48 345
233 49 345
234 50 345
235 51 345
236 52 345
237 53 22
238 54 22
239 55 22
240 56 22
241 57 22
242 58 22
243 59 22
244 60 22
245 61 22
246 62 22
247 63 22
248 64 22
249 65 22
250 66 22
251 67 22
252 68 22];
253
254 Ns=120*60/(2); %Nominal speed of synchronous machines
```

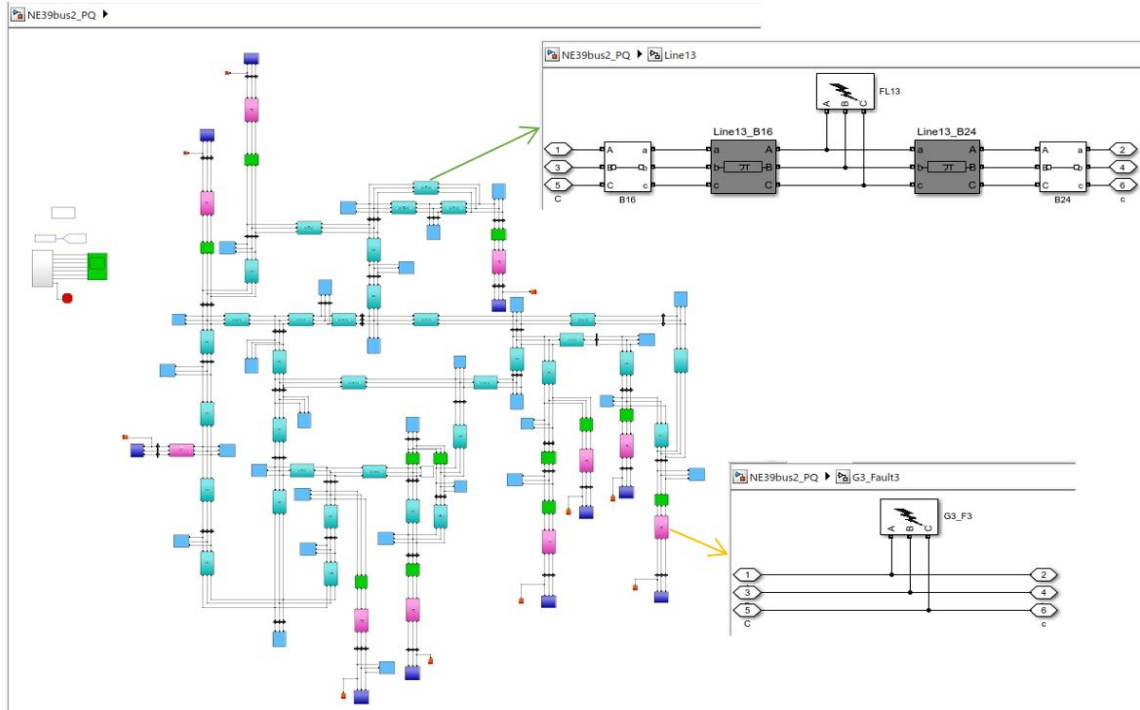
A.2 Modèles Simulink des réseaux IEEE 39 –bus et IEEE 68 -bus

A.2.1 Modèle Simulink du réseau IEEE 39 -bus

La Figure 38 présente le modèle Simulink du réseau IEEE 39 –bus utilisé pour la génération simple et rapide des signaux de stabilité [43]. Elle présente un exemple de configuration de la ligne 13 située entre le bus B16 et le bus B24. Chaque ligne est équipée de deux block Three-phase Breaker et d'un bloc Three-phase Fault, tous disponibles dans la librairie SPS. Elle présente un exemple de configuration de la barre génératrice B3 est équipée d'un bloc Three-phase Fault, situé à l'entrée du transformateur T2. Suivant cette configuration, chaque barre génératrice du réseau est équipée d'un bloc Three-phase fault.



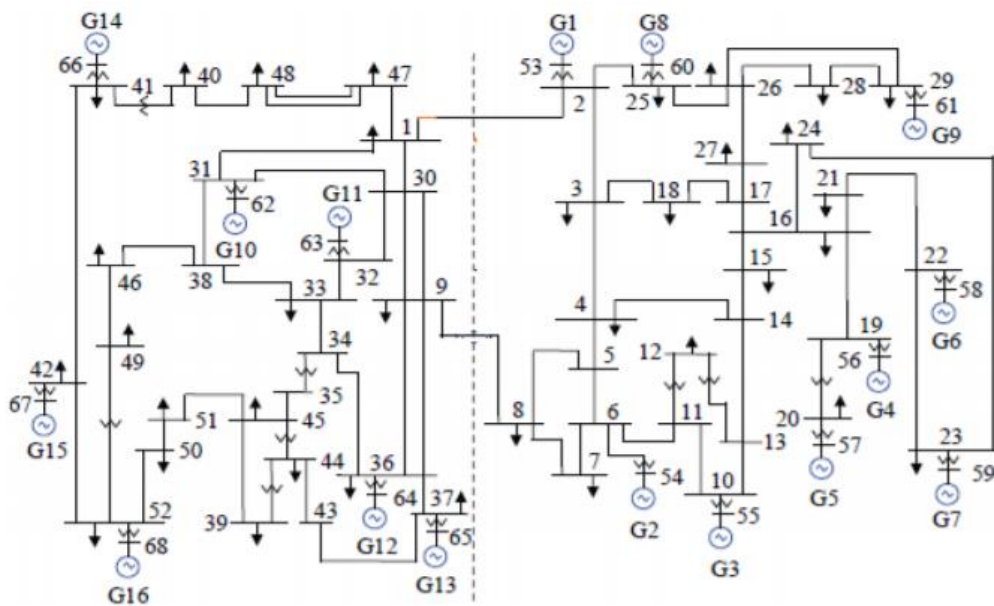
(a) : Diagramme unifilaire du réseau IEEE 39 –bus.



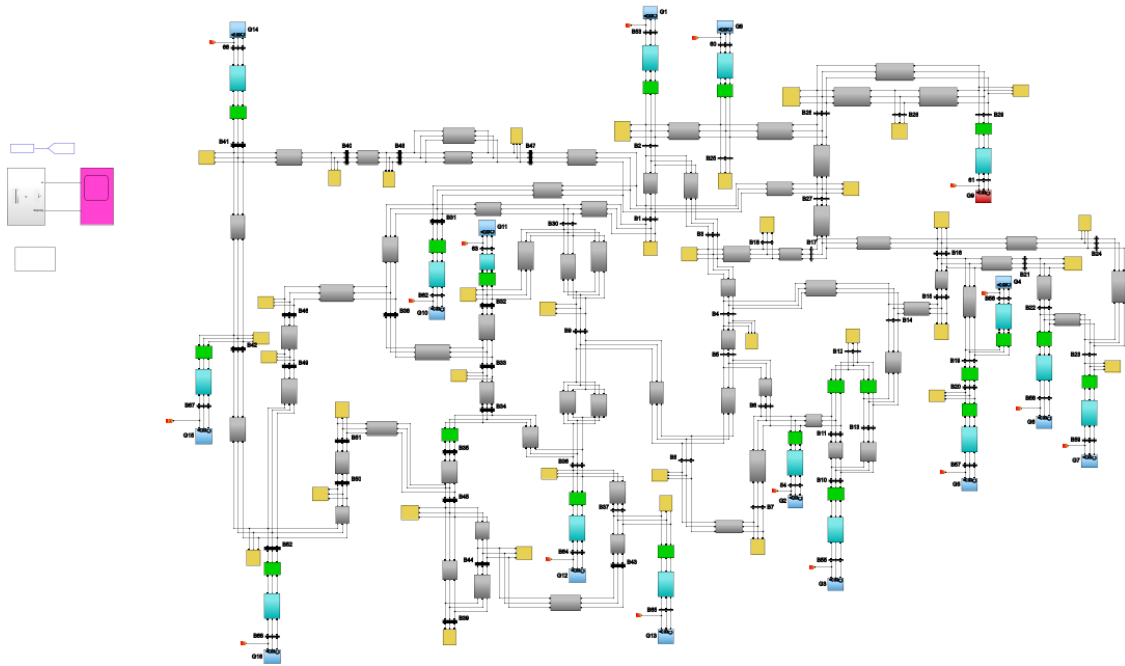
(b) : Modèle Simulink du réseau IEEE 39 –bus [43].

Figure 38 : Diagramme unifilaire (a) et configuration Simulink (b) du réseau IEEE 39 – bus utilisée pour la génération des signaux de stabilité.

A.2.2 Modèle Simulink du réseau IEEE 68 -bus



(a) Diagramme unifilaire du réseau IEEE 68 –bus.



(b) : Modèle Simulink du réseau IEEE 68 –bus.

Figure 39: Diagramme unifilaire (a) et configuration Simulink (b) du réseau IEEE 68 – bus utilisée pour la génération des signaux de stabilité.

A.3 Code Matlab de génération des signaux de stabilité des données

```

1 % =====
2 % An example code for the bisection algorithm.
3 % Network:IEEE 39-bus
4 % Author: Raoult Teukam Dabou
5 % Power Electronics and Industrial Control Laboratory (LEEPCI), Laval University
6 % Date: 02-11-2021.
7 % Ce code permet de calculer par simulation le temps critique d'élimination des défauts
8 % lorsqu'un défaut biphasé terre est simulé à 50% de chaque ligne.
9 % =====
10 %% Défaut Triphasé-Terre
11 ig=1;close all
12 %% Line 1
13 Dncy=1/2;
14 for i=1:31
15     switch i

```

```

16 case 2
17 % setting up CT1,CT2&e--Ph AB-G
18     stab=0; e=Dncy; nmach=10;
19     mdl='NE39bus2_PQ'; load_system(mdl);
20     [~,txt,~] = xlsread('CB names.xlsx');
21 while stab==0
22     CT2=2*e-1; CT1=1;
23     while (CT2-CT1>e)
24         CT=(CT1+CT2)/2; x=1; y=CT;
25         % Fixer le switching time du breaker premier breaker
26         pathname = strcat(mdl,'/',char(txt{i,18}),',' char(txt{i,13}));
27         set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes',[x+y/60 10]);
28         % Fixer le switching time du breaker deuxième breaker
29         pathname = strcat(mdl,'/',char(txt{i,18}),',' char(txt{i,14}));
30         set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes',[x+y/60 10]);
31         % Fixer le temps de simulation du défaut.
32         pathname = strcat(mdl,'/',char(txt{i,18}),',' char(txt{i,17}));
33         set_param(pathname,'External','off','SwitchTimes',[x x+y/60]);
34         % Simulation du défaut 2PhAB-G
35         set_param(pathname,'FaultA',1); set_param(pathname,'FaultB',1);
36         set_param(pathname,'FaultC',1); set_param(pathname,'GroundFault',1);
37         LF = power_loadflow(mdl); sim(mdl);
38         % Fixer à une valeur supérieure à la simulation le temps de commutation des disjoncteurs
39         pathname = strcat(mdl,'/',char(txt{i,18}),',' char(txt{i,13}));
40         set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes','11');
41         % Fixer à une valeur supérieure à la simulation le temps de commutation des disjoncteurs
42         pathname = strcat(mdl,'/',char(txt{i,18}),',' char(txt{i,14}));
43         set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes','11');
44         % Fixer à une valeur supérieure à la simulation le temps de commutation des défauts
45         pathname = strcat(mdl,'/',char(txt{i,18}),',' char(txt{i,17}));
46         set_param(pathname,'External','off','SwitchTimes','11');
47         pathname = strcat(mdl,'/',char(txt{i,18}),',' char(txt{i,17}));
48         set_param(pathname,'FaultA',0); set_param(pathname,'FaultB',0);
49         set_param(pathname,'FaultC',0); set_param(pathname,'GroundFault',0);
50         stab=max(stability); g=2.5;
51     if stab==0
52         CT1=CT; s1_L11_=CT1;
53         s11_=mach; nmach=10;
54         A=mach.time; w_stable= mach.signals(2).values;
55         Vd_stable= mach.signals(5).values; Vq_stable= mach.signals(4).values;
56         Iq_stable= mach.signals(6).values; Id_stable= mach.signals(7).values;
57     if g==e
58         g1=1;
59         Vitesse_stable(1:601,nmach*(g1-1)+1:nmach*g1)=s11_.signals(2).values(1:601,1:10);
60         status_stable(1:601,(g1-1)+1:g1)=s11_.signals(3).values(1:601,1);

```

```

61     CCT_stable(1,(g1-1)+1:g1)=CT1;
62     Time_stable(1:601,(g1-1)+1:g1)=s11_.time(1:601,1); gg=2;
63     end
64     if g~=e
65         Vitesse_stable(1:601,nmach*(gg-1)+1:nmach*gg)=s11_.signals(2).values(1:601,1:10);
66         status_stable(1:601,(gg-1)+1:gg)=s11_.signals(3).values(1:601,1); CCT_stable(1,(gg-1)+1:gg)=CT1;
67         Time_stable(1:601,(gg-1)+1:gg)=s11_.time(1:601,1); gg=gg+1;
68     end
69     figure(i);
70     subplot(211);plot(mach.time,mach.signals(1).values);grid;subplot(212);
71     plot(mach.time,mach.signals(2).values);grid title(['CCT_3Ph=',num2str(CT)]);
72     else
73         CT2=CT;
74     end
75 end
76     CCT=CT2;
77     e=e+Dncy;
78 end
79     s1_L1=CT2;s1=mach;
80     A1=mach.time; Id_unstable= mach.signals(7).values;
81     w_unstable= mach.signals(2).values; Vd_unstable= mach.signals(5).values;
82     Vq_unstable= mach.signals(4).values; Iq_unstable= mach.signals(6).values;
83     % figure(i+34); subplot(211);plot(mach.time,mach.signals(1).values);grid;subplot(212);
84     % plot(mach.time,mach.signals(2).values);grid title(['CCT=',num2str(CT)]);
85     f=1;
86     Vitesse_unstable(1:601,nmach*(f-1)+1:nmach*f)=s1.signals(2).values(1:601,1:10);
87     status_unstable(1:601,(f-1)+1:f)=s1.signals(3).values(1:601,1);
88     Time_unstable(1:601,(f-1)+1:f)=s1.time(1:601,1);
89     fname = strcat('Data','_',char(txt{i,12}));
90     unstable=s1;stable=s11_; CCT_unstable=CT2;
91     save(fname,'unstable','stable','CCT_stable','CCT_unstable');
92     save(['Line1.mat','Vitesse_unstable','Vitesse_stable','status_stable','status_unstable','Time_unstable',
93     'Time_stable','CCT_stable','CCT_unstable']);clear Vitesse_unstable; clear Vitesse_stable; clear status_stable;
94     case 3
95     %% Line 2
96     Dncy=1/2; stab=0; e=Dncy; nmach=10;
97     mdl='NE39bus2_PQ'; load_system(mdl);
98     [~,txt,~] = xlsread('CB names.xlsx');
99     while stab==0
100     CT2=2*e-1; CT1=1;
101     while (CT2-CT1>e)
102         CT=(CT1+CT2)/2; x=1; y=CT;
103         % Fixer le switching time du breaker premier breaker
104         pathname = strcat(mdl,'/',char(txt{i,18}),'/',char(txt{i,13}));

```

```

105 set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes',[x+y/60 10]);
106 % Fixer le switching time du breaker deuxième breaker
107 pathname = strcat(mdl,/,char(txt{i,18}),/,char(txt{i,14}));
108 set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes',[x+y/60 10]);
109 % Fixer le temps de simulation du défaut.
110 pathname = strcat(mdl,/,char(txt{i,18}),/,char(txt{i,17}));
111 set_param(pathname,'External','off','SwitchTimes',[x x+y/60]);
112 % Simulation du défaut 2PhAB-G
113 set_param(pathname,'FaultA',1); set_param(pathname,'FaultB',1);
114 set_param(pathname,'FaultC',1); set_param(pathname,'GroundFault',1);
115 LF = power_loadflow(mdl); sim(mdl);
116 % Fixer à une valeur supérieure à la simulation le temps de commutation des disjoncteurs
117 pathname = strcat(mdl,/,char(txt{i,18}),/,char(txt{i,13}));
118 set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes','11');
119 % Fixer à une valeur supérieure à la simulation le temps de commutation des disjoncteurs
120 pathname = strcat(mdl,/,char(txt{i,18}),/,char(txt{i,14}));
121 set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes','11');
122 % Fixer à une valeur supérieure à la simulation le temps de commutation des défauts
123 pathname = strcat(mdl,/,char(txt{i,18}),/,char(txt{i,17}));
124 set_param(pathname,'External','off','SwitchTimes','11');
125 pathname = strcat(mdl,/,char(txt{i,18}),/,char(txt{i,17}));
126 set_param(pathname,'FaultA',0); set_param(pathname,'FaultB',0);
127 set_param(pathname,'FaultC',0); set_param(pathname,'GroundFault',0);
128 stab=max(stability);g=2.5;
129 if stab==0
130     CT1=CT; s2_L22=CT1;
131     s22_=mach;
132     A=mach.time; w_stable= mach.signals(2).values;
133     Vd_stable= mach.signals(5).values; Vq_stable= mach.signals(4).values;
134     Iq_stable= mach.signals(6).values; Id_stable= mach.signals(7).values;
135 %     figure(i);
136 %     subplot(21 1);plot(mach.time,mach.signals(1).values);grid;subplot(21 2);
137 %     plot(mach.time,mach.signals(2).values);grid title(['CCT=',num2str(CT)]);
138 % stable
139 if g==e
140     g1=1;
141     Vitesse_stable(1:601,nmach*(g1-1)+1:nmach*g1)=s22_.signals(2).values(1:601,1:10);
142     status_stable(1:601,(g1-1)+1:g1)=s22_.signals(3).values(1:601,1);
143     CCT_stable(1,(g1-1)+1:g1)=CT1;
144     Time_stable(1:601,(g1-1)+1:g1)=s22_.time(1:601,1);gg=2;
145 end
146 if g~=e
147     Vitesse_stable(1:601,nmach*(gg-1)+1:nmach*gg)=s22_.signals(2).values(1:601,1:10);
148     status_stable(1:601,(gg-1)+1:gg)=s22_.signals(3).values(1:601,1);
149     CCT_stable(1,(gg-1)+1:gg)=CT1;

```

```

150     Time_stable(1:601,(gg-1)+1:gg)=s22_.time(1:601,1); gg=gg+1;
151     end
152     else
153         CT2=CT;
154     end
155 end
156     CCT=CT2;
157     e=e+Dncy;
158 end
159     s2_L2=CT2; s2=mach;
160     A1=mach.time; Id_unstable= mach.signals(7).values;
161     w_unstable= mach.signals(2).values; Vd_unstable= mach.signals(5).values;
162     Vq_unstable= mach.signals(4).values; Iq_unstable= mach.signals(6).values;
163     figure(i+34);
164     subplot(211);plot(mach.time,mach.signals(1).values);grid;subplot(212);
165     plot(mach.time,mach.signals(2).values);grid title(['CCT=',num2str(CT)]);
166     % unstable
167     f=1;
168     Vitesse_unstable(1:601,nmach*(f-1)+1:nmach*f)=s2.signals(2).values(1:601,1:10);
169     status_unstable(1:601,(f-1)+1:f)=s2.signals(3).values(1:601,1);
170     Time_unstable(1:601,(f-1)+1:f)=s2.time(1:601,1); CCT_unstable=CT2;
171     fname = strcat('Data','_',char(txt{i,12}));
172     unstable=s2;stable=s22_;
173     save(fname,'unstable','stable','CCT_stable','CCT_unstable');
174     save(['.Line2.mat','Vitesse_unstable','Vitesse_stable','status_stable','status_unstable','Time_unstable',
175 'Time_stable','CCT_stable','CCT_unstable']); %%L3
176     clear Vitesse_unstable; clear Vitesse_stable; clear status_stable; clear status_unstable;
177     clear Time_unstable, clear Time_stable; clear CCT_stable; clear CCT_unstable;
178 case 4
179 %% Line 3
180     Dncy=1/2; stab=0; e=Dncy; nmach=10;
181     mdl='NE39bus2_PQ'; load_system(mdl);
182     [~,txt,~] = xlsread('CB names.xlsx');
183     while stab==0
184         CT2=2*e-1; CT1=1;
185         while (CT2-CT1>e)
186             CT=(CT1+CT2)/2; x=1; y=CT;
187             % Fixer le switching time du breaker premier breaker
188             pathname = strcat(mdl,'/',char(txt{i,18}),'/',char(txt{i,13}));
189             set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes',[x+y/60 10]);
190             % Fixer le switching time du breaker deuxième breaker
191             pathname = strcat(mdl,'/',char(txt{i,18}),'/',char(txt{i,14}));
192             set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes',[x+y/60 10]);
193             % Fixer le temps de simulation du défaut.
194             pathname = strcat(mdl,'/',char(txt{i,18}),'/',char(txt{i,17}));

```

```

194 set_param(pathname,'External','off','SwitchTimes','[x x+y/60]');
195 % Simulation du défaut 2PhAB-G
196 set_param(pathname,'FaultA',1); set_param(pathname,'FaultB',1);
197 set_param(pathname,'FaultC',1); set_param(pathname,'GroundFault',1);
198 LF = power_loadflow mdl; sim mdl;
199 % Fixer à une valeur supérieure à la simulation le temps de commutation des disjoncteurs
200 pathname = strcat mdl,'/',char(txt{i,18}),',' char(txt{i,13});
201 set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes','11');
202 % Fixer à une valeur supérieure à la simulation le temps de commutation des disjoncteurs
203 pathname = strcat mdl,'/',char(txt{i,18}),',' char(txt{i,14});
204 set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes','11');
205 % Fixer à une valeur supérieure à la simulation le temps de commutation des défauts
206 pathname = strcat mdl,'/',char(txt{i,18}),',' char(txt{i,17});
207 set_param(pathname,'External','off','SwitchTimes','11');
208 pathname = strcat mdl,'/',char(txt{i,18}),',' char(txt{i,17});
209 set_param(pathname,'FaultA',0); set_param(pathname,'FaultB',0);
210 set_param(pathname,'FaultC',0); set_param(pathname,'GroundFault',0);
211 stab=max(stability);g=2.5;
212 if stab==0
213     CT1=CT; s3_L33=CT1;
214     s33=mach;
215     A=mach.time; w_stable= mach.signals(2).values;
216     Vd_stable= mach.signals(5).values; Vq_stable= mach.signals(4).values;
217     Iq_stable= mach.signals(6).values; Id_stable= mach.signals(7).values;
218 % figure(i);
219 % subplot(211);plot(mach.time,mach.signals(1).values);grid;subplot(212);
220 % plot(mach.time,mach.signals(2).values);grid title(['CCT=',num2str(CT)]);
221 % stable
222 if g==e
223     g1=1;
224     Vitesse_stable(1:601,nmach*(g1-1)+1:nmach*g1)=s33.signals(2).values(1:601,1:10);
225     status_stable(1:601,(g1-1)+1:g1)=s33.signals(3).values(1:601,1);
226     CCT_stable(1,(g1-1)+1:g1)=CT1;
227     Time_stable(1:601,(g1-1)+1:g1)=s33.time(1:601,1);gg=2;
228 end
229 if g~=e
230     Vitesse_stable(1:601,nmach*(gg-1)+1:nmach*gg)=s33.signals(2).values(1:601,1:10);
231     status_stable(1:601,(gg-1)+1:gg)=s33.signals(3).values(1:601,1);
232     CCT_stable(1,(gg-1)+1:gg)=CT1;
233     Time_stable(1:601,(gg-1)+1:gg)=s33.time(1:601,1); gg=gg+1;
234 end
235 else
236     CT2=CT;
237 end
238 end

```

```

239     CCT=CT2; e=e+Dncy;
240 end
241     s3_L3=CT2; s3=mach;
242     A1=mach.time; Id_unstable= mach.signals(7).values;
243     w_unstable= mach.signals(2).values; Vd_unstable= mach.signals(5).values;
244     Vq_unstable= mach.signals(4).values; Iq_unstable= mach.signals(6).values;
245 %     figure(i+34);
246 %     subplot(211);plot(mach.time,mach.signals(1).values);grid;subplot(212);
247 %     plot(mach.time,mach.signals(2).values);grid title(['CCT=',num2str(CT)]);
248 % unstable
249     f=1;
250     Vitesse_unstable(1:601,nmach*(f-1)+1:nmach*f)=s3.signals(2).values(1:601,1:10);
251     status_unstable(1:601,(f-1)+1:f)=s3.signals(3).values(1:601,1);
252     Time_unstable(1:601,(f-1)+1:f)=s3.time(1:601,1); CCT_unstable=CT2;
253     fname = strcat('Data','_',char(txt{i,12}));
254     unstable=s3;stable=s33;
255     save(fname,'unstable','stable','CCT_stable','CCT_unstable');
256     save('.\Line3.mat','Vitesse_unstable','Vitesse_stable','status_stable','status_unstable',
257 'Time_unstable','Time_stable','CCT_stable','CCT_unstable');    %%L4
    clear Vitesse_unstable; clear Vitesse_stable; clear status_stable; clear status_unstable;
258 clear Time_unstable, clear Time_stable; clear CCT_stable; clear CCT_unstable;
259
260     case 5
261 %% Line 4
262     Dncy=1/2; stab=0; e=Dncy; nmach=10;
263     mdl='NE39bus2_PQ'; load_system(mdl);
264     [~,txt,~] = xlsread('CB names.xlsx');
265 while stab==0
266 CT2=2*e-1; CT1=1;
267 while (CT2-CT1>e)
268     CT=(CT1+CT2)/2; x=1; y=CT;
269     % Fixer le switching time du breaker premier breaker
270     pathname = strcat(mdl,'/',char(txt{i,18}),'/',char(txt{i,13}));
271     set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes',[x+y/60 10]);
272     % Fixer le switching time du breaker deuxième breaker
273     pathname = strcat(mdl,'/',char(txt{i,18}),'/',char(txt{i,14}));
274     set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes',[x+y/60 10]);
275     % Fixer le temps de simulation du défaut.
276     pathname = strcat(mdl,'/',char(txt{i,18}),'/',char(txt{i,17}));
277     set_param(pathname,'External','off','SwitchTimes',[x x+y/60]);
278     % Simulation du défaut 2PhAB-G
279     set_param(pathname,'FaultA',1); set_param(pathname,'FaultB',1);
280     set_param(pathname,'FaultC',1); set_param(pathname,'GroundFault',1);
281     LF = power_loadflow(mdl); sim(mdl);
282     % Fixer à une valeur supérieure à la simulation le temps de commutation des disjoncteurs

```



```

283     pathname = strcat(mdl, '/', char(txt{i,18}), '/', char(txt{i,13}));
284     set_param(pathname, 'SnubberResistance', 'inf', 'External', 'off', 'SwitchTimes', '11');
285     % Fixer à une valeur supérieure à la simulation le temps de commutation des disjoncteurs
286     pathname = strcat(mdl, '/', char(txt{i,18}), '/', char(txt{i,14}));
287     set_param(pathname, 'SnubberResistance', 'inf', 'External', 'off', 'SwitchTimes', '11');
288     % Fixer à une valeur supérieure à la simulation le temps de commutation des défauts
289     pathname = strcat(mdl, '/', char(txt{i,18}), '/', char(txt{i,17}));
290     set_param(pathname, 'External', 'off', 'SwitchTimes', '11');
291     pathname = strcat(mdl, '/', char(txt{i,18}), '/', char(txt{i,17}));
292     set_param(pathname, 'FaultA', 0); set_param(pathname, 'FaultB', 0);
293     set_param(pathname, 'FaultC', 0); set_param(pathname, 'GroundFault', 0);
294     stab=max(stability);g=2.5;
295     if stab==0
296         CT1=CT; s4_L44=CT1;
297         s44=mach;
298         A=mach.time; w_stable= mach.signals(2).values;
299         Vd_stable= mach.signals(5).values; Vq_stable= mach.signals(4).values;
300         Iq_stable= mach.signals(6).values; Id_stable= mach.signals(7).values;
301     %     figure(i);
302     %     subplot(211);plot(mach.time,mach.signals(1).values);grid;subplot(212);
303     %     plot(mach.time,mach.signals(2).values);grid title(['CCT=',num2str(CT)]);
304     % stable
305     if g==e
306         g1=1;
307         Vitesse_stable(1:601,nmach*(g1-1)+1:nmach*g1)=s44.signals(2).values(1:601,1:10);
308         status_stable(1:601,(g1-1)+1:g1)=s44.signals(3).values(1:601,1);
309         CCT_stable(1,(g1-1)+1:g1)=CT1;
310         Time_stable(1:601,(g1-1)+1:g1)=s44.time(1:601,1);gg=2;
311     end
312     if g~=e
313         Vitesse_stable(1:601,nmach*(gg-1)+1:nmach*gg)=s44.signals(2).values(1:601,1:10);
314         status_stable(1:601,(gg-1)+1:gg)=s44.signals(3).values(1:601,1);
315         CCT_stable(1,(gg-1)+1:gg)=CT1;
316         Time_stable(1:601,(gg-1)+1:gg)=s44.time(1:601,1); gg=gg+1;
317     end
318     else
319         CT2=CT;
320     end
321 end
322     CCT=CT2; e=e+Dncy;
323 end
324     s4_L4=CT2; s4=mach;
325     A1=mach.time; Id_unstable= mach.signals(7).values;
326     w_unstable= mach.signals(2).values; Vd_unstable= mach.signals(5).values;
327     Vq_unstable= mach.signals(4).values; Iq_unstable= mach.signals(6).values;

```

```

328 % figure(i+34);
329 % subplot(211);plot(mach.time,mach.signals(1).values);grid;subplot(212);
330 % plot(mach.time,mach.signals(2).values);grid title(['CCT=',num2str(CT)]
331 % unstable
332 f=1;
333 Vitesse_unstable(1:601,nmach*(f-1)+1:nmach*f)=s4.signals(2).values(1:601,1:10);
334 status_unstable(1:601,(f-1)+1:f)=s4.signals(3).values(1:601,1);
335 Time_unstable(1:601,(f-1)+1:f)=s4.time(1:601,1); CCT_unstable=CT2;
336 fname = strcat('Data','_',char(txt{i,12}));
337 unstable=s4;stable=s44;
338 save(fname,'unstable','stable','CCT_stable','CCT_unstable');
339 save('\Line4.mat','Vitesse_unstable','Vitesse_stable','status_stable','status_unstable',
340 'Time_unstable','Time_stable','CCT_stable','CCT_unstable'); %%L5
clear Vitesse_unstable; clear Vitesse_stable; clear status_stable; clear status_unstable;
341 clear Time_unstable, clear Time_stable; clear CCT_stable; clear CCT_unstable;
342
343 case 6
344 %% Line 5
345 Dncy=1/2; stab=0; e=Dncy; nmach=10;
346 mdl='NE39bus2_PQ'; load_system(mdl);
347 [~,txt,~] = xlsread('CB names.xlsx');
348 while stab==0
349 CT2=2*e-1; CT1=1;
350 while (CT2-CT1>e)
351 CT=(CT1+CT2)/2; x=1; y=CT;
352 % Fixer le switching time du breaker premier breaker
353 pathname = strcat(mdl,'/',char(txt{i,18}),'/',char(txt{i,13}));
354 set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes',[x+y/60 10]);
355 % Fixer le switching time du breaker deuxième breaker
356 pathname = strcat(mdl,'/',char(txt{i,18}),'/',char(txt{i,14}));
357 set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes',[x+y/60 10]);
358 % Fixer le temps de simulation du défaut.
359 pathname = strcat(mdl,'/',char(txt{i,18}),'/',char(txt{i,17}));
360 set_param(pathname,'External','off','SwitchTimes',[x x+y/60]);
361 % Simulation du défaut 2PhAB-G
362 set_param(pathname,'FaultA',1); set_param(pathname,'FaultB',1);
363 set_param(pathname,'FaultC',1); set_param(pathname,'GroundFault',1);
364 LF = power_loadflow(mdl); sim(mdl);
365 % Fixer à une valeur supérieure à la simulation le temps de commutation des disjoncteurs
366 pathname = strcat(mdl,'/',char(txt{i,18}),'/',char(txt{i,13}));
367 set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes','11');
368 % Fixer à une valeur supérieure à la simulation le temps de commutation des disjoncteurs
369 pathname = strcat(mdl,'/',char(txt{i,18}),'/',char(txt{i,14}));
370 set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes','11');
371 % Fixer à une valeur supérieure à la simulation le temps de commutation des défauts

```

```

372     pathname = strcat(mdl, '/', char(txt{i,18}), '/', char(txt{i,17}));
373     set_param(pathname, 'External', 'off', 'SwitchTimes', '11');
374     pathname = strcat(mdl, '/', char(txt{i,18}), '/', char(txt{i,17}));
375     set_param(pathname, 'FaultA', 0); set_param(pathname, 'FaultB', 0);
376     set_param(pathname, 'FaultC', 0); set_param(pathname, 'GroundFault', 0);
377     stab=max(stability);g=2.5;
378     if stab==0
379         CT1=CT; s5_L55=CT1;
380         s55=mach;
381         A=mach.time; w_stable= mach.signals(2).values;
382         Vd_stable= mach.signals(5).values; Vq_stable= mach.signals(4).values;
383         Iq_stable= mach.signals(6).values; Id_stable= mach.signals(7).values;
384     % figure(i);
385     %     subplot(211);plot(mach.time,mach.signals(1).values);grid;subplot(212);
386     %     plot(mach.time,mach.signals(2).values);grid title(['CCT=',num2str(CT)]);
387     % stable
388     if g==e
389         g1=1;
390         Vitesse_stable(1:601,nmach*(g1-1)+1:nmach*g1)=s55.signals(2).values(1:601,1:10);
391         status_stable(1:601,(g1-1)+1:g1)=s55.signals(3).values(1:601,1);
392         CCT_stable(1,(g1-1)+1:g1)=CT1;
393         Time_stable(1:601,(g1-1)+1:g1)=s55.time(1:601,1);gg=2;
394     end
395     if g~=e
396         Vitesse_stable(1:601,nmach*(gg-1)+1:nmach*gg)=s55.signals(2).values(1:601,1:10);
397         status_stable(1:601,(gg-1)+1:gg)=s55.signals(3).values(1:601,1);
398         CCT_stable(1,(gg-1)+1:gg)=CT1;
399         Time_stable(1:601,(gg-1)+1:gg)=s55.time(1:601,1); gg=gg+1;
400     end
401
402     else
403         CT2=CT;
404     end
405 end
406     CCT=CT2; e=e+Dncy;
407 end
408     s5_L5=CT2;s5=mach;
409     A1=mach.time; Id_unstable= mach.signals(7).values;
410     w_unstable= mach.signals(2).values; Vd_unstable= mach.signals(5).values;
411     Vq_unstable= mach.signals(4).values; Iq_unstable= mach.signals(6).values;
412     % figure(i+34);
413     %     subplot(211);plot(mach.time,mach.signals(1).values);grid;subplot(212);
414     %     plot(mach.time,mach.signals(2).values);grid title(['CCT=',num2str(CT)]
415
416     % unstable

```

```

417     f=1;
418     Vitesse_unstable(1:601,nmach*(f-1)+1:nmach*f)=s5.signals(2).values(1:601,1:10);
419     status_unstable(1:601,(f-1)+1:f)=s5.signals(3).values(1:601,1);
420     Time_unstable(1:601,(f-1)+1:f)=s5.time(1:601,1); CCT_unstable=CT2;
421     fname = strcat('Data','_',char(txt{i,12}));
422     unstable=s5;stable=s55;
423     save(fname,'unstable','stable','CCT_stable','CCT_unstable');
424     save('\Line5.mat','Vitesse_unstable','Vitesse_stable','status_stable','status_unstable','Time_unstable',
425 'Time_stable','CCT_stable','CCT_unstable');    %%L6
426     clear Vitesse_unstable; clear Vitesse_stable; clear status_stable; clear status_unstable;
427     clear Time_unstable, clear Time_stable; clear CCT_stable; clear CCT_unstable;
428
429     case 7
430     %% Line 6
431     Dncy=1/2; stab=0; e=Dncy; nmach=10;
432     mdl='NE39bus2_PQ'; load_system(mdl);
433     [~,txt,~] = xlsread('CB names.xlsx');
434     while stab==0
435     CT2=2*e-1; CT1=1;
436     while (CT2-CT1>e)
437         CT=(CT1+CT2)/2; x=1; y=CT;
438         % Fixer le switching time du breaker premier breaker
439         pathname = strcat(mdl,'/',char(txt{i,18}),'/',char(txt{i,13}));
440         set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes',[x+y/60 10]);
441         % Fixer le switching time du breaker deuxième breaker
442         pathname = strcat(mdl,'/',char(txt{i,18}),'/',char(txt{i,14}));
443         set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes',[x+y/60 10]);
444         % Fixer le temps de simulation du défaut.
445         pathname = strcat(mdl,'/',char(txt{i,18}),'/',char(txt{i,17}));
446         set_param(pathname,'External','off','SwitchTimes',[x x+y/60]);
447         % Simulation du défaut 2PhAB-G
448         set_param(pathname,'FaultA',1); set_param(pathname,'FaultB',1);
449         set_param(pathname,'FaultC',1); set_param(pathname,'GroundFault',1);
450         LF = power_loadflow(mdl); sim(mdl);
451         % Fixer à une valeur supérieure à la simulation le temps de commutation des disjoncteurs
452         pathname = strcat(mdl,'/',char(txt{i,18}),'/',char(txt{i,13}));
453         set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes','11');
454         % Fixer à une valeur supérieure à la simulation le temps de commutation des disjoncteurs
455         pathname = strcat(mdl,'/',char(txt{i,18}),'/',char(txt{i,14}));
456         set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes','11');
457         % Fixer à une valeur supérieure à la simulation le temps de commutation des défauts
458         pathname = strcat(mdl,'/',char(txt{i,18}),'/',char(txt{i,17}));
459         set_param(pathname,'External','off','SwitchTimes','11');
460         pathname = strcat(mdl,'/',char(txt{i,18}),'/',char(txt{i,17}));
461         set_param(pathname,'FaultA',0); set_param(pathname,'FaultB',0);

```

```

461     set_param(pathname,'FaultC',0); set_param(pathname,'GroundFault',0);
462     stab=max(stability);g=2.5;
463     if stab==0
464         CT1=CT; s6_L66=CT1;
465         s66=mach;
466         A=mach.time; w_stable= mach.signals(2).values;
467         Vd_stable= mach.signals(5).values; Vq_stable= mach.signals(4).values;
468         Iq_stable= mach.signals(6).values; Id_stable= mach.signals(7).values;
469     %     figure(i);
470     %     subplot(211);plot(mach.time,mach.signals(1).values);grid;subplot(212);
471     %     plot(mach.time,mach.signals(2).values);grid title(['CCT=',num2str(CT)]);
472     % stable
473     if g==e
474         g1=1;
475         Vitesse_stable(1:601,nmach*(g1-1)+1:nmach*g1)=s66.signals(2).values(1:601,1:10);
476         status_stable(1:601,(g1-1)+1:g1)=s66.signals(3).values(1:601,1);
477         CCT_stable(1,(g1-1)+1:g1)=CT1;
478         Time_stable(1:601,(g1-1)+1:g1)=s66.time(1:601,1);gg=2;
479     end
480     if g~=e
481         Vitesse_stable(1:601,nmach*(gg-1)+1:nmach*gg)=s66.signals(2).values(1:601,1:10);
482         status_stable(1:601,(gg-1)+1:gg)=s66.signals(3).values(1:601,1);
483         CCT_stable(1,(gg-1)+1:gg)=CT1;
484         Time_stable(1:601,(gg-1)+1:gg)=s66.time(1:601,1); gg=gg+1;
485     end
486     else
487         CT2=CT;
488
489     end
490 end
491     e=e+Dncy; CCT=CT2;
492 end
493     s6_L6=CT2; s6=mach;
494     A1=mach.time; Id_unstable= mach.signals(7).values;
495     w_unstable= mach.signals(2).values; Vd_unstable= mach.signals(5).values;
496     Vq_unstable= mach.signals(4).values; Iq_unstable= mach.signals(6).values;
497     %     figure(i+34);
498     %     subplot(211);plot(mach.time,mach.signals(1).values);grid;subplot(212);
499     %     plot(mach.time,mach.signals(2).values);grid title(['CCT=',num2str(CT)]
500     % unstable
501     f=1;
502     Vitesse_unstable(1:601,nmach*(f-1)+1:nmach*f)=s6.signals(2).values(1:601,1:10);
503     status_unstable(1:601,(f-1)+1:f)=s6.signals(3).values(1:601,1);
504     Time_unstable(1:601,(f-1)+1:f)=s6.time(1:601,1); CCT_unstable=CT2;
505     fname = strcat('Data','_',char(txt{i,12}));

```

```

506     unstable=s6;stable=s66;
507     save(fname,'unstable','stable','CCT_stable','CCT_unstable');
508     save('.\Line6.mat','Vitesse_unstable','Vitesse_stable','status_stable','status_unstable',
509 'Time_unstable','Time_stable','CCT_stable','CCT_unstable');    %%L7
    clear Vitesse_unstable; clear Vitesse_stable; clear status_stable; clear status_unstable;
510 clear Time_unstable, clear Time_stable; clear CCT_stable; clear CCT_unstable;
511
512     case 8
513 %% Line 7
514     Dncy=1/2; stab=0; e=Dncy; nmach=10;
515     mdl='NE39bus2_PQ'; load_system(mdl);
516     [~,txt,~] = xlsread('CB names.xlsx');
517 while stab==0
518 CT2=2*e-1; CT1=1;
519 while (CT2-CT1>e)
520     CT=(CT1+CT2)/2; x=1; y=CT;
521     % Fixer le switching time du breaker premier breaker
522     pathname = strcat(mdl,'\char{txt{i,18}}','\char{txt{i,13}});
523     set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes',[x+y/60 10]);
524     % Fixer le switching time du breaker deuxième breaker
525     pathname = strcat(mdl,'\char{txt{i,18}}','\char{txt{i,14}});
526     set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes',[x+y/60 10]);
527     % Fixer le temps de simulation du défaut.
528     pathname = strcat(mdl,'\char{txt{i,18}}','\char{txt{i,17}});
529     set_param(pathname,'External','off','SwitchTimes',[x x+y/60]);
530     % Simulation du défaut 2PhAB-G
531     set_param(pathname,'FaultA',1); set_param(pathname,'FaultB',1);
532     set_param(pathname,'FaultC',1); set_param(pathname,'GroundFault',1);
533     LF = power_loadflow(mdl); sim(mdl);
534     % Fixer à une valeur supérieure à la simulation le temps de commutation des disjoncteurs
535     pathname = strcat(mdl,'\char{txt{i,18}}','\char{txt{i,13}});
536     set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes','11');
537     % Fixer à une valeur supérieure à la simulation le temps de commutation des disjoncteurs
538     pathname = strcat(mdl,'\char{txt{i,18}}','\char{txt{i,14}});
539     set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes','11');
540     % Fixer à une valeur supérieure à la simulation le temps de commutation des défauts
541     pathname = strcat(mdl,'\char{txt{i,18}}','\char{txt{i,17}});
542     set_param(pathname,'External','off','SwitchTimes','11');
543     pathname = strcat(mdl,'\char{txt{i,18}}','\char{txt{i,17}});
544     set_param(pathname,'FaultA',0); set_param(pathname,'FaultB',0);
545     set_param(pathname,'FaultC',0); set_param(pathname,'GroundFault',0);
546     stab=max(stability);g=2.5;
547 if stab==0
548     CT1=CT; s7_L77=CT1;
549     s77=mach;

```

```

550     A=mach.time; w_stable= mach.signals(2).values;
551     Vd_stable= mach.signals(5).values; Vq_stable= mach.signals(4).values;
552     Iq_stable= mach.signals(6).values; Id_stable= mach.signals(7).values;
553 %   figure(i);
554 %   subplot(211);plot(mach.time,mach.signals(1).values);grid;subplot(212);
555 %   plot(mach.time,mach.signals(2).values);grid title(['CCT=',num2str(CT)]);
556 % stable
557     if g==e
558         g1=1;
559         Vitesse_stable(1:601,nmach*(g1-1)+1:nmach*g1)=s77.signals(2).values(1:601,1:10);
560         status_stable(1:601,(g1-1)+1:g1)=s77.signals(3).values(1:601,1);
561         CCT_stable(1,(g1-1)+1:g1)=CT1;
562         Time_stable(1:601,(g1-1)+1:g1)=s77.time(1:601,1);gg=2;
563     end
564     if g~=e
565         Vitesse_stable(1:601,nmach*(gg-1)+1:nmach*gg)=s77.signals(2).values(1:601,1:10);
566         status_stable(1:601,(gg-1)+1:gg)=s77.signals(3).values(1:601,1);
567         CCT_stable(1,(gg-1)+1:gg)=CT1;
568         Time_stable(1:601,(gg-1)+1:gg)=s77.time(1:601,1); gg=gg+1;
569     end
570     else
571         CT2=CT;
572     end
573 end
574     CCT=CT2; e=e+Dncy;
575 end
576     s7_L7=CT2; s7=mach;
577     A1=mach.time; Id_unstable= mach.signals(7).values;
578     w_unstable= mach.signals(2).values; Vd_unstable= mach.signals(5).values;
579     Vq_unstable= mach.signals(4).values; Iq_unstable= mach.signals(6).values;
580     % figure(i+34);
581 %   subplot(211);plot(mach.time,mach.signals(1).values);grid;subplot(212);
582 %   plot(mach.time,mach.signals(2).values);grid title(['CCT=',num2str(CT)]
583 % unstable
584     f=1;
585     Vitesse_unstable(1:601,nmach*(f-1)+1:nmach*f)=s7.signals(2).values(1:601,1:10);
586     status_unstable(1:601,(f-1)+1:f)=s7.signals(3).values(1:601,1);
587     Time_unstable(1:601,(f-1)+1:f)=s7.time(1:601,1); CCT_unstable=CT2;
588     fname = strcat('Data','_',char(txt{i,12}));
589     unstable=s7;stable=s77;
590     save(fname,'unstable','stable','CCT_stable','CCT_unstable');
591     save('.\Line7.mat','Vitesse_unstable','Vitesse_stable','status_stable','status_unstable',
592 'Time_unstable','Time_stable','CCT_stable','CCT_unstable');    %%L8
593     clear Vitesse_unstable; clear Vitesse_stable; clear status_stable; clear status_unstable;
593     clear Time_unstable, clear Time_stable; clear CCT_stable; clear CCT_unstable;

```

```

594     case 9
595 %% Line 8
596     Dncy=1/2; stab=0; e=Dncy; nmach=10;
597     mdl='NE39bus2_PQ'; load_system(mdl);
598     [~,txt,~] = xlsread('CB names.xlsx');
599     while stab==0
600         CT2=2*e-1; CT1=1;
601         while (CT2-CT1>e)
602             CT=(CT1+CT2)/2; x=1; y=CT;
603             % Fixer le switching time du breaker premier breaker
604             pathname = strcat(mdl,'/',char(txt{i,18}),',' char(txt{i,13}));
605             set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes',[x+y/60 10]);
606             % Fixer le switching time du breaker deuxième breaker
607             pathname = strcat(mdl,'/',char(txt{i,18}),',' char(txt{i,14}));
608             set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes',[x+y/60 10]);
609             % Fixer le temps de simulation du défaut.
610             pathname = strcat(mdl,'/',char(txt{i,18}),',' char(txt{i,17}));
611             set_param(pathname,'External','off','SwitchTimes',[x x+y/60]);
612             % Simulation du défaut 2PhAB-G
613             set_param(pathname,'FaultA',1); set_param(pathname,'FaultB',1);
614             set_param(pathname,'FaultC',1); set_param(pathname,'GroundFault',1);
615             LF = power_loadflow(mdl); sim(mdl);
616             % Fixer à une valeur supérieure à la simulation le temps de commutation des disjoncteurs
617             pathname = strcat(mdl,'/',char(txt{i,18}),',' char(txt{i,13}));
618             set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes','11');
619             % Fixer à une valeur supérieure à la simulation le temps de commutation des disjoncteurs
620             pathname = strcat(mdl,'/',char(txt{i,18}),',' char(txt{i,14}));
621             set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes','11');
622             % Fixer à une valeur supérieure à la simulation le temps de commutation des défauts
623             pathname = strcat(mdl,'/',char(txt{i,18}),',' char(txt{i,17}));
624             set_param(pathname,'External','off','SwitchTimes','11');
625             pathname = strcat(mdl,'/',char(txt{i,18}),',' char(txt{i,17}));
626             set_param(pathname,'FaultA',0); set_param(pathname,'FaultB',0);
627             set_param(pathname,'FaultC',0); set_param(pathname,'GroundFault',0);
628             stab=max(stability);g=2.5;
629         if stab==0
630             CT1=CT; s8_L88=CT1;
631             s88=mach;
632             A=mach.time; w_stable= mach.signals(2).values;
633             Vd_stable= mach.signals(5).values; Vq_stable= mach.signals(4).values;
634             Iq_stable= mach.signals(6).values; Id_stable= mach.signals(7).values;
635             % figure(i);subplot(211);plot(mach.time,mach.signals(1).values);grid;subplot(212);
636             % plot(mach.time,mach.signals(2).values);grid title(['CCT=',num2str(CT)]);
637             % stable
638             if g==e

```



```

639     g1=1;
640     Vitesse_stable(1:601,nmach*(g1-1)+1:nmach*g1)=s88.signals(2).values(1:601,1:10);
641     status_stable(1:601,(g1-1)+1:g1)=s88.signals(3).values(1:601,1);
642     CCT_stable(1,(g1-1)+1:g1)=CT1;
643     Time_stable(1:601,(g1-1)+1:g1)=s88.time(1:601,1);gg=2;
644     end
645     if g~=e
646         Vitesse_stable(1:601,nmach*(gg-1)+1:nmach*gg)=s88.signals(2).values(1:601,1:10);
647         status_stable(1:601,(gg-1)+1:gg)=s88.signals(3).values(1:601,1);
648         CCT_stable(1,(gg-1)+1:gg)=CT1;
649         Time_stable(1:601,(gg-1)+1:gg)=s88.time(1:601,1); gg=gg+1;
650     end
651     else
652         CT2=CT;
653     end
654 end
655     CCT=CT2; e=e+Dncy;
656 end
657     s8_L8=CT2; s8=mach;
658     A1=mach.time; Id_unstable= mach.signals(7).values;
659     w_unstable= mach.signals(2).values; Vd_unstable= mach.signals(5).values;
660     Vq_unstable= mach.signals(4).values; Iq_unstable= mach.signals(6).values;
661 %     figure(i+34);
662 %     subplot(211);plot(mach.time,mach.signals(1).values);grid;subplot(212);
663 %     plot(mach.time,mach.signals(2).values);grid title(['CCT=',num2str(CT)])
664 % unstable
665     f=1;
666     Vitesse_unstable(1:601,nmach*(f-1)+1:nmach*f)=s8.signals(2).values(1:601,1:10);
667     status_unstable(1:601,(f-1)+1:f)=s8.signals(3).values(1:601,1);
668     Time_unstable(1:601,(f-1)+1:f)=s8.time(1:601,1); CCT_unstable=CT2;
669     fname = strcat('Data','_',char(txt{i,12}));
670     unstable=s8;stable=s88;
671     save(fname,'unstable','stable','CCT_stable','CCT_unstable');
672     save('\Line8.mat','Vitesse_unstable','Vitesse_stable','status_stable','status_unstable',
673 'Time_unstable','Time_stable','CCT_stable','CCT_unstable');    %%L9
674     clear Vitesse_unstable; clear Vitesse_stable; clear status_stable; clear status_unstable;
675     clear Time_unstable, clear Time_stable; clear CCT_stable; clear CCT_unstable;
676
677     case 10
678     %% Line 9
679     Dncy=1/2; stab=0; e=Dncy; nmach=10;
680     mdl='NE39bus2_PQ'; load_system(mdl);
681     [~,txt,~] = xlsread('CB names.xlsx');
682     while stab==0
683         CT2=2*e-1; CT1=1;

```

```

683 while (CT2-CT1>e)
684     CT=(CT1+CT2)/2; x=1; y=CT;
685     % Fixer le switching time du breaker premier breaker
686     pathname = strcat(mdl, '/', char(txt{i,18}), '/', char(txt{i,13}));
687     set_param(pathname, 'SnubberResistance', 'inf', 'External', 'off', 'SwitchTimes', '[x+y/60 10]');
688     % Fixer le switching time du breaker deuxième breaker
689     pathname = strcat(mdl, '/', char(txt{i,18}), '/', char(txt{i,14}));
690     set_param(pathname, 'SnubberResistance', 'inf', 'External', 'off', 'SwitchTimes', '[x+y/60 10]');
691     % Fixer le temps de simulation du défaut.
692     pathname = strcat(mdl, '/', char(txt{i,18}), '/', char(txt{i,17}));
693     set_param(pathname, 'External', 'off', 'SwitchTimes', '[x x+y/60]');
694     % Simulation du défaut 2PhAB-G
695     set_param(pathname, 'FaultA', 1); set_param(pathname, 'FaultB', 1);
696     set_param(pathname, 'FaultC', 1); set_param(pathname, 'GroundFault', 1);
697     LF = power_loadflow(mdl); sim(mdl);
698     % Fixer à une valeur supérieure à la simulation le temps de commutation des disjoncteurs
699     pathname = strcat(mdl, '/', char(txt{i,18}), '/', char(txt{i,13}));
700     set_param(pathname, 'SnubberResistance', 'inf', 'External', 'off', 'SwitchTimes', '11');
701     % Fixer à une valeur supérieure à la simulation le temps de commutation des disjoncteurs
702     pathname = strcat(mdl, '/', char(txt{i,18}), '/', char(txt{i,14}));
703     set_param(pathname, 'SnubberResistance', 'inf', 'External', 'off', 'SwitchTimes', '11');
704     % Fixer à une valeur supérieure à la simulation le temps de commutation des défauts
705     pathname = strcat(mdl, '/', char(txt{i,18}), '/', char(txt{i,17}));
706     set_param(pathname, 'External', 'off', 'SwitchTimes', '11');
707     pathname = strcat(mdl, '/', char(txt{i,18}), '/', char(txt{i,17}));
708     set_param(pathname, 'FaultA', 0); set_param(pathname, 'FaultB', 0);
709     set_param(pathname, 'FaultC', 0); set_param(pathname, 'GroundFault', 0);
710     stab=max(stability);g=2.5;
711     if stab==0
712         CT1=CT; s9_L99=CT1;
713         s99=mach;
714         A=mach.time; w_stable= mach.signals(2).values;
715         Vd_stable= mach.signals(5).values; Vq_stable= mach.signals(4).values;
716         Iq_stable= mach.signals(6).values; Id_stable= mach.signals(7).values;
717     %     figure(i);subplot(211);plot(mach.time,mach.signals(1).values);grid;subplot(212);
718     %     plot(mach.time,mach.signals(2).values);grid title(['CCT=',num2str(CT)]);
719     % stable
720     if g==e
721         g1=1;
722         Vitesse_stable(1:601,nmach*(g1-1)+1:nmach*g1)=s99.signals(2).values(1:601,1:10);
723         status_stable(1:601,(g1-1)+1:g1)=s99.signals(3).values(1:601,1);
724         CCT_stable(1,(g1-1)+1:g1)=CT1;
725         Time_stable(1:601,(g1-1)+1:g1)=s99.time(1:601,1);gg=2;
726     end
727     if g~=e

```

```

728     Vitesse_stable(1:601,nmach*(gg-1)+1:nmach*gg)=s99.signals(2).values(1:601,1:10);
729     status_stable(1:601,(gg-1)+1:gg)=s99.signals(3).values(1:601,1);
730     CCT_stable(1,(gg-1)+1:gg)=CT1;
731     Time_stable(1:601,(gg-1)+1:gg)=s99.time(1:601,1); gg=gg+1;
732     end
733     else
734         CT2=CT;
735     end
736 end
737     CCT=CT2; e=e+Dncy;
738 end
739     s9_L9=CT2; s9=mach;
740     A1=mach.time; Id_unstable= mach.signals(7).values;
741     w_unstable= mach.signals(2).values; Vd_unstable= mach.signals(5).values;
742     Vq_unstable= mach.signals(4).values; Iq_unstable= mach.signals(6).values;
743 %     figure(i+34);
744 %     subplot(211);plot(mach.time,mach.signals(1).values);grid;subplot(212);
745 %     plot(mach.time,mach.signals(2).values);grid title(['CCT=',num2str(CT)]
746 % unstable
747     f=1;
748     Vitesse_unstable(1:520,nmach*(f-1)+1:nmach*f)=s9.signals(2).values(1:520,1:10);
749     status_unstable(1:520,(f-1)+1:f)=s9.signals(3).values(1:520,1);
750     Time_unstable(1:520,(f-1)+1:f)=s9.time(1:520,1); CCT_unstable=CT2;
751     fname = strcat('Data','_',char(txt{i,12}));
752     unstable=s9;stable=s99;
753     save(fname,'unstable','stable','CCT_stable','CCT_unstable');
754     save(['.Line9.mat','Vitesse_unstable','Vitesse_stable','status_stable','status_unstable',
755 'Time_unstable','Time_stable','CCT_stable','CCT_unstable'];%%L10
756     clear Vitesse_unstable; clear Vitesse_stable; clear status_stable; clear status_unstable;
757     clear Time_unstable, clear Time_stable; clear CCT_stable; clear CCT_unstable;
758
759     case 11
760     %% Line 10
761     Dncy=1/2; stab=0; e=Dncy; nmach=10;
762     mdl='NE39bus2_PQ'; load_system(mdl);
763     [~,txt,~] = xlsread('CB names.xlsx');
764     while stab==0
765     CT2=2*e-1; CT1=1;
766     while (CT2-CT1>e)
767         CT=(CT1+CT2)/2; x=1; y=CT;
768         % Fixer le switching time du breaker premier breaker
769         pathname = strcat(mdl,'/',char(txt{i,18}),'/',char(txt{i,13}));
770         set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes',[x+y/60 10]);
771         % Fixer le switching time du breaker deuxième breaker
772         pathname = strcat(mdl,'/',char(txt{i,18}),'/',char(txt{i,14}));

```

```

772 set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes',[x+y/60 10]);
773 % Fixer le temps de simulation du défaut.
774 pathname = strcat(mdl,'/',char(txt{i,18}),'/',char(txt{i,17}));
775 set_param(pathname,'External','off','SwitchTimes',[x x+y/60]);
776 % Simulation du défaut 2PhAB-G
777 set_param(pathname,'FaultA',1); set_param(pathname,'FaultB',1);
778 set_param(pathname,'FaultC',1); set_param(pathname,'GroundFault',1);
779 LF = power_loadflow(mdl); sim(mdl);
780 % Fixer à une valeur supérieure à la simulation le temps de commutation des disjoncteurs
781 pathname = strcat(mdl,'/',char(txt{i,18}),'/',char(txt{i,13}));
782 set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes','11');
783 % Fixer à une valeur supérieure à la simulation le temps de commutation des disjoncteurs
784 pathname = strcat(mdl,'/',char(txt{i,18}),'/',char(txt{i,14}));
785 set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes','11');
786 % Fixer à une valeur supérieure à la simulation le temps de commutation des défauts
787 pathname = strcat(mdl,'/',char(txt{i,18}),'/',char(txt{i,17}));
788 set_param(pathname,'External','off','SwitchTimes','11');
789 pathname = strcat(mdl,'/',char(txt{i,18}),'/',char(txt{i,17}));
790 set_param(pathname,'FaultA',0); set_param(pathname,'FaultB',0);
791 set_param(pathname,'FaultC',0); set_param(pathname,'GroundFault',0);
792 stab=max(stability);g=2.5;
793 if stab==0
794     CT1=CT; s10_L1010=CT1;
795     s1010=mach;
796     A=mach.time; w_stable= mach.signals(2).values;
797     Vd_stable= mach.signals(5).values; Vq_stable= mach.signals(4).values;
798     Iq_stable= mach.signals(6).values; Id_stable= mach.signals(7).values;
799 %     figure(i);subplot(211);plot(mach.time,mach.signals(1).values);grid;subplot(212);
800 %     plot(mach.time,mach.signals(2).values);grid title(['CCT=',num2str(CT)]);
801 % stable
802     if g==e
803         g1=1;
804         Vitesse_stable(1:601,nmach*(g1-1)+1:nmach*g1)=s1010.signals(2).values(1:601,1:10);
805         status_stable(1:601,(g1-1)+1:g1)=s1010.signals(3).values(1:601,1);
806         CCT_stable(1,(g1-1)+1:g1)=CT1;
807         Time_stable(1:601,(g1-1)+1:g1)=s1010.time(1:601,1);gg=2;
808     end
809     if g~=e
810         Vitesse_stable(1:601,nmach*(gg-1)+1:nmach*gg)=s1010.signals(2).values(1:601,1:10);
811         status_stable(1:601,(gg-1)+1:gg)=s1010.signals(3).values(1:601,1);
812         CCT_stable(1,(gg-1)+1:gg)=CT1;
813         Time_stable(1:601,(gg-1)+1:gg)=s1010.time(1:601,1); gg=gg+1;
814     end
815     else
816         CT2=CT;

```

```

817     end
818 end
819     e=e+Dncy; CCT=CT2;
820 end
821     s10_L10=CT2;
822     s10=mach;
823     A1=mach.time; Id_unstable= mach.signals(7).values;
824     w_unstable= mach.signals(2).values; Vd_unstable= mach.signals(5).values;
825     Vq_unstable= mach.signals(4).values; Iq_unstable= mach.signals(6).values;
826 %   figure(i+34);
827 %   subplot(211);plot(mach.time,mach.signals(1).values);grid;subplot(212);
828 %   plot(mach.time,mach.signals(2).values);grid title(['CCT=',num2str(CT)]
829 % unstable
830     f=1; sss1=size(s10.time,1);
831     Vitesse_unstable(1:sss1,nmach*(f-1)+1:nmach*f)=s10.signals(2).values(1:sss1,1:10);
832     status_unstable(1:sss1,(f-1)+1:f)=s10.signals(3).values(1:sss1,1);
833     Time_unstable(1:sss1,(f-1)+1:f)=s10.time(1:sss1,1); CCT_unstable=CT2;
834     fname = strcat('Data','_',char(txt{i,12}));
835     unstable=s10;stable=s10;
836     save(fname,'unstable','stable','CCT_stable','CCT_unstable');
837     save(['Line10.mat','Vitesse_unstable','Vitesse_stable','status_stable','status_unstable',
838 'Time_unstable','Time_stable','CCT_stable','CCT_unstable']);    %%L11
839 clear Vitesse_unstable; clear Vitesse_stable; clear status_stable; clear status_unstable;
840 clear Time_unstable, clear Time_stable; clear CCT_stable; clear CCT_unstable;
841
842 case 12
843 %% Line 11
844     Dncy=1/2; stab=0; e=Dncy; nmach=10;
845     mdl='NE39bus2_PQ'; load_system(mdl);
846     [~,txt,~] = xlsread('CB names.xlsx');
847 while stab==0
848     CT2=2*e-1; CT1=1;
849 while (CT2-CT1>e)
850     CT=(CT1+CT2)/2; x=1; y=CT;
851     % Fixer le switching time du breaker premier breaker
852     pathname = strcat(mdl,'/',char(txt{i,18}),'/',char(txt{i,13}));
853     set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes',[x+y/60 10]);
854     % Fixer le switching time du breaker deuxième breaker
855     pathname = strcat(mdl,'/',char(txt{i,18}),'/',char(txt{i,14}));
856     set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes',[x+y/60 10]);
857     % Fixer le temps de simulation du défaut.
858     pathname = strcat(mdl,'/',char(txt{i,18}),'/',char(txt{i,17}));
859     set_param(pathname,'External','off','SwitchTimes',[x x+y/60]);
860     % Simulation du défaut 2PhAB-G
861     set_param(pathname,'FaultA',1); set_param(pathname,'FaultB',1);

```

```

861 set_param(pathname,'FaultC',1); set_param(pathname,'GroundFault',1);
862 LF = power_loadflow mdl; sim mdl;
863 % Fixer à une valeur supérieure à la simulation le temps de commutation des disjoncteurs
864 pathname = strcat mdl,'/',char(txt{i,18}),',' char(txt{i,13}));
865 set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes','11');
866 % Fixer à une valeur supérieure à la simulation le temps de commutation des disjoncteurs
867 pathname = strcat mdl,'/',char(txt{i,18}),',' char(txt{i,14}));
868 set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes','11');
869 % Fixer à une valeur supérieure à la simulation le temps de commutation des défauts
870 pathname = strcat mdl,'/',char(txt{i,18}),',' char(txt{i,17}));
871 set_param(pathname,'External','off','SwitchTimes','11');
872 pathname = strcat mdl,'/',char(txt{i,18}),',' char(txt{i,17}));
873 set_param(pathname,'FaultA',0); set_param(pathname,'FaultB',0);
874 set_param(pathname,'FaultC',0); set_param(pathname,'GroundFault',0);
875 stab=max(stability);g=2.5;
876 if stab==0
877     CT1=CT; s11_L1111=CT1;
878     s1111=mach;
879     A=mach.time; w_stable= mach.signals(2).values;
880     Vd_stable= mach.signals(5).values; Vq_stable= mach.signals(4).values;
881     Iq_stable= mach.signals(6).values; Id_stable= mach.signals(7).values;
882 %     figure(i); subplot(211);plot(mach.time,mach.signals(1).values);grid;subplot(212);
883 %     plot(mach.time,mach.signals(2).values);grid title(['CCT=',num2str(CT)]);
884 % stable
885     if g==e
886         g1=1;
887         Vitesse_stable(1:601,nmach*(g1-1)+1:nmach*g1)=s1111.signals(2).values(1:601,1:10);
888         status_stable(1:601,(g1-1)+1:g1)=s1111.signals(3).values(1:601,1);
889         CCT_stable(1,(g1-1)+1:g1)=CT1;
890         Time_stable(1:601,(g1-1)+1:g1)=s1111.time(1:601,1);gg=2;
891     end
892     if g~=e
893         Vitesse_stable(1:601,nmach*(gg-1)+1:nmach*gg)=s1111.signals(2).values(1:601,1:10);
894         status_stable(1:601,(gg-1)+1:gg)=s1111.signals(3).values(1:601,1);
895         CCT_stable(1,(gg-1)+1:gg)=CT1;
896         Time_stable(1:601,(gg-1)+1:gg)=s1111.time(1:601,1); gg=gg+1;
897     end
898     else
899         CT2=CT;
900     end
901 end
902     e=e+Dncy;CCT=CT2;
903 end
904     s11_L11=CT2; s11=mach;
905     A1=mach.time; Id_unstable= mach.signals(7).values;

```

```

906     w_unstable= mach.signals(2).values; Vd_unstable= mach.signals(5).values;
907     Vq_unstable= mach.signals(4).values; Iq_unstable= mach.signals(6).values;
908 %     figure(i+34); subplot(211);plot(mach.time,mach.signals(1).values);grid;subplot(212);
909 %     plot(mach.time,mach.signals(2).values);grid title(['CCT=',num2str(CT)])
910 % unstable
911     f=1;sss2=size(s11.time,1);
912     Vitesse_unstable(1:sss2,nmach*(f-1)+1:nmach*f)=s11.signals(2).values(1:sss2,1:10);
913     status_unstable(1:sss2,(f-1)+1:f)=s11.signals(3).values(1:sss2,1);
914     Time_unstable(1:sss2,(f-1)+1:f)=s11.time(1:sss2,1); CCT_unstable=CT2;
915     fname = strcat('Data','_',char(txt{i,12}));
916     unstable=s11;stable=s1111;
917     save(fname,'unstable','stable','CCT_stable','CCT_unstable');
918     save('.\Line11.mat','Vitesse_unstable','Vitesse_stable','status_stable','status_unstable',
919 'Time_unstable','Time_stable','CCT_stable','CCT_unstable');    %%L12
    clear Vitesse_unstable; clear Vitesse_stable; clear status_stable; clear status_unstable;
920 clear Time_unstable, clear Time_stable; clear CCT_stable; clear CCT_unstable;
921
922     case 13
923 %% Line 12
924     Dncy=1/2; stab=0; e=Dncy; nmach=10;
925     mdl='NE39bus2_PQ'; load_system(mdl);
926     [~,txt,~] = xlsread('CB names.xlsx');
927 while stab==0
928 CT2=2*e-1; CT1=1;
929 while (CT2-CT1>e)
930     CT=(CT1+CT2)/2; x=1; y=CT;
931     % Fixer le switching time du breaker premier breaker
932     pathname = strcat(mdl, '/', char(txt{i,18}), '/', char(txt{i,13}));
933     set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes',[x+y/60 10]);
934     % Fixer le switching time du breaker deuxième breaker
935     pathname = strcat(mdl, '/', char(txt{i,18}), '/', char(txt{i,14}));
936     set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes',[x+y/60 10]);
937     % Fixer le temps de simulation du défaut.
938     pathname = strcat(mdl, '/', char(txt{i,18}), '/', char(txt{i,17}));
939     set_param(pathname,'External','off','SwitchTimes',[x x+y/60]);
940     % Simulation du défaut 2PhAB-G
941     set_param(pathname,'FaultA',1); set_param(pathname,'FaultB',1);
942     set_param(pathname,'FaultC',1); set_param(pathname,'GroundFault',1);
943     LF = power_loadflow(mdl); sim(mdl);
944     % Fixer à une valeur supérieure à la simulation le temps de commutation des disjoncteurs
945     pathname = strcat(mdl, '/', char(txt{i,18}), '/', char(txt{i,13}));
946     set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes','11');
947     % Fixer à une valeur supérieure à la simulation le temps de commutation des disjoncteurs
948     pathname = strcat(mdl, '/', char(txt{i,18}), '/', char(txt{i,14}));
949     set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes','11');

```

```

950 % Fixer à une valeur supérieure à la simulation le temps de commutation des défauts
951 pathname = strcat(mdl,'/',char(txt{i,18}),',' char(txt{i,17}));
952 set_param(pathname,'External','off','SwitchTimes','11');
953 pathname = strcat(mdl,'/',char(txt{i,18}),',' char(txt{i,17}));
954 set_param(pathname,'FaultA',0); set_param(pathname,'FaultB',0);
955 set_param(pathname,'FaultC',0); set_param(pathname,'GroundFault',0);
956 stab=max(stability);g=2.5;
957 if stab==0
958     CT1=CT; s12_L1212=CT1;
959     s1212=mach;
960     A=mach.time; w_stable= mach.signals(2).values;
961     Vd_stable= mach.signals(5).values; Vq_stable= mach.signals(4).values;
962     Iq_stable= mach.signals(6).values; Id_stable= mach.signals(7).values;
963 %     figure(i); subplot(211);plot(mach.time,mach.signals(1).values);grid;subplot(212);
964 %     plot(mach.time,mach.signals(2).values);grid title(['CCT=',num2str(CT)]);
965 % stable
966     if g==e
967         g1=1;
968         Vitesse_stable(1:601,nmach*(g1-1)+1:nmach*g1)=s1212.signals(2).values(1:601,1:10);
969         status_stable(1:601,(g1-1)+1:g1)=s1212.signals(3).values(1:601,1);
970         CCT_stable(1,(g1-1)+1:g1)=CT1;
971         Time_stable(1:601,(g1-1)+1:g1)=s1212.time(1:601,1);gg=2;
972     end
973     if g~=e
974         Vitesse_stable(1:601,nmach*(gg-1)+1:nmach*gg)=s1212.signals(2).values(1:601,1:10);
975         status_stable(1:601,(gg-1)+1:gg)=s1212.signals(3).values(1:601,1);
976         CCT_stable(1,(gg-1)+1:gg)=CT1;
977         Time_stable(1:601,(gg-1)+1:gg)=s1212.time(1:601,1); gg=gg+1;
978     end
979     else
980         CT2=CT;
981     end
982 end
983 e=e+Dncy;CCT=CT2;
984 end
985 s12_L12=CT2;s12=mach;
986 A1=mach.time; Id_unstable= mach.signals(7).values;
987 w_unstable= mach.signals(2).values; Vd_unstable= mach.signals(5).values;
988 Vq_unstable= mach.signals(4).values; Iq_unstable= mach.signals(6).values;
989 %     figure(i+34);
990 %     subplot(211);plot(mach.time,mach.signals(1).values);grid;subplot(212);
991 %     plot(mach.time,mach.signals(2).values);grid title(['CCT=',num2str(CT)]
992 % unstable
993 f=1;sss3=size(s12.time,1);
994 Vitesse_unstable(1:sss3,nmach*(f-1)+1:nmach*f)=s12.signals(2).values(1:sss3,1:10);

```



```

995     status_unstable(1:sss3,(f-1)+1:f)=s12.signals(3).values(1:sss3,1);
996     Time_unstable(1:sss3,(f-1)+1:f)=s12.time(1:sss3,1); CCT_unstable=CT2;
997     fname = strcat('Data','_',char(txt{i,12}));
998     unstable=s12;stable=s1212;
999     save(fname,'unstable','stable','CCT_stable','CCT_unstable');
1000     save('\Line13.mat','Vitesse_unstable','Vitesse_stable','status_stable','status_unstable',
1001 'Time_unstable','Time_stable','CCT_stable','CCT_unstable');    %% L13
    clear Vitesse_unstable; clear Vitesse_stable; clear status_stable; clear status_unstable;
1002     clear Time_unstable, clear Time_stable; clear CCT_stable; clear CCT_unstable;
1003
1004     case 14
1005     %% Line 13
1006     Dncy=1/2; stab=0; e=Dncy; nmach=10;
1007     mdl='NE39bus2_PQ'; load_system(mdl);
1008     [~,txt,~] = xlsread('CB names.xlsx');
1009     while stab==0
1010     CT2=2*e-1; CT1=1;
1011     while (CT2-CT1>e)
1012         CT=(CT1+CT2)/2; x=1; y=CT;
1013         % Fixer le switching time du breaker premier breaker
1014         pathname = strcat(mdl,'/',char(txt{i,18}),',' char(txt{i,13}));
1015         set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes',[x+y/60 10]);
1016         % Fixer le switching time du breaker deuxième breaker
1017         pathname = strcat(mdl,'/',char(txt{i,18}),',' char(txt{i,14}));
1018         set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes',[x+y/60 10]);
1019         % Fixer le temps de simulation du défaut.
1020         pathname = strcat(mdl,'/',char(txt{i,18}),',' char(txt{i,17}));
1021         set_param(pathname,'External','off','SwitchTimes',[x x+y/60]);
1022         % Simulation du défaut 2PhAB-G
1023         set_param(pathname,'FaultA',1); set_param(pathname,'FaultB',1);
1024         set_param(pathname,'FaultC',1); set_param(pathname,'GroundFault',1);
1025         LF = power_loadflow(mdl); sim(mdl);
1026         % Fixer à une valeur supérieure à la simulation le temps de commutation des disjoncteurs
1027         pathname = strcat(mdl,'/',char(txt{i,18}),',' char(txt{i,13}));
1028         set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes','11');
1029         % Fixer à une valeur supérieure à la simulation le temps de commutation des disjoncteurs
1030         pathname = strcat(mdl,'/',char(txt{i,18}),',' char(txt{i,14}));
1031         set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes','11');
1032         % Fixer à une valeur supérieure à la simulation le temps de commutation des défauts
1033         pathname = strcat(mdl,'/',char(txt{i,18}),',' char(txt{i,17}));
1034         set_param(pathname,'External','off','SwitchTimes','11');
1035         pathname = strcat(mdl,'/',char(txt{i,18}),',' char(txt{i,17}));
1036         set_param(pathname,'FaultA',0); set_param(pathname,'FaultB',0);
1037         set_param(pathname,'FaultC',0); set_param(pathname,'GroundFault',0);
1038         stab=max(stability);g=2.5;

```

```

1039     if stab==0
1040         CT1=CT; s13_L1313=CT1;
1041         s1313=mach;
1042         A=mach.time; w_stable= mach.signals(2).values;
1043         Vd_stable= mach.signals(5).values; Vq_stable= mach.signals(4).values;
1044         Iq_stable= mach.signals(6).values; Id_stable= mach.signals(7).values;
1045     %     figure(i); subplot(211);plot(mach.time,mach.signals(1).values);grid;subplot(212);
1046     %     plot(mach.time,mach.signals(2).values);grid title(['CCT=',num2str(CT)]);
1047     % stable
1048     if g==e
1049         g1=1;
1050         Vitesse_stable(1:601,nmach*(g1-1)+1:nmach*g1)=s1313.signals(2).values(1:601,1:10);
1051         status_stable(1:601,(g1-1)+1:g1)=s1313.signals(3).values(1:601,1);
1052         CCT_stable(1,(g1-1)+1:g1)=CT1;
1053         Time_stable(1:601,(g1-1)+1:g1)=s1313.time(1:601,1);gg=2;
1054     end
1055     if g~=e
1056         Vitesse_stable(1:601,nmach*(gg-1)+1:nmach*gg)=s1313.signals(2).values(1:601,1:10);
1057         status_stable(1:601,(gg-1)+1:gg)=s1313.signals(3).values(1:601,1);
1058         CCT_stable(1,(gg-1)+1:gg)=CT1;
1059         Time_stable(1:601,(gg-1)+1:gg)=s1313.time(1:601,1); gg=gg+1;
1060     end
1061     else
1062         CT2=CT;
1063     end
1064 end
1065     e=e+Dncy; CCT=CT2;
1066 end
1067     s13_L13=CT2; s13=mach;
1068     A1=mach.time; Id_unstable= mach.signals(7).values;
1069     w_unstable= mach.signals(2).values; Vd_unstable= mach.signals(5).values;
1070     Vq_unstable= mach.signals(4).values; Iq_unstable= mach.signals(6).values;
1071     %     figure(i+34);
1072     %     subplot(211);plot(mach.time,mach.signals(1).values);grid;subplot(212);
1073     %     plot(mach.time,mach.signals(2).values);grid title(['CCT=',num2str(CT)]
1074     % unstable
1075     f=1;sss13=size(s13.time,1);
1076     Vitesse_unstable(1:sss13,nmach*(f-1)+1:nmach*f)=s13.signals(2).values(1:sss13,1:10);
1077     status_unstable(1:sss13,(f-1)+1:f)=s13.signals(3).values(1:sss13,1);
1078     Time_unstable(1:sss13,(f-1)+1:f)=s13.time(1:sss13,1); CCT_unstable=CT2;
1079     fname = strcat('Data','_',char(txt{i,12}));
1080     unstable=s13;stable=s1313;
1081     save(fname,'unstable','stable','CCT_stable','CCT_unstable');
1082     save('\Line13.mat','Vitesse_unstable','Vitesse_stable','status_stable','status_unstable',
1083     'Time_unstable','Time_stable','CCT_stable','CCT_unstable');%%L14

```

```

clear Vitesse_unstable; clear Vitesse_stable; clear status_stable; clear status_unstable;
1084 clear Time_unstable, clear Time_stable; clear CCT_stable; clear CCT_unstable;
1085
1086     case 15
1087 %% Line 14
1088     Dncy=1/2; stab=0; e=Dncy; nmach=10;
1089     mdl='NE39bus2_PQ'; load_system(mdl);
1090     [~,txt,~] = xlsread('CB names.xlsx');
1091     while stab==0
1092         CT2=2*e-1; CT1=1;
1093         while (CT2-CT1>e)
1094             CT=(CT1+CT2)/2; x=1; y=CT;
1095             % Fixer le switching time du breaker premier breaker
1096             pathname = strcat(mdl,'/',char(txt{i,18}),',' char(txt{i,13}));
1097             set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes',[x+y/60 10]);
1098             % Fixer le switching time du breaker deuxième breaker
1099             pathname = strcat(mdl,'/',char(txt{i,18}),',' char(txt{i,14}));
1100             set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes',[x+y/60 10]);
1101             % Fixer le temps de simulation du défaut.
1102             pathname = strcat(mdl,'/',char(txt{i,18}),',' char(txt{i,17}));
1103             set_param(pathname,'External','off','SwitchTimes',[x x+y/60]);
1104             % Simulation du défaut 2PhAB-G
1105             set_param(pathname,'FaultA',1); set_param(pathname,'FaultB',1);
1106             set_param(pathname,'FaultC',1); set_param(pathname,'GroundFault',1);
1107             LF = power_loadflow(mdl); sim(mdl);
1108             % Fixer à une valeur supérieure à la simulation le temps de commutation des disjoncteurs
1109             pathname = strcat(mdl,'/',char(txt{i,18}),',' char(txt{i,13}));
1110             set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes','11');
1111             % Fixer à une valeur supérieure à la simulation le temps de commutation des disjoncteurs
1112             pathname = strcat(mdl,'/',char(txt{i,18}),',' char(txt{i,14}));
1113             set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes','11');
1114             % Fixer à une valeur supérieure à la simulation le temps de commutation des défauts
1115             pathname = strcat(mdl,'/',char(txt{i,18}),',' char(txt{i,17}));
1116             set_param(pathname,'External','off','SwitchTimes','11');
1117             pathname = strcat(mdl,'/',char(txt{i,18}),',' char(txt{i,17}));
1118             set_param(pathname,'FaultA',0); set_param(pathname,'FaultB',0);
1119             set_param(pathname,'FaultC',0); set_param(pathname,'GroundFault',0);
1120             stab=max(stability);g=2.5;
1121         if stab==0
1122             CT1=CT; s14_L1414=CT1;
1123             s1414=mach;
1124             A=mach.time; w_stable= mach.signals(2).values;
1125             Vd_stable= mach.signals(5).values; Vq_stable= mach.signals(4).values;
1126             Iq_stable= mach.signals(6).values; Id_stable= mach.signals(7).values;
1127     %     figure(i); subplot(211);plot(mach.time,mach.signals(1).values);grid;subplot(212);

```

```

1128 % plot(mach.time,mach.signals(2).values);grid title(['CCT=',num2str(CT)]);
1129 % stable
1130 if g==e
1131     g1=1;
1132     Vitesse_stable(1:601,nmach*(g1-1)+1:nmach*g1)=s1414.signals(2).values(1:601,1:10);
1133     status_stable(1:601,(g1-1)+1:g1)=s1414.signals(3).values(1:601,1);
1134     CCT_stable(1,(g1-1)+1:g1)=CT1;
1135     Time_stable(1:601,(g1-1)+1:g1)=s1414.time(1:601,1);gg=2;
1136 end
1137 if g~=e
1138     Vitesse_stable(1:601,nmach*(gg-1)+1:nmach*gg)=s1414.signals(2).values(1:601,1:10);
1139     status_stable(1:601,(gg-1)+1:gg)=s1414.signals(3).values(1:601,1);
1140     CCT_stable(1,(gg-1)+1:gg)=CT1;
1141     Time_stable(1:601,(gg-1)+1:gg)=s1414.time(1:601,1); gg=gg+1;
1142 end
1143 else
1144     CT2=CT;
1145 end
1146 end
1147 CCT=CT2;e=e+Dncy;
1148 end
1149 s14_L14=CT2; s14=mach;
1150 A1=mach.time; Id_unstable= mach.signals(7).values;
1151 w_unstable= mach.signals(2).values; Vd_unstable= mach.signals(5).values;
1152 Vq_unstable= mach.signals(4).values; Iq_unstable= mach.signals(6).values;
1153 % figure(i+34);
1154 % subplot(211);plot(mach.time,mach.signals(1).values);grid;subplot(212);
1155 % plot(mach.time,mach.signals(2).values);grid title(['CCT=',num2str(CT)]
1156 % unstable
1157 f=1;sss14=size(s14.time,1);
1158 Vitesse_unstable(1:sss14,nmach*(f-1)+1:nmach*f)=s14.signals(2).values(1:sss14,1:10);
1159 status_unstable(1:sss14,(f-1)+1:f)=s14.signals(3).values(1:sss14,1);
1160 Time_unstable(1:sss14,(f-1)+1:f)=s14.time(1:sss14,1); CCT_unstable=CT2;
1161 fname = strcat('Data','_',char(txt{i,12}));
1162 unstable=s14;stable=s1414;
1163 save(fname,'unstable','stable','CCT_stable','CCT_unstable');
1164 save('.\Line14.mat','Vitesse_unstable','Vitesse_stable','status_stable','status_unstable',
1165 'Time_unstable','Time_stable','CCT_stable','CCT_unstable');%%L15
clear Vitesse_unstable; clear Vitesse_stable; clear status_stable; clear status_unstable;
1166 clear Time_unstable, clear Time_stable; clear CCT_stable; clear CCT_unstable;
1167 case 16
1168 %% Line 15
1169 Dncy=1/2; stab=0; e=Dncy; nmach=10;
1170 mdl='NE39bus2_PQ'; load_system(mdl);
1171 [~,txt,~] = xlsread('CB names.xlsx');

```

```

1172 while stab==0
1173 CT2=2*e-1; CT1=1;
1174 while (CT2-CT1>e)
1175     CT=(CT1+CT2)/2; x=1; y=CT;
1176     % Fixer le switching time du breaker premier breaker
1177     pathname = strcat(mdl,'/',char(txt{i,18}),',' char(txt{i,13}));
1178     set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes',[x+y/60 10]);
1179     % Fixer le switching time du breaker deuxième breaker
1180     pathname = strcat(mdl,'/',char(txt{i,18}),',' char(txt{i,14}));
1181     set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes',[x+y/60 10]);
1182     % Fixer le temps de simulation du défaut.
1183     pathname = strcat(mdl,'/',char(txt{i,18}),',' char(txt{i,17}));
1184     set_param(pathname,'External','off','SwitchTimes',[x x+y/60]);
1185     % Simulation du défaut 2PhAB-G
1186     set_param(pathname,'FaultA',1); set_param(pathname,'FaultB',1);
1187     set_param(pathname,'FaultC',1); set_param(pathname,'GroundFault',1);
1188     LF = power_loadflow(mdl); sim(mdl);
1189     % Fixer à une valeur supérieure à la simulation le temps de commutation des disjoncteurs
1190     pathname = strcat(mdl,'/',char(txt{i,18}),',' char(txt{i,13}));
1191     set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes','11');
1192     % Fixer à une valeur supérieure à la simulation le temps de commutation des disjoncteurs
1193     pathname = strcat(mdl,'/',char(txt{i,18}),',' char(txt{i,14}));
1194     set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes','11');
1195     % Fixer à une valeur supérieure à la simulation le temps de commutation des défauts
1196     pathname = strcat(mdl,'/',char(txt{i,18}),',' char(txt{i,17}));
1197     set_param(pathname,'External','off','SwitchTimes','11');
1198     pathname = strcat(mdl,'/',char(txt{i,18}),',' char(txt{i,17}));
1199     set_param(pathname,'FaultA',0); set_param(pathname,'FaultB',0);
1200     set_param(pathname,'FaultC',0); set_param(pathname,'GroundFault',0);
1201     stab=max(stability);g=2.5;
1202 if stab==0
1203     CT1=CT; s15_L1515=CT1;
1204     s1515=mach; A=mach.time; w_stable= mach.signals(2).values;
1205     Vd_stable= mach.signals(5).values; Vq_stable= mach.signals(4).values;
1206     Iq_stable= mach.signals(6).values; Id_stable= mach.signals(7).values;
1207 %     figure(i);
1208 %     subplot(211);plot(mach.time,mach.signals(1).values);grid;subplot(212);
1209 %     plot(mach.time,mach.signals(2).values);grid title(['CCT=',num2str(CT)]);
1210 % stable
1211 if g==e
1212     g1=1;
1213     Vitesse_stable(1:601,nmach*(g1-1)+1:nmach*g1)=s1515.signals(2).values(1:601,1:10);
1214     status_stable(1:601,(g1-1)+1:g1)=s1515.signals(3).values(1:601,1);
1215     CCT_stable(1,(g1-1)+1:g1)=CT1;
1216     Time_stable(1:601,(g1-1)+1:g1)=s1515.time(1:601,1);gg=2;

```

```

1217 end
1218 if g~=e
1219     Vitesse_stable(1:601,nmach*(gg-1)+1:nmach*gg)=s1515.signals(2).values(1:601,1:10);
1220     status_stable(1:601,(gg-1)+1:gg)=s1515.signals(3).values(1:601,1);
1221     CCT_stable(1,(gg-1)+1:gg)=CT1;
1222     Time_stable(1:601,(gg-1)+1:gg)=s1515.time(1:601,1); gg=gg+1;
1223 end
1224 else
1225     CT2=CT;
1226 end
1227 end
1228     e=e+Dncy; CCT=CT2;
1229 end
1230     s15_L15=CT2; s15=mach;
1231     A1=mach.time; Id_unstable= mach.signals(7).values;
1232     w_unstable= mach.signals(2).values; Vd_unstable= mach.signals(5).values;
1233     Vq_unstable= mach.signals(4).values; Iq_unstable= mach.signals(6).values;
1234 %     figure(i+34);
1235 %     subplot(211);plot(mach.time,mach.signals(1).values);grid;subplot(212);
1236 %     plot(mach.time,mach.signals(2).values);grid title(['CCT=',num2str(CT)]
1237 % unstable
1238     f=1;sss15=size(s15.time,1);
1239     Vitesse_unstable(1:sss15,nmach*(f-1)+1:nmach*f)=s15.signals(2).values(1:sss15,1:10);
1240     status_unstable(1:sss15,(f-1)+1:f)=s15.signals(3).values(1:sss15,1);
1241     Time_unstable(1:sss15,(f-1)+1:f)=s15.time(1:sss15,1); CCT_unstable=CT2;
1242     fname = strcat('Data','_',char(txt{i,12}));
1243     unstable=s15;stable=s1515;
1244     save(fname,'unstable','stable','CCT_stable','CCT_unstable');
1245     save(['\Line15.mat','Vitesse_unstable','Vitesse_stable','status_stable','status_unstable',
1246 'Time_unstable','Time_stable','CCT_stable','CCT_unstable']); %%L16
1247     clear Vitesse_unstable; clear Vitesse_stable; clear status_stable; clear status_unstable;
1248     clear Time_unstable, clear Time_stable; clear CCT_stable; clear CCT_unstable;
1249
1249     case 17
1250 %% Line 16
1251     Dncy=1/2; stab=0; e=Dncy; nmach=10;
1252     mdl='NE39bus2_PQ'; load_system(mdl);
1253     [~,txt,~] = xlsread('CB names.xlsx');
1254     while stab==0
1255         CT2=2*e-1; CT1=1;
1256         while (CT2-CT1>e)
1257             CT=(CT1+CT2)/2; x=1; y=CT;
1258             % Fixer le switching time du breaker premier breaker
1259             pathname = strcat(mdl,'/',char(txt{i,18}),'/',char(txt{i,13}));
1260             set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes',[x+y/60 10]);

```

```

1261 % Fixer le switching time du breaker deuxième breaker
1262 pathname = strcat(mdl,'/',char(txt{i,18}),',' char(txt{i,14}));
1263 set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes',[x+y/60 10]);
1264 % Fixer le temps de simulation du défaut.
1265 pathname = strcat(mdl,'/',char(txt{i,18}),',' char(txt{i,17}));
1266 set_param(pathname,'External','off','SwitchTimes',[x x+y/60]);
1267 % Simulation du défaut 2PhAB-G
1268 set_param(pathname,'FaultA',1); set_param(pathname,'FaultB',1);
1269 set_param(pathname,'FaultC',1); set_param(pathname,'GroundFault',1);
1270 LF = power_loadflow(mdl); sim(mdl);
1271 % Fixer à une valeur supérieure à la simulation le temps de commutation des disjoncteurs
1272 pathname = strcat(mdl,'/',char(txt{i,18}),',' char(txt{i,13}));
1273 set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes','11');
1274 % Fixer à une valeur supérieure à la simulation le temps de commutation des disjoncteurs
1275 pathname = strcat(mdl,'/',char(txt{i,18}),',' char(txt{i,14}));
1276 set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes','11');
1277 % Fixer à une valeur supérieure à la simulation le temps de commutation des défauts
1278 pathname = strcat(mdl,'/',char(txt{i,18}),',' char(txt{i,17}));
1279 set_param(pathname,'External','off','SwitchTimes','11');
1280 pathname = strcat(mdl,'/',char(txt{i,18}),',' char(txt{i,17}));
1281 set_param(pathname,'FaultA',0); set_param(pathname,'FaultB',0);
1282 set_param(pathname,'FaultC',0); set_param(pathname,'GroundFault',0);
1283 stab=max(stability);g=2.5;
1284 if stab==0
1285     CT1=CT; s16_L1616=CT1;
1286     s1616=mach;
1287     A=mach.time; w_stable= mach.signals(2).values;
1288     Vd_stable= mach.signals(5).values; Vq_stable= mach.signals(4).values;
1289     Iq_stable= mach.signals(6).values; Id_stable= mach.signals(7).values;
1290 %     figure(i); subplot(211);plot(mach.time,mach.signals(1).values);grid;subplot(212);
1291 %     plot(mach.time,mach.signals(2).values);grid title(['CCT=',num2str(CT)]);
1292 % stable
1293     if g==e
1294         g1=1;
1295         Vitesse_stable(1:601,nmach*(g1-1)+1:nmach*g1)=s1616.signals(2).values(1:601,1:10);
1296         status_stable(1:601,(g1-1)+1:g1)=s1616.signals(3).values(1:601,1);
1297         CCT_stable(1,(g1-1)+1:g1)=CT1;
1298         Time_stable(1:601,(g1-1)+1:g1)=s1616.time(1:601,1);gg=2;
1299     end
1300     if g~=e
1301         Vitesse_stable(1:601,nmach*(gg-1)+1:nmach*gg)=s1616.signals(2).values(1:601,1:10);
1302         status_stable(1:601,(gg-1)+1:gg)=s1616.signals(3).values(1:601,1);
1303         CCT_stable(1,(gg-1)+1:gg)=CT1;
1304         Time_stable(1:601,(gg-1)+1:gg)=s1616.time(1:601,1); gg=gg+1;
1305     end

```

```

1306     else
1307         CT2=CT;
1308     end
1309 end
1310 CCT=CT2; e=e+Dncy;
1311 end
1312     s16_L16=CT2;
1313     s16=mach;
1314         A1=mach.time; Id_unstable= mach.signals(7).values;
1315     w_unstable= mach.signals(2).values; Vd_unstable= mach.signals(5).values;
1316     Vq_unstable= mach.signals(4).values; Iq_unstable= mach.signals(6).values;
1317 %     figure(i+34); subplot(211);plot(mach.time,mach.signals(1).values);grid;subplot(212);
1318 %     plot(mach.time,mach.signals(2).values);grid title(['CCT=',num2str(CT)]
1319
1320 % unstable
1321     f=1;sss16=size(s16.time,1);
1322     Vitesse_unstable(1:sss16,nmach*(f-1)+1:nmach*f)=s16.signals(2).values(1:sss16,1:10);
1323     status_unstable(1:sss16,(f-1)+1:f)=s16.signals(3).values(1:sss16,1);
1324     Time_unstable(1:sss16,(f-1)+1:f)=s16.time(1:sss16,1); CCT_unstable=CT2;
1325     fname = strcat('Data','_',char(txt{i,12}));
1326     unstable=s16;stable=s1616;
1327     save(fname,'unstable','stable','CCT_stable','CCT_unstable');
1328     save('.\Line16.mat','Vitesse_unstable','Vitesse_stable','status_stable','status_unstable',
1329 'Time_unstable','Time_stable','CCT_stable','CCT_unstable');    %%L17
1330 clear Vitesse_unstable; clear Vitesse_stable; clear status_stable; clear status_unstable;
1331 clear Time_unstable, clear Time_stable; clear CCT_stable; clear CCT_unstable;
1332
1333 case 18
1334 %% Line 17
1335     Dncy=1/2; stab=0; e=Dncy; nmach=10;
1336     mdl='NE39bus2_PQ'; load_system(mdl);
1337     [~,txt,~] = xlsread('CB names.xlsx');
1338     while stab==0
1339         CT2=2*e-1; CT1=1;
1340         while (CT2-CT1>e)
1341             CT=(CT1+CT2)/2; x=1; y=CT;
1342             % Fixer le switching time du breaker premier breaker
1343             pathname = strcat(mdl,'/',char(txt{i,18}),'/',char(txt{i,13}));
1344             set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes',[x+y/60 10]);
1345             % Fixer le switching time du breaker deuxième breaker
1346             pathname = strcat(mdl,'/',char(txt{i,18}),'/',char(txt{i,14}));
1347             set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes',[x+y/60 10]);
1348             % Fixer le temps de simulation du défaut.
1349             pathname = strcat(mdl,'/',char(txt{i,18}),'/',char(txt{i,17}));
1350             set_param(pathname,'External','off','SwitchTimes',[x x+y/60]);

```



```

1350 % Simulation du défaut 2PhAB-G
1351 set_param(pathname,'FaultA',1); set_param(pathname,'FaultB',1);
1352 set_param(pathname,'FaultC',1); set_param(pathname,'GroundFault',1);
1353 LF = power_loadflow mdl; sim mdl;
1354 % Fixer à une valeur supérieure à la simulation le temps de commutation des disjoncteurs
1355 pathname = strcat mdl, '/', char txt{i,18}, '/', char txt{i,13});
1356 set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes','11');
1357 % Fixer à une valeur supérieure à la simulation le temps de commutation des disjoncteurs
1358 pathname = strcat mdl, '/', char txt{i,18}, '/', char txt{i,14});
1359 set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes','11');
1360 % Fixer à une valeur supérieure à la simulation le temps de commutation des défauts
1361 pathname = strcat mdl, '/', char txt{i,18}, '/', char txt{i,17});
1362 set_param(pathname,'External','off','SwitchTimes','11');
1363 pathname = strcat mdl, '/', char txt{i,18}, '/', char txt{i,17});
1364 set_param(pathname,'FaultA',0); set_param(pathname,'FaultB',0);
1365 set_param(pathname,'FaultC',0); set_param(pathname,'GroundFault',0);
1366 stab=max(stability);g=2.5;
1367 if stab==0
1368     CT1=CT; s17_L1717=CT1;
1369     s1717=mach;
1370     A=mach.time; w_stable= mach.signals(2).values;
1371     Vd_stable= mach.signals(5).values; Vq_stable= mach.signals(4).values;
1372     Iq_stable= mach.signals(6).values; Id_stable= mach.signals(7).values;
1373 %     figure(i); subplot(211);plot(mach.time,mach.signals(1).values);grid;subplot(212);
1374 %     plot(mach.time,mach.signals(2).values);grid title(['CCT=',num2str(CT)]);
1375 % stable
1376     if g==e
1377         g1=1;
1378         Vitesse_stable(1:601,nmach*(g1-1)+1:nmach*g1)=s1717.signals(2).values(1:601,1:10);
1379         status_stable(1:601,(g1-1)+1:g1)=s1717.signals(3).values(1:601,1);
1380         CCT_stable(1,(g1-1)+1:g1)=CT1;
1381         Time_stable(1:601,(g1-1)+1:g1)=s1717.time(1:601,1);gg=2;
1382     end
1383     if g~=e
1384         Vitesse_stable(1:601,nmach*(gg-1)+1:nmach*gg)=s1717.signals(2).values(1:601,1:10);
1385         status_stable(1:601,(gg-1)+1:gg)=s1717.signals(3).values(1:601,1);
1386         CCT_stable(1,(gg-1)+1:gg)=CT1;
1387         Time_stable(1:601,(gg-1)+1:gg)=s1717.time(1:601,1); gg=gg+1;
1388     end
1389     else
1390         CT2=CT;
1391     end
1392 end
1393 CCT=CT2; e=e+Dncy;
1394 end

```

```

1395     s17_L17=CT2;s17=mach;
1396     A1=mach.time; Id_unstable= mach.signals(7).values;
1397     w_unstable= mach.signals(2).values; Vd_unstable= mach.signals(5).values;
1398     Vq_unstable= mach.signals(4).values; Iq_unstable= mach.signals(6).values;
1399 %     figure(i+34);
1400 %     subplot(211);plot(mach.time,mach.signals(1).values);grid;subplot(212);
1401 %     plot(mach.time,mach.signals(2).values);grid title(['CCT=',num2str(CT)]
1402 % unstable
1403     f=1;sss17=size(s17.time,1);
1404     Vitesse_unstable(1:sss17,nmach*(f-1)+1:nmach*f)=s17.signals(2).values(1:sss17,1:10);
1405     status_unstable(1:sss17,(f-1)+1:f)=s17.signals(3).values(1:sss17,1);
1406     Time_unstable(1:sss17,(f-1)+1:f)=s17.time(1:sss17,1); CCT_unstable=CT2;
1407     fname = strcat('Data','_',char(txt{i,12}));
1408     unstable=s17;stable=s1717;
1409     save(fname,'unstable','stable','CCT_stable','CCT_unstable');
1410     save(['Line17.mat','Vitesse_unstable','Vitesse_stable','status_stable','status_unstable',
1411 'Time_unstable','Time_stable','CCT_stable','CCT_unstable']);          %%L18
1412     clear Vitesse_unstable; clear Vitesse_stable; clear status_stable; clear status_unstable;
1413     clear Time_unstable, clear Time_stable; clear CCT_stable; clear CCT_unstable;
1414
1414     case 19
1415 %% Line 18
1416     Dncy=1/2; stab=0; e=Dncy; nmach=10;
1417     mdl='NE39bus2_PQ'; load_system(mdl);
1418     [~,txt,~] = xlsread('CB names.xlsx');
1419     while stab==0
1420     CT2=2*e-1; CT1=1;
1421     while (CT2-CT1>e)
1422         CT=(CT1+CT2)/2; x=1; y=CT;
1423         % Fixer le switching time du breaker premier breaker
1424         pathname = strcat(mdl,'/',char(txt{i,18}),'/',char(txt{i,13}));
1425         set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes',[x+y/60 10]);
1426         % Fixer le switching time du breaker deuxième breaker
1427         pathname = strcat(mdl,'/',char(txt{i,18}),'/',char(txt{i,14}));
1428         set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes',[x+y/60 10]);
1429         % Fixer le temps de simulation du défaut.
1430         pathname = strcat(mdl,'/',char(txt{i,18}),'/',char(txt{i,17}));
1431         set_param(pathname,'External','off','SwitchTimes',[x x+y/60]);
1432         % Simulation du défaut 2PhAB-G
1433         set_param(pathname,'FaultA',1); set_param(pathname,'FaultB',1);
1434         set_param(pathname,'FaultC',1); set_param(pathname,'GroundFault',1);
1435         LF = power_loadflow(mdl); sim(mdl);
1436         % Fixer à une valeur supérieure à la simulation le temps de commutation des disjoncteurs
1437         pathname = strcat(mdl,'/',char(txt{i,18}),'/',char(txt{i,13}));
1438         set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes','11');

```

```

1439 % Fixer à une valeur supérieure à la simulation le temps de commutation des disjoncteurs
1440 pathname = strcat(mdl,'/',char(txt{i,18}),',' char(txt{i,14}));
1441 set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes','11');
1442 % Fixer à une valeur supérieure à la simulation le temps de commutation des défauts
1443 pathname = strcat(mdl,'/',char(txt{i,18}),',' char(txt{i,17}));
1444 set_param(pathname,'External','off','SwitchTimes','11');
1445 pathname = strcat(mdl,'/',char(txt{i,18}),',' char(txt{i,17}));
1446 set_param(pathname,'FaultA',0); set_param(pathname,'FaultB',0);
1447 set_param(pathname,'FaultC',0); set_param(pathname,'GroundFault',0);
1448 stab=max(stability);g=2.5;
1449 if stab==0
1450     CT1=CT; s18_L18=CT1;
1451     s1818=mach;
1452     A=mach.time; w_stable= mach.signals(2).values;
1453     Vd_stable= mach.signals(5).values; Vq_stable= mach.signals(4).values;
1454     Iq_stable= mach.signals(6).values; Id_stable= mach.signals(7).values;
1455 %     figure(i);subplot(211);plot(mach.time,mach.signals(1).values);grid;subplot(212);
1456 %     plot(mach.time,mach.signals(2).values);grid title(['CCT=',num2str(CT)]);
1457 % stable
1458     if g==e
1459         g1=1;
1460         Vitesse_stable(1:601,nmach*(g1-1)+1:nmach*g1)=s1818.signals(2).values(1:601,1:10);
1461         status_stable(1:601,(g1-1)+1:g1)=s1818.signals(3).values(1:601,1);
1462         CCT_stable(1,(g1-1)+1:g1)=CT1;
1463         Time_stable(1:601,(g1-1)+1:g1)=s1818.time(1:601,1);gg=2;
1464     end
1465     if g~=e
1466         Vitesse_stable(1:601,nmach*(gg-1)+1:nmach*gg)=s1818.signals(2).values(1:601,1:10);
1467         status_stable(1:601,(gg-1)+1:gg)=s1818.signals(3).values(1:601,1);
1468         CCT_stable(1,(gg-1)+1:gg)=CT1;
1469         Time_stable(1:601,(gg-1)+1:gg)=s1818.time(1:601,1); gg=gg+1;
1470     end
1471     else
1472         CT2=CT;
1473
1474     end
1475 end
1476     e=e+Dncy; CCT=CT2;
1477 end
1478     s18_L18=CT2; s18=mach;
1479     A1=mach.time; Id_unstable= mach.signals(7).values;
1480     w_unstable= mach.signals(2).values; Vd_unstable= mach.signals(5).values;
1481     Vq_unstable= mach.signals(4).values; Iq_unstable= mach.signals(6).values;
1482 %     figure(i+34);
1483 %     subplot(211);plot(mach.time,mach.signals(1).values);grid;subplot(212);

```

```

1484 % plot(mach.time,mach.signals(2).values);grid title(['CCT=',num2str(CT)]
1485 % unstable
1486 f=1;sss18=size(s18.time,1);
1487 Vitesse_unstable(1:sss18,nmach*(f-1)+1:nmach*f)=s18.signals(2).values(1:sss18,1:10);
1488 status_unstable(1:sss18,(f-1)+1:f)=s18.signals(3).values(1:sss18,1);
1489 Time_unstable(1:sss18,(f-1)+1:f)=s18.time(1:sss18,1); CCT_unstable=CCT2;
1490 fname = strcat('Data','_',char(txt{i,12}));
1491 unstable=s18;stable=s1818;
1492 save(fname,'unstable','stable','CCT_stable','CCT_unstable');
1493 save('.\Line18.mat','Vitesse_unstable','Vitesse_stable','status_stable','status_unstable',
1494 Time_unstable','Time_stable','CCT_stable','CCT_unstable');%%L19
clear Vitesse_unstable; clear Vitesse_stable; clear status_stable; clear status_unstable;
1495 clear Time_unstable, clear Time_stable; clear CCT_stable; clear CCT_unstable;
1496
1497 case 20
1498 %% Line 19
1499 Dncy=1/2; stab=0; e=Dncy; nmach=10;
1500 mdl='NE39bus2_PQ'; load_system(mdl);
1501 [~,txt,~] = xlsread('CB names.xlsx');
1502 while stab==0
1503 CT2=2*e-1; CT1=1;
1504 while (CT2-CT1>e)
1505 CT=(CT1+CT2)/2; x=1; y=CT;
1506 % Fixer le switching time du breaker premier breaker
1507 pathname = strcat(mdl,'/',char(txt{i,18}),'/',char(txt{i,13}));
1508 set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes',[x+y/60 10]);
1509 % Fixer le switching time du breaker deuxième breaker
1510 pathname = strcat(mdl,'/',char(txt{i,18}),'/',char(txt{i,14}));
1511 set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes',[x+y/60 10]);
1512 % Fixer le temps de simulation du défaut.
1513 pathname = strcat(mdl,'/',char(txt{i,18}),'/',char(txt{i,17}));
1514 set_param(pathname,'External','off','SwitchTimes',[x x+y/60]);
1515 % Simulation du défaut 2PhAB-G
1516 set_param(pathname,'FaultA',1); set_param(pathname,'FaultB',1);
1517 set_param(pathname,'FaultC',1); set_param(pathname,'GroundFault',1);
1518 LF = power_loadflow(mdl); sim(mdl);
1519 % Fixer à une valeur supérieure à la simulation le temps de commutation des disjoncteurs
1520 pathname = strcat(mdl,'/',char(txt{i,18}),'/',char(txt{i,13}));
1521 set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes','11');
1522 % Fixer à une valeur supérieure à la simulation le temps de commutation des disjoncteurs
1523 pathname = strcat(mdl,'/',char(txt{i,18}),'/',char(txt{i,14}));
1524 set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes','11');
1525 % Fixer à une valeur supérieure à la simulation le temps de commutation des défauts
1526 pathname = strcat(mdl,'/',char(txt{i,18}),'/',char(txt{i,17}));
1527 set_param(pathname,'External','off','SwitchTimes','11');

```

```

1528     pathname = strcat(mdl, '/', char(txt{i,18}), '/', char(txt{i,17}));
1529     set_param(pathname, 'FaultA', 0); set_param(pathname, 'FaultB', 0);
1530     set_param(pathname, 'FaultC', 0); set_param(pathname, 'GroundFault', 0);
1531     stab=max(stability);g=2.5;
1532     if stab==0
1533         CT1=CT; s19_L1919=CT1;
1534         s1919=mach;
1535         A=mach.time; w_stable= mach.signals(2).values;
1536         Vd_stable= mach.signals(5).values; Vq_stable= mach.signals(4).values;
1537         Iq_stable= mach.signals(6).values; Id_stable= mach.signals(7).values;
1538     %     figure(i);subplot(211);plot(mach.time,mach.signals(1).values);grid;subplot(212);
1539     %     plot(mach.time,mach.signals(2).values);grid title(['CCT=',num2str(CT)]);
1540     % stable
1541     if g==e
1542         g1=1;
1543         Vitesse_stable(1:601,nmach*(g1-1)+1:nmach*g1)=s1919.signals(2).values(1:601,1:10);
1544         status_stable(1:601,(g1-1)+1:g1)=s1919.signals(3).values(1:601,1);
1545         CCT_stable(1,(g1-1)+1:g1)=CT1;
1546         Time_stable(1:601,(g1-1)+1:g1)=s1919.time(1:601,1);gg=2;
1547     end
1548     if g~=e
1549         Vitesse_stable(1:601,nmach*(gg-1)+1:nmach*gg)=s1919.signals(2).values(1:601,1:10);
1550         status_stable(1:601,(gg-1)+1:gg)=s1919.signals(3).values(1:601,1);
1551         CCT_stable(1,(gg-1)+1:gg)=CT1;
1552         Time_stable(1:601,(gg-1)+1:gg)=s1919.time(1:601,1); gg=gg+1;
1553     end
1554     else
1555         CT2=CT;
1556     end
1557 end
1558 end
1559     e=e+Dncy; CCT=CT2;
1560 end
1561     s19_L19=CT2;s19=mach;
1562     A1=mach.time; Id_unstable= mach.signals(7).values;
1563     w_unstable= mach.signals(2).values; Vd_unstable= mach.signals(5).values;
1564     Vq_unstable= mach.signals(4).values; Iq_unstable= mach.signals(6).values;
1565     %     figure(i+34);
1566     %     subplot(211);plot(mach.time,mach.signals(1).values);grid;subplot(212);
1567     %     plot(mach.time,mach.signals(2).values);grid title(['CCT=',num2str(CT)]
1568     % unstable
1569     f=1;sss19=size(s19.time,1);
1570     Vitesse_unstable(1:sss19,nmach*(f-1)+1:nmach*f)=s19.signals(2).values(1:sss19,1:10);
1571     status_unstable(1:sss19,(f-1)+1:f)=s19.signals(3).values(1:sss19,1);
1572     Time_unstable(1:sss19,(f-1)+1:f)=s19.time(1:sss19,1); CCT_unstable=CT2;

```

```

1573     fname = strcat('Data','_',char(txt{i,12}));
1574     unstable=s19;stable=s1919;
1575     save(fname,'unstable','stable','CCT_stable','CCT_unstable');
1576     save('\Line19.mat','Vitesse_unstable','Vitesse_stable','status_stable','status_unstable',
1577 'Time_unstable','Time_stable','CCT_stable','CCT_unstable');%%L20
    clear Vitesse_unstable; clear Vitesse_stable; clear status_stable; clear status_unstable;
1578 clear Time_unstable, clear Time_stable; clear CCT_stable; clear CCT_unstable;
1579
1580     case 21
1581 %% Line 20
1582     Dncy=1/2; stab=0; e=Dncy; nmach=10;
1583     mdl='NE39bus2_PQ'; load_system(mdl);
1584     [~,txt,~] = xlsread('CB names.xlsx');
1585 while stab==0
1586 CT2=2*e-1; CT1=1;
1587 while (CT2-CT1>e)
1588     CT=(CT1+CT2)/2; x=1; y=CT;
1589     % Fixer le switching time du breaker premier breaker
1590     pathname = strcat(mdl,'/',char(txt{i,18}),'/',char(txt{i,13}));
1591     set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes',[x+y/60 10]);
1592     % Fixer le switching time du breaker deuxième breaker
1593     pathname = strcat(mdl,'/',char(txt{i,18}),'/',char(txt{i,14}));
1594     set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes',[x+y/60 10]);
1595     % Fixer le temps de simulation du défaut.
1596     pathname = strcat(mdl,'/',char(txt{i,18}),'/',char(txt{i,17}));
1597     set_param(pathname,'External','off','SwitchTimes',[x x+y/60]);
1598     % Simulation du défaut 2PhAB-G
1599     set_param(pathname,'FaultA',1); set_param(pathname,'FaultB',1);
1600     set_param(pathname,'FaultC',1); set_param(pathname,'GroundFault',1);
1601     LF = power_loadflow(mdl); sim(mdl);
1602     % Fixer à une valeur supérieure à la simulation le temps de commutation des disjoncteurs
1603     pathname = strcat(mdl,'/',char(txt{i,18}),'/',char(txt{i,13}));
1604     set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes','11');
1605     % Fixer à une valeur supérieure à la simulation le temps de commutation des disjoncteurs
1606     pathname = strcat(mdl,'/',char(txt{i,18}),'/',char(txt{i,14}));
1607     set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes','11');
1608     % Fixer à une valeur supérieure à la simulation le temps de commutation des défauts
1609     pathname = strcat(mdl,'/',char(txt{i,18}),'/',char(txt{i,17}));
1610     set_param(pathname,'External','off','SwitchTimes','11');
1611     pathname = strcat(mdl,'/',char(txt{i,18}),'/',char(txt{i,17}));
1612     set_param(pathname,'FaultA',0); set_param(pathname,'FaultB',0);
1613     set_param(pathname,'FaultC',0); set_param(pathname,'GroundFault',0);
1614     stab=max(stability);g=2.5;
1615 if stab==0
1616     CT1=CT; s20_L2020=CT1;

```

```

1617     s2020=mach;
1618     A=mach.time; w_stable= mach.signals(2).values;
1619     Vd_stable= mach.signals(5).values; Vq_stable= mach.signals(4).values;
1620     Iq_stable= mach.signals(6).values; Id_stable= mach.signals(7).values;
1621 %     figure(i); subplot(211);plot(mach.time,mach.signals(1).values);grid;subplot(212);
1622 %     plot(mach.time,mach.signals(2).values);grid title(['CCT=',num2str(CT)]);
1623 % stable
1624     if g==e
1625         g1=1;
1626         Vitesse_stable(1:601,nmach*(g1-1)+1:nmach*g1)=s2020.signals(2).values(1:601,1:10);
1627         status_stable(1:601,(g1-1)+1:g1)=s2020.signals(3).values(1:601,1);
1628         CCT_stable(1,(g1-1)+1:g1)=CT1;
1629         Time_stable(1:601,(g1-1)+1:g1)=s2020.time(1:601,1);gg=2;
1630     end
1631     if g~=e
1632         Vitesse_stable(1:601,nmach*(gg-1)+1:nmach*gg)=s2020.signals(2).values(1:601,1:10);
1633         status_stable(1:601,(gg-1)+1:gg)=s2020.signals(3).values(1:601,1);
1634         CCT_stable(1,(gg-1)+1:gg)=CT1;
1635         Time_stable(1:601,(gg-1)+1:gg)=s2020.time(1:601,1); gg=gg+1;
1636     end
1637     else
1638         CT2=CT;
1639     end
1640 end
1641     CCT=CT2; e=e+Dncy;
1642 end
1643     s20_L20=CT2;s20=mach;
1644     A1=mach.time; Id_unstable= mach.signals(7).values;
1645     w_unstable= mach.signals(2).values; Vd_unstable= mach.signals(5).values;
1646     Vq_unstable= mach.signals(4).values; Iq_unstable= mach.signals(6).values;
1647 %     figure(i+34);
1648 %     subplot(211);plot(mach.time,mach.signals(1).values);grid;subplot(212);
1649 %     plot(mach.time,mach.signals(2).values);grid title(['CCT=',num2str(CT)]
1650 % unstable
1651     f=1;sss20=size(s20.time,1);
1652     Vitesse_unstable(1:sss20,nmach*(f-1)+1:nmach*f)=s20.signals(2).values(1:sss20,1:10);
1653     status_unstable(1:sss20,(f-1)+1:f)=s20.signals(3).values(1:sss20,1);
1654     Time_unstable(1:sss20,(f-1)+1:f)=s20.time(1:sss20,1); CCT_unstable=CT2;
1655     fname = strcat('Data','_',char(txt{i,12}));
1656     unstable=s20;stable=s2020;
1657     save(fname,'unstable','stable','CCT_stable','CCT_unstable');
1658     save('.\Line20.mat','Vitesse_unstable','Vitesse_stable','status_stable','status_unstable',
1659 'Time_unstable','Time_stable','CCT_stable','CCT_unstable');%%L21
1660 clear Vitesse_unstable; clear Vitesse_stable; clear status_stable; clear status_unstable;
clear Time_unstable; clear Time_stable; clear CCT_stable; clear CCT_unstable;

```

```

1661
1662     case 22
1663 %% Line 21
1664     Dncy=1/2; stab=0; e=Dncy; nmach=10;
1665     mdl='NE39bus2_PQ'; load_system(mdl);
1666     [~,txt,~] = xlsread('CB names.xlsx');
1667     while stab==0
1668         CT2=2*e-1; CT1=1;
1669         while (CT2-CT1>e)
1670             CT=(CT1+CT2)/2; x=1; y=CT;
1671             % Fixer le switching time du breaker premier breaker
1672             pathname = strcat(mdl,'/',char(txt{i,18}),',' char(txt{i,13}));
1673             set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes',[x+y/60 10]);
1674             % Fixer le switching time du breaker deuxième breaker
1675             pathname = strcat(mdl,'/',char(txt{i,18}),',' char(txt{i,14}));
1676             set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes',[x+y/60 10]);
1677             % Fixer le temps de simulation du défaut.
1678             pathname = strcat(mdl,'/',char(txt{i,18}),',' char(txt{i,17}));
1679             set_param(pathname,'External','off','SwitchTimes',[x x+y/60]);
1680             % Simulation du défaut 2PhAB-G
1681             set_param(pathname,'FaultA',1); set_param(pathname,'FaultB',1);
1682             set_param(pathname,'FaultC',1); set_param(pathname,'GroundFault',1);
1683             LF = power_loadflow(mdl); sim(mdl);
1684             % Fixer à une valeur supérieure à la simulation le temps de commutation des disjoncteurs
1685             pathname = strcat(mdl,'/',char(txt{i,18}),',' char(txt{i,13}));
1686             set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes','11');
1687             % Fixer à une valeur supérieure à la simulation le temps de commutation des disjoncteurs
1688             pathname = strcat(mdl,'/',char(txt{i,18}),',' char(txt{i,14}));
1689             set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes','11');
1690             % Fixer à une valeur supérieure à la simulation le temps de commutation des défauts
1691             pathname = strcat(mdl,'/',char(txt{i,18}),',' char(txt{i,17}));
1692             set_param(pathname,'External','off','SwitchTimes','11');
1693             pathname = strcat(mdl,'/',char(txt{i,18}),',' char(txt{i,17}));
1694             set_param(pathname,'FaultA',0); set_param(pathname,'FaultB',0);
1695             set_param(pathname,'FaultC',0); set_param(pathname,'GroundFault',0);
1696             stab=max(stability);g=2.5;
1697         if stab==0
1698             CT1=CT; s21_L2121=CT1;
1699             s2121=mach;
1700             A=mach.time; w_stable= mach.signals(2).values;
1701             Vd_stable= mach.signals(5).values; Vq_stable= mach.signals(4).values;
1702             Iq_stable= mach.signals(6).values; Id_stable= mach.signals(7).values;
1703         %     figure(i); subplot(211);plot(mach.time,mach.signals(1).values);grid;subplot(212);
1704         %     plot(mach.time,mach.signals(2).values);grid title(['CCT=',num2str(CT)]);
1705         % stable

```



```

1706     if g==e
1707         g1=1;
1708         Vitesse_stable(1:601,nmach*(g1-1)+1:nmach*g1)=s2121.signals(2).values(1:601,1:10);
1709         status_stable(1:601,(g1-1)+1:g1)=s2121.signals(3).values(1:601,1);
1710         CCT_stable(1,(g1-1)+1:g1)=CT1;
1711         Time_stable(1:601,(g1-1)+1:g1)=s2121.time(1:601,1);gg=2;
1712     end
1713     if g~=e
1714         Vitesse_stable(1:601,nmach*(gg-1)+1:nmach*gg)=s2121.signals(2).values(1:601,1:10);
1715         status_stable(1:601,(gg-1)+1:gg)=s2121.signals(3).values(1:601,1);
1716         CCT_stable(1,(gg-1)+1:gg)=CT1;
1717         Time_stable(1:601,(gg-1)+1:gg)=s2121.time(1:601,1); gg=gg+1;
1718     end
1719     else
1720         CT2=CT;
1721
1722     end
1723 end
1724     e=e+Dncy; CCT=CT2;
1725 end
1726     s21=mach; s21_L21=CT2;
1727     A1=mach.time; Id_unstable= mach.signals(7).values;
1728     w_unstable= mach.signals(2).values; Vd_unstable= mach.signals(5).values;
1729     Vq_unstable= mach.signals(4).values; Iq_unstable= mach.signals(6).values;
1730 %     figure(i+34);
1731 %     subplot(211);plot(mach.time,mach.signals(1).values);grid;subplot(212);
1732 %     plot(mach.time,mach.signals(2).values);grid title(['CCT=',num2str(CT)])
1733 %     unstable
1734     f=1;sss21=size(s21.time,1);
1735     Vitesse_unstable(1:sss21,nmach*(f-1)+1:nmach*f)=s21.signals(2).values(1:sss21,1:10);
1736     status_unstable(1:sss21,(f-1)+1:f)=s21.signals(3).values(1:sss21,1);
1737     Time_unstable(1:sss21,(f-1)+1:f)=s21.time(1:sss21,1); CCT_unstable=CT2;
1738     fname = strcat('Data','_',char(txt{i,12}));
1739     unstable=s21;stable=s2121;
1740     save(fname,'unstable','stable','CCT_stable','CCT_unstable');
1741     save(['\Line21.mat','Vitesse_unstable','Vitesse_stable','status_stable','status_unstable',
1742 'Time_unstable','Time_stable','CCT_stable','CCT_unstable']);%%L22
1743     clear Vitesse_unstable; clear Vitesse_stable; clear status_stable; clear status_unstable;
1744     clear Time_unstable, clear Time_stable; clear CCT_stable; clear CCT_unstable;
1745
1746     case 23
1747     %% Line 22
1748     Dncy=1/2; stab=0; e=Dncy; nmach=10;
1749     mdl='NE39bus2_PQ'; load_system(mdl);
1750     [~,txt,~] = xlsread('CB names.xlsx');

```

```

1750 while stab==0
1751 CT2=2*e-1; CT1=1;
1752 while (CT2-CT1>e)
1753     CT=(CT1+CT2)/2; x=1; y=CT;
1754     % Fixer le switching time du breaker premier breaker
1755     pathname = strcat(mdl,'/',char(txt{i,18}),',' char(txt{i,13}));
1756     set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes',[x+y/60 10]);
1757     % Fixer le switching time du breaker deuxième breaker
1758     pathname = strcat(mdl,'/',char(txt{i,18}),',' char(txt{i,14}));
1759     set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes',[x+y/60 10]);
1760     % Fixer le temps de simulation du défaut.
1761     pathname = strcat(mdl,'/',char(txt{i,18}),',' char(txt{i,17}));
1762     set_param(pathname,'External','off','SwitchTimes',[x x+y/60]);
1763     % Simulation du défaut 2PhAB-G
1764     set_param(pathname,'FaultA',1); set_param(pathname,'FaultB',1);
1765     set_param(pathname,'FaultC',1); set_param(pathname,'GroundFault',1);
1766     LF = power_loadflow(mdl); sim(mdl);
1767     % Fixer à une valeur supérieure à la simulation le temps de commutation des disjoncteurs
1768     pathname = strcat(mdl,'/',char(txt{i,18}),',' char(txt{i,13}));
1769     set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes','11');
1770     % Fixer à une valeur supérieure à la simulation le temps de commutation des disjoncteurs
1771     pathname = strcat(mdl,'/',char(txt{i,18}),',' char(txt{i,14}));
1772     set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes','11');
1773     % Fixer à une valeur supérieure à la simulation le temps de commutation des défauts
1774     pathname = strcat(mdl,'/',char(txt{i,18}),',' char(txt{i,17}));
1775     set_param(pathname,'External','off','SwitchTimes','11');
1776     pathname = strcat(mdl,'/',char(txt{i,18}),',' char(txt{i,17}));
1777     set_param(pathname,'FaultA',0); set_param(pathname,'FaultB',0);
1778     set_param(pathname,'FaultC',0); set_param(pathname,'GroundFault',0);
1779     stab=max(stability);g=2.5;
1780 if stab==0
1781     CT1=CT; s22_L2222=CT1;
1782     s2222=mach;
1783     A=mach.time; w_stable= mach.signals(2).values;
1784     Vd_stable= mach.signals(5).values; Vq_stable= mach.signals(4).values;
1785     Iq_stable= mach.signals(6).values; Id_stable= mach.signals(7).values;
1786 %     figure(i);subplot(211);plot(mach.time,mach.signals(1).values);grid;subplot(212);
1787 %     plot(mach.time,mach.signals(2).values);grid title(['CCT=',num2str(CT)]);
1788 % stable
1789     if g==e
1790         g1=1;
1791         Vitesse_stable(1:601,nmach*(g1-1)+1:nmach*g1)=s2222.signals(2).values(1:601,1:10);
1792         status_stable(1:601,(g1-1)+1:g1)=s2222.signals(3).values(1:601,1);
1793         CCT_stable(1,(g1-1)+1:g1)=CT1;
1794         Time_stable(1:601,(g1-1)+1:g1)=s2222.time(1:601,1);gg=2;

```

```

1795     end
1796     if g~=e
1797         Vitesse_stable(1:601,nmach*(gg-1)+1:nmach*gg)=s2222.signals(2).values(1:601,1:10);
1798         status_stable(1:601,(gg-1)+1:gg)=s2222.signals(3).values(1:601,1);
1799         CCT_stable(1,(gg-1)+1:gg)=CT1;
1800         Time_stable(1:601,(gg-1)+1:gg)=s2222.time(1:601,1); gg=gg+1;
1801     end
1802     else
1803         CT2=CT;
1804     end
1805 end
1806     e=e+Dncy; CCT=CT2;
1807 end
1808     s22_L22=CT2;s22=mach;
1809     A1=mach.time; Id_unstable= mach.signals(7).values;
1810     w_unstable= mach.signals(2).values; Vd_unstable= mach.signals(5).values;
1811     Vq_unstable= mach.signals(4).values; Iq_unstable= mach.signals(6).values;
1812 %     figure(i+34);
1813 %     subplot(211);plot(mach.time,mach.signals(1).values);grid;subplot(212);
1814 %     plot(mach.time,mach.signals(2).values);grid title(['CCT=',num2str(CT)]
1815 % unstable
1816     f=1;sss22=size(s22.time,1);
1817     Vitesse_unstable(1:sss22,nmach*(f-1)+1:nmach*f)=s22.signals(2).values(1:sss22,1:10);
1818     status_unstable(1:sss22,(f-1)+1:f)=s22.signals(3).values(1:sss22,1);
1819     Time_unstable(1:sss22,(f-1)+1:f)=s22.time(1:sss22,1); CCT_unstable=CT2;
1820     fname = strcat('Data','_',char(txt{i,12}));
1821     unstable=s22;stable=s2222;
1822     save(fname,'unstable','stable','CCT_stable','CCT_unstable');
1823     save(['\Line22.mat','Vitesse_unstable','Vitesse_stable','status_stable','status_unstable',
1824 'Time_unstable','Time_stable','CCT_stable','CCT_unstable']);%%L23
1825     clear Vitesse_unstable; clear Vitesse_stable; clear status_stable; clear status_unstable;
1826     clear Time_unstable, clear Time_stable; clear CCT_stable; clear CCT_unstable;
1827
1827     case 24
1828 %% Line 23
1829     Dncy=1/2; stab=0; e=Dncy; nmach=10;
1830     mdl='NE39bus2_PQ'; load_system(mdl);
1831     [~,txt,~] = xlsread('CB names.xlsx');
1832     while stab==0
1833         CT2=2*e-1; CT1=1;
1834         while (CT2-CT1>e)
1835             CT=(CT1+CT2)/2; x=1; y=CT;
1836             % Fixer le switching time du breaker premier breaker
1837             pathname = strcat(mdl,'/',char(txt{i,18}),'/',char(txt{i,13}));
1838             set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes',[x+y/60 10]);

```

```

1839 % Fixer le switching time du breaker deuxième breaker
1840 pathname = strcat(mdl,'/',char(txt{i,18}),',' char(txt{i,14}));
1841 set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes',[x+y/60 10]);
1842 % Fixer le temps de simulation du défaut.
1843 pathname = strcat(mdl,'/',char(txt{i,18}),',' char(txt{i,17}));
1844 set_param(pathname,'External','off','SwitchTimes',[x x+y/60]);
1845 % Simulation du défaut 2PhAB-G
1846 set_param(pathname,'FaultA',1); set_param(pathname,'FaultB',1);
1847 set_param(pathname,'FaultC',1); set_param(pathname,'GroundFault',1);
1848 LF = power_loadflow(mdl); sim(mdl);
1849 % Fixer à une valeur supérieure à la simulation le temps de commutation des disjoncteurs
1850 pathname = strcat(mdl,'/',char(txt{i,18}),',' char(txt{i,13}));
1851 set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes','11');
1852 % Fixer à une valeur supérieure à la simulation le temps de commutation des disjoncteurs
1853 pathname = strcat(mdl,'/',char(txt{i,18}),',' char(txt{i,14}));
1854 set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes','11');
1855 % Fixer à une valeur supérieure à la simulation le temps de commutation des défauts
1856 pathname = strcat(mdl,'/',char(txt{i,18}),',' char(txt{i,17}));
1857 set_param(pathname,'External','off','SwitchTimes','11');
1858 pathname = strcat(mdl,'/',char(txt{i,18}),',' char(txt{i,17}));
1859 set_param(pathname,'FaultA',0); set_param(pathname,'FaultB',0);
1860 set_param(pathname,'FaultC',0); set_param(pathname,'GroundFault',0);
1861 stab=max(stability);g=2.5;
1862 if stab==0
1863     CT1=CT; s23_L2323=CT1;
1864     s2323=mach;
1865     A=mach.time; w_stable= mach.signals(2).values;
1866     Vd_stable= mach.signals(5).values; Vq_stable= mach.signals(4).values;
1867     Iq_stable= mach.signals(6).values; Id_stable= mach.signals(7).values;
1868 %     figure(i); subplot(211);plot(mach.time,mach.signals(1).values);grid;subplot(212);
1869 %     plot(mach.time,mach.signals(2).values);grid title(['CCT=',num2str(CT)]);
1870 % stable
1871     if g==e
1872         g1=1;
1873         Vitesse_stable(1:601,nmach*(g1-1)+1:nmach*g1)=s2323.signals(2).values(1:601,1:10);
1874         status_stable(1:601,(g1-1)+1:g1)=s2323.signals(3).values(1:601,1);
1875         CCT_stable(1,(g1-1)+1:g1)=CT1;
1876         Time_stable(1:601,(g1-1)+1:g1)=s2323.time(1:601,1);gg=2;
1877     end
1878     if g~=e
1879         Vitesse_stable(1:601,nmach*(gg-1)+1:nmach*gg)=s2323.signals(2).values(1:601,1:10);
1880         status_stable(1:601,(gg-1)+1:gg)=s2323.signals(3).values(1:601,1);
1881         CCT_stable(1,(gg-1)+1:gg)=CT1;
1882         Time_stable(1:601,(gg-1)+1:gg)=s2323.time(1:601,1); gg=gg+1;
1883     end

```

```

1884     else
1885         CT2=CT;
1886     end
1887 end
1888     CCT=CT2; e=e+Dncy;
1889 end
1890     s23_L23=CT2;s23=mach;
1891     A1=mach.time; Id_unstable= mach.signals(7).values;
1892     w_unstable= mach.signals(2).values; Vd_unstable= mach.signals(5).values;
1893     Vq_unstable= mach.signals(4).values; Iq_unstable= mach.signals(6).values;
1894 % figure(i+34);
1895 %     subplot(211);plot(mach.time,mach.signals(1).values);grid;subplot(212);
1896 %     plot(mach.time,mach.signals(2).values);grid title(['CCT=',num2str(CT)]
1897 % unstable
1898     f=1;sss23=size(s23.time,1);
1899     Vitesse_unstable(1:sss23,nmach*(f-1)+1:nmach*f)=s23.signals(2).values(1:sss23,1:10);
1900     status_unstable(1:sss23,(f-1)+1:f)=s23.signals(3).values(1:sss23,1);
1901     Time_unstable(1:sss23,(f-1)+1:f)=s23.time(1:sss23,1); CCT_unstable=CT2;
1902     fname = strcat('Data','_',char(txt{i,12}));
1903     unstable=s23;stable=s2323;
1904     save(fname,'unstable','stable','CCT_stable','CCT_unstable');
1905     save(['.Line23.mat','Vitesse_unstable','Vitesse_stable','status_stable','status_unstable',
1906 'Time_unstable','Time_stable','CCT_stable','CCT_unstable'];%%L24
1907     clear Vitesse_unstable; clear Vitesse_stable; clear status_stable; clear status_unstable;
1908     clear Time_unstable, clear Time_stable; clear CCT_stable; clear CCT_unstable;
1909
1909     case 25
1910 %% Line 24
1911     Dncy=1/2; stab=0; e=Dncy; nmach=10;
1912     mdl='NE39bus2_PQ'; load_system(mdl);
1913     [~,txt,~] = xlsread('CB names.xlsx');
1914     while stab==0
1915     CT2=2*e-1; CT1=1;
1916     while (CT2-CT1>e)
1917         CT=(CT1+CT2)/2; x=1; y=CT;
1918         % Fixer le switching time du breaker premier breaker
1919         pathname = strcat(mdl,'/',char(txt{i,18}),'/',char(txt{i,13}));
1920         set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes',[x+y/60 10]);
1921         % Fixer le switching time du breaker deuxième breaker
1922         pathname = strcat(mdl,'/',char(txt{i,18}),'/',char(txt{i,14}));
1923         set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes',[x+y/60 10]);
1924         % Fixer le temps de simulation du défaut.
1925         pathname = strcat(mdl,'/',char(txt{i,18}),'/',char(txt{i,17}));
1926         set_param(pathname,'External','off','SwitchTimes',[x x+y/60]);
1927         % Simulation du défaut 2PhAB-G

```

```

1928 set_param(pathname,'FaultA',1); set_param(pathname,'FaultB',1);
1929 set_param(pathname,'FaultC',1); set_param(pathname,'GroundFault',1);
1930 LF = power_loadflow mdl; sim mdl;
1931 % Fixer à une valeur supérieure à la simulation le temps de commutation des disjoncteurs
1932 pathname = strcat mdl,'/',char(txt{i,18}),',' char(txt{i,13});
1933 set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes','11');
1934 % Fixer à une valeur supérieure à la simulation le temps de commutation des disjoncteurs
1935 pathname = strcat mdl,'/',char(txt{i,18}),',' char(txt{i,14});
1936 set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes','11');
1937 % Fixer à une valeur supérieure à la simulation le temps de commutation des défauts
1938 pathname = strcat mdl,'/',char(txt{i,18}),',' char(txt{i,17});
1939 set_param(pathname,'External','off','SwitchTimes','11');
1940 pathname = strcat mdl,'/',char(txt{i,18}),',' char(txt{i,17});
1941 set_param(pathname,'FaultA',0); set_param(pathname,'FaultB',0);
1942 set_param(pathname,'FaultC',0); set_param(pathname,'GroundFault',0);
1943 stab=max(stability);g=2.5;
1944 if stab==0
1945     CT1=CT; s24_L2424=CT1;
1946     s2424=mach;
1947     A=mach.time; w_stable= mach.signals(2).values;
1948     Vd_stable= mach.signals(5).values; Vq_stable= mach.signals(4).values;
1949     Iq_stable= mach.signals(6).values; Id_stable= mach.signals(7).values;
1950 %     figure(i); subplot(211);plot(mach.time,mach.signals(1).values);grid;subplot(212);
1951 %     plot(mach.time,mach.signals(2).values);grid title(['CCT=',num2str(CT)]);
1952 % stable
1953     if g==e
1954         g1=1;
1955         Vitesse_stable(1:601,nmach*(g1-1)+1:nmach*g1)=s2424.signals(2).values(1:601,1:10);
1956         status_stable(1:601,(g1-1)+1:g1)=s2424.signals(3).values(1:601,1);
1957         CCT_stable(1,(g1-1)+1:g1)=CT1;
1958         Time_stable(1:601,(g1-1)+1:g1)=s2424.time(1:601,1);gg=2;
1959     end
1960     if g~=e
1961         Vitesse_stable(1:601,nmach*(gg-1)+1:nmach*gg)=s2424.signals(2).values(1:601,1:10);
1962         status_stable(1:601,(gg-1)+1:gg)=s2424.signals(3).values(1:601,1);
1963         CCT_stable(1,(gg-1)+1:gg)=CT1;
1964         Time_stable(1:601,(gg-1)+1:gg)=s2424.time(1:601,1); gg=gg+1;
1965     end
1966     else
1967         CT2=CT;
1968     end
1969 end
1970     CCT=CT2;e=e+Dncy;
1971 end
1972     s24_L24=CT2; s24=mach;

```

```

1973     A1=mach.time; Id_unstable= mach.signals(7).values;
1974     w_unstable= mach.signals(2).values; Vd_unstable= mach.signals(5).values;
1975     Vq_unstable= mach.signals(4).values; Iq_unstable= mach.signals(6).values;
1976 %     figure(i+34);
1977 %     subplot(211);plot(mach.time,mach.signals(1).values);grid;subplot(212);
1978 %     plot(mach.time,mach.signals(2).values);grid title(['CCT=',num2str(CT)])
1979 % unstable
1980     f=1;sss24=size(s24.time,1);
1981     Vitesse_unstable(1:sss24,nmach*(f-1)+1:nmach*f)=s24.signals(2).values(1:sss24,1:10);
1982     status_unstable(1:sss24,(f-1)+1:f)=s24.signals(3).values(1:sss24,1);
1983     Time_unstable(1:sss24,(f-1)+1:f)=s24.time(1:sss24,1); CCT_unstable=CT2;
1984     fname = strcat('Data','_',char(txt{i,12}));
1985     unstable=s24;stable=s2424;
1986     save(fname,'unstable','stable','CCT_stable','CCT_unstable');
1987     save(['.Line24.mat','Vitesse_unstable','Vitesse_stable','status_stable','status_unstable',
1988 'Time_unstable','Time_stable','CCT_stable','CCT_unstable']);%%L25
1989 clear Vitesse_unstable; clear Vitesse_stable; clear status_stable; clear status_unstable;
1990 clear Time_unstable, clear Time_stable; clear CCT_stable; clear CCT_unstable;
1991 case 26
1992 %% Line 25
1993     Dncy=1/2; stab=0; e=Dncy; nmach=10;
1994     mdl='NE39bus2_PQ'; load_system(mdl);
1995     [~,txt,~] = xlsread('CB names.xlsx');
1996     while stab==0
1997         CT2=2*e-1; CT1=1;
1998         while (CT2-CT1>e)
1999             CT=(CT1+CT2)/2; x=1; y=CT;
2000             % Fixer le switching time du breaker premier breaker
2001             pathname = strcat(mdl,'/',char(txt{i,18}),'/',char(txt{i,13}));
2002             set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes',[x+y/60 10]);
2003             % Fixer le switching time du breaker deuxième breaker
2004             pathname = strcat(mdl,'/',char(txt{i,18}),'/',char(txt{i,14}));
2005             set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes',[x+y/60 10]);
2006             % Fixer le temps de simulation du défaut.
2007             pathname = strcat(mdl,'/',char(txt{i,18}),'/',char(txt{i,17}));
2008             set_param(pathname,'External','off','SwitchTimes',[x x+y/60]);
2009             % Simulation du défaut 2PhAB-G
2010             set_param(pathname,'FaultA',1); set_param(pathname,'FaultB',1);
2011             set_param(pathname,'FaultC',1); set_param(pathname,'GroundFault',1);
2012             LF = power_loadflow(mdl); sim(mdl);
2013             % Fixer à une valeur supérieure à la simulation le temps de commutation des disjoncteurs
2014             pathname = strcat(mdl,'/',char(txt{i,18}),'/',char(txt{i,13}));
2015             set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes','11');
2016             % Fixer à une valeur supérieure à la simulation le temps de commutation des disjoncteurs
2017             pathname = strcat(mdl,'/',char(txt{i,18}),'/',char(txt{i,14}));

```

```

2017 set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes','11');
2018 % Fixer à une valeur supérieure à la simulation le temps de commutation des défauts
2019 pathname = strcat(mdl,'/',char(txt{i,18}),','/',char(txt{i,17}));
2020 set_param(pathname,'External','off','SwitchTimes','11');
2021 pathname = strcat(mdl,'/',char(txt{i,18}),','/',char(txt{i,17}));
2022 set_param(pathname,'FaultA',0); set_param(pathname,'FaultB',0);
2023 set_param(pathname,'FaultC',0); set_param(pathname,'GroundFault',0);
2024 stab=max(stability);g=2.5;
2025 if stab==0
2026     CT1=CT; s25_L2525=CT1;
2027     s2525=mach;
2028     A=mach.time; w_stable= mach.signals(2).values;
2029     Vd_stable= mach.signals(5).values; Vq_stable= mach.signals(4).values;
2030     Iq_stable= mach.signals(6).values; Id_stable= mach.signals(7).values;
2031 %     figure(i); subplot(211);plot(mach.time,mach.signals(1).values);grid;subplot(212);
2032 %     plot(mach.time,mach.signals(2).values);grid title(['CCT=',num2str(CT)]);
2033 % stable
2034     if g==e
2035         g1=1;
2036         Vitesse_stable(1:601,nmach*(g1-1)+1:nmach*g1)=s2525.signals(2).values(1:601,1:10);
2037         status_stable(1:601,(g1-1)+1:g1)=s2525.signals(3).values(1:601,1);
2038         CCT_stable(1,(g1-1)+1:g1)=CT1;
2039         Time_stable(1:601,(g1-1)+1:g1)=s2525.time(1:601,1);gg=2;
2040     end
2041     if g~=e
2042         Vitesse_stable(1:601,nmach*(gg-1)+1:nmach*gg)=s2525.signals(2).values(1:601,1:10);
2043         status_stable(1:601,(gg-1)+1:gg)=s2525.signals(3).values(1:601,1);
2044         CCT_stable(1,(gg-1)+1:gg)=CT1;
2045         Time_stable(1:601,(gg-1)+1:gg)=s2525.time(1:601,1); gg=gg+1;
2046     end
2047     else
2048         CT2=CT;
2049     end
2050 end
2051     CCT=CT2; e=e+Dncy;
2052 end
2053     s25_L25=CT2;s25=mach;
2054     A1=mach.time; Id_unstable= mach.signals(7).values;
2055     w_unstable= mach.signals(2).values; Vd_unstable= mach.signals(5).values;
2056     Vq_unstable= mach.signals(4).values; Iq_unstable= mach.signals(6).values;
2057 % figure(i+34);
2058 %     subplot(211);plot(mach.time,mach.signals(1).values);grid;subplot(212);
2059 %     plot(mach.time,mach.signals(2).values);grid title(['CCT=',num2str(CT)]
2060 % unstable
2061     f=1;sss25=size(s25.time,1);

```



```

2062     Vitesse_unstable(1:sss25,nmach*(f-1)+1:nmach*f)=s25.signals(2).values(1:sss25,1:10);
2063     status_unstable(1:sss25,(f-1)+1:f)=s25.signals(3).values(1:sss25,1);
2064     Time_unstable(1:sss25,(f-1)+1:f)=s25.time(1:sss25,1); CCT_unstable=CCT2;
2065     fname = strcat('Data','_',char(txt{i,12}));
2066     unstable=s25;stable=s2525;
2067     save(fname,'unstable','stable','CCT_stable','CCT_unstable');
2068     save('.\Line25.mat','Vitesse_unstable','Vitesse_stable','status_stable','status_unstable',
2069 'Time_unstable','Time_stable','CCT_stable','CCT_unstable');%%L26
    clear Vitesse_unstable; clear Vitesse_stable; clear status_stable; clear status_unstable;
2070 clear Time_unstable, clear Time_stable; clear CCT_stable; clear CCT_unstable;
2071
2072     case 27
2073 %% Line 26
2074     Dncy=1/2; stab=0; e=Dncy; nmach=10;
2075     mdl='NE39bus2_PQ'; load_system(mdl);
2076     [~,txt,~] = xlsread('CB names.xlsx');
2077     while stab==0
2078     CT2=2*e-1; CT1=1;
2079     while (CT2-CT1>e)
2080         CT=(CT1+CT2)/2; x=1; y=CT;
2081         % Fixer le switching time du breaker premier breaker
2082         pathname = strcat(mdl,'/',char(txt{i,18}),'/',char(txt{i,13}));
2083         set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes',[x+y/60 10]);
2084         % Fixer le switching time du breaker deuxième breaker
2085         pathname = strcat(mdl,'/',char(txt{i,18}),'/',char(txt{i,14}));
2086         set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes',[x+y/60 10]);
2087         % Fixer le temps de simulation du défaut.
2088         pathname = strcat(mdl,'/',char(txt{i,18}),'/',char(txt{i,17}));
2089         set_param(pathname,'External','off','SwitchTimes',[x x+y/60]);
2090         % Simulation du défaut 2PhAB-G
2091         set_param(pathname,'FaultA',1); set_param(pathname,'FaultB',1);
2092         set_param(pathname,'FaultC',1); set_param(pathname,'GroundFault',1);
2093         LF = power_loadflow(mdl); sim(mdl);
2094         % Fixer à une valeur supérieure à la simulation le temps de commutation des disjoncteurs
2095         pathname = strcat(mdl,'/',char(txt{i,18}),'/',char(txt{i,13}));
2096         set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes','11');
2097         % Fixer à une valeur supérieure à la simulation le temps de commutation des disjoncteurs
2098         pathname = strcat(mdl,'/',char(txt{i,18}),'/',char(txt{i,14}));
2099         set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes','11');
2100         % Fixer à une valeur supérieure à la simulation le temps de commutation des défauts
2101         pathname = strcat(mdl,'/',char(txt{i,18}),'/',char(txt{i,17}));
2102         set_param(pathname,'External','off','SwitchTimes','11');
2103         pathname = strcat(mdl,'/',char(txt{i,18}),'/',char(txt{i,17}));
2104         set_param(pathname,'FaultA',0); set_param(pathname,'FaultB',0);
2105         set_param(pathname,'FaultC',0); set_param(pathname,'GroundFault',0);

```

```

2106     stab=max(stability);g=2.5;
2107     if stab==0
2108         CT1=CT; s26_L2626=CT1;
2109         s2626=mach;
2110         A=mach.time; w_stable= mach.signals(2).values;
2111         Vd_stable= mach.signals(5).values; Vq_stable= mach.signals(4).values;
2112         Iq_stable= mach.signals(6).values; Id_stable= mach.signals(7).values;
2113     %     figure(i); subplot(211);plot(mach.time,mach.signals(1).values);grid;subplot(212);
2114     %     plot(mach.time,mach.signals(2).values);grid title(['CCT=',num2str(CT)]);
2115     % stable
2116     if g==e
2117         g1=1;
2118         Vitesse_stable(1:601,nmach*(g1-1)+1:nmach*g1)=s2626.signals(2).values(1:601,1:10);
2119         status_stable(1:601,(g1-1)+1:g1)=s2626.signals(3).values(1:601,1);
2120         CCT_stable(1,(g1-1)+1:g1)=CT1;
2121         Time_stable(1:601,(g1-1)+1:g1)=s2626.time(1:601,1);gg=2;
2122     end
2123     if g~=e
2124         Vitesse_stable(1:601,nmach*(gg-1)+1:nmach*gg)=s2626.signals(2).values(1:601,1:10);
2125         status_stable(1:601,(gg-1)+1:gg)=s2626.signals(3).values(1:601,1);
2126         CCT_stable(1,(gg-1)+1:gg)=CT1;
2127         Time_stable(1:601,(gg-1)+1:gg)=s2626.time(1:601,1); gg=gg+1;
2128     end
2129     else
2130         CT2=CT;
2131     end
2132 end
2133     CCT=CT2; e=e+Dncy;
2134 end
2135     s26_L26=CT2; s26=mach;
2136     A1=mach.time; Id_unstable= mach.signals(7).values;
2137     w_unstable= mach.signals(2).values; Vd_unstable= mach.signals(5).values;
2138     Vq_unstable= mach.signals(4).values; Iq_unstable= mach.signals(6).values;
2139     %     figure(i+34);
2140     %     subplot(211);plot(mach.time,mach.signals(1).values);grid;subplot(212);
2141     %     plot(mach.time,mach.signals(2).values);grid
2142     %     title(['CCT=',num2str(CT)])
2143     % unstable
2144     f=1;sss26=size(s26.time,1);
2145     Vitesse_unstable(1:sss26,nmach*(f-1)+1:nmach*f)=s26.signals(2).values(1:sss26,1:10);
2146     status_unstable(1:sss26,(f-1)+1:f)=s26.signals(3).values(1:sss26,1);
2147     Time_unstable(1:sss26,(f-1)+1:f)=s26.time(1:sss26,1); CCT_unstable=CT2;
2148     fname = strcat('Data','_',char(txt{i,12}));
2149     unstable=s26;stable=s2626;
2150     save(fname,'unstable','stable','CCT_stable','CCT_unstable');

```

```

2150     save('\Line26.mat','Vitesse_unstable','Vitesse_stable','status_stable','status_unstable',
2151 'Time_unstable','Time_stable','CCT_stable','CCT_unstable'); %% L27
      clear Vitesse_unstable; clear Vitesse_stable; clear status_stable; clear status_unstable;
2152 clear Time_unstable, clear Time_stable; clear CCT_stable; clear CCT_unstable;
2153     case 28
2154 %% Line 27
2155     Dncy=1/2; stab=0; e=Dncy; nmach=10;
2156     mdl='NE39bus2_PQ'; load_system(mdl);
2157     [~,txt,~] = xlsread('CB names.xlsx');
2158 while stab==0
2159 CT2=2*e-1; CT1=1;
2160 while (CT2-CT1>e)
2161     CT=(CT1+CT2)/2; x=1; y=CT;
2162     % Fixer le switching time du breaker premier breaker
2163     pathname = strcat(mdl,'/',char(txt{i,18}),','/,char(txt{i,13}));
2164     set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes',[x+y/60 10]);
2165     % Fixer le switching time du breaker deuxième breaker
2166     pathname = strcat(mdl,'/',char(txt{i,18}),','/,char(txt{i,14}));
2167     set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes',[x+y/60 10]);
2168     % Fixer le temps de simulation du défaut.
2169     pathname = strcat(mdl,'/',char(txt{i,18}),','/,char(txt{i,17}));
2170     set_param(pathname,'External','off','SwitchTimes',[x x+y/60]);
2171     % Simulation du défaut 2PhAB-G
2172     set_param(pathname,'FaultA',1); set_param(pathname,'FaultB',1);
2173     set_param(pathname,'FaultC',1); set_param(pathname,'GroundFault',1);
2174     LF = power_loadflow(mdl); sim(mdl);
2175     % Fixer à une valeur supérieure à la simulation le temps de commutation des disjoncteurs
2176     pathname = strcat(mdl,'/',char(txt{i,18}),','/,char(txt{i,13}));
2177     set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes','11');
2178     % Fixer à une valeur supérieure à la simulation le temps de commutation des disjoncteurs
2179     pathname = strcat(mdl,'/',char(txt{i,18}),','/,char(txt{i,14}));
2180     set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes','11');
2181     % Fixer à une valeur supérieure à la simulation le temps de commutation des défauts
2182     pathname = strcat(mdl,'/',char(txt{i,18}),','/,char(txt{i,17}));
2183     set_param(pathname,'External','off','SwitchTimes','11');
2184     pathname = strcat(mdl,'/',char(txt{i,18}),','/,char(txt{i,17}));
2185     set_param(pathname,'FaultA',0); set_param(pathname,'FaultB',0);
2186     set_param(pathname,'FaultC',0); set_param(pathname,'GroundFault',0);
2187     stab=max(stability);g=2.5;
2188 if stab==0
2189     CT1=CT; s27_L2727=CT1;
2190     s2727=mach;
2191     A=mach.time; w_stable= mach.signals(2).values;
2192     Vd_stable= mach.signals(5).values; Vq_stable= mach.signals(4).values;
2193     Iq_stable= mach.signals(6).values; Id_stable= mach.signals(7).values;

```

```

% figure(i); subplot(211);plot(mach.time,mach.signals(1).values);grid;subplot(212);
2194 plot(mach.time,mach.signals(2).values);grid
2195 % title(['CCT=',num2str(CT)]);
2196 % stable
2197     if g==e
2198         g1=1;
2199         Vitesse_stable(1:601,nmach*(g1-1)+1:nmach*g1)=s2727.signals(2).values(1:601,1:10);
2200         status_stable(1:601,(g1-1)+1:g1)=s2727.signals(3).values(1:601,1);
2201         CCT_stable(1,(g1-1)+1:g1)=CT1;
2202         Time_stable(1:601,(g1-1)+1:g1)=s2727.time(1:601,1);gg=2;
2203     end
2204     if g~=e
2205         Vitesse_stable(1:601,nmach*(gg-1)+1:nmach*gg)=s2727.signals(2).values(1:601,1:10);
2206         status_stable(1:601,(gg-1)+1:gg)=s2727.signals(3).values(1:601,1);
2207         CCT_stable(1,(gg-1)+1:gg)=CT1;
2208         Time_stable(1:601,(gg-1)+1:gg)=s2727.time(1:601,1); gg=gg+1;
2209     end
2210     else
2211         CT2=CT;
2212     end
2213 end
2214     CCT=CT2; e=e+Dncy;
2215 end
2216     s27_L27=CT2; s27=mach;
2217     A1=mach.time; Id_unstable= mach.signals(7).values;
2218     w_unstable= mach.signals(2).values; Vd_unstable= mach.signals(5).values;
2219     Vq_unstable= mach.signals(4).values; Iq_unstable= mach.signals(6).values;
2220 % figure(i+34);
% subplot(211);plot(mach.time,mach.signals(1).values);grid;subplot(212);
2221 plot(mach.time,mach.signals(2).values);grid
2222 % title(['CCT=',num2str(CT)]
2223 % unstable
2224     f=1; sss27=size(s27.time,1);
2225     Vitesse_unstable(1:sss27,nmach*(f-1)+1:nmach*f)=s27.signals(2).values(1:sss27,1:10);
2226     status_unstable(1:sss27,(f-1)+1:f)=s27.signals(3).values(1:sss27,1);
2227     Time_unstable(1:sss27,(f-1)+1:f)=s27.time(1:sss27,1); CCT_unstable=CT2;
2228     fname = strcat('Data','_',char(txt{i,12}));
2229     unstable=s27;stable=s2727;
2230     save(fname,'unstable','stable','CCT_stable','CCT_unstable');
2231     save(['.Line27.mat','Vitesse_unstable','Vitesse_stable','status_stable','status_unstable',
2232 'Time_unstable','Time_stable','CCT_stable','CCT_unstable'];%%L28
clear Vitesse_unstable; clear Vitesse_stable; clear status_stable; clear status_unstable;
2233 clear Time_unstable, clear Time_stable; clear CCT_stable; clear CCT_unstable;
2234
2235     case 29
2236 %% Line 28

```

```

2237 Dncy=1/2; stab=0; e=Dncy; nmach=10;
2238 mdl='NE39bus2_PQ'; load_system(mdl);
2239 [~,txt,~] = xlsread('CB names.xlsx');
2240 while stab==0
2241 CT2=2*e-1; CT1=1;
2242 while (CT2-CT1>e)
2243     CT=(CT1+CT2)/2; x=1; y=CT;
2244     % Fixer le switching time du breaker premier breaker
2245     pathname = strcat(mdl,'/',char(txt{i,18}),',' char(txt{i,13}));
2246     set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes',[x+y/60 10]);
2247     % Fixer le switching time du breaker deuxième breaker
2248     pathname = strcat(mdl,'/',char(txt{i,18}),',' char(txt{i,14}));
2249     set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes',[x+y/60 10]);
2250     % Fixer le temps de simulation du défaut.
2251     pathname = strcat(mdl,'/',char(txt{i,18}),',' char(txt{i,17}));
2252     set_param(pathname,'External','off','SwitchTimes',[x x+y/60]);
2253     % Simulation du défaut 2PhAB-G
2254     set_param(pathname,'FaultA',1); set_param(pathname,'FaultB',1);
2255     set_param(pathname,'FaultC',1); set_param(pathname,'GroundFault',1);
2256     LF = power_loadflow(mdl); sim(mdl);
2257     % Fixer à une valeur supérieure à la simulation le temps de commutation des disjoncteurs
2258     pathname = strcat(mdl,'/',char(txt{i,18}),',' char(txt{i,13}));
2259     set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes','11');
2260     % Fixer à une valeur supérieure à la simulation le temps de commutation des disjoncteurs
2261     pathname = strcat(mdl,'/',char(txt{i,18}),',' char(txt{i,14}));
2262     set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes','11');
2263     % Fixer à une valeur supérieure à la simulation le temps de commutation des défauts
2264     pathname = strcat(mdl,'/',char(txt{i,18}),',' char(txt{i,17}));
2265     set_param(pathname,'External','off','SwitchTimes','11');
2266     pathname = strcat(mdl,'/',char(txt{i,18}),',' char(txt{i,17}));
2267     set_param(pathname,'FaultA',0); set_param(pathname,'FaultB',0);
2268     set_param(pathname,'FaultC',0); set_param(pathname,'GroundFault',0);
2269     stab=max(stability);g=2.5;
2270 if stab==0
2271     CT1=CT; s28_L2828=CT1;
2272     s2828=mach;
2273     A=mach.time; w_stable= mach.signals(2).values;
2274     Vd_stable= mach.signals(5).values; Vq_stable= mach.signals(4).values;
2275     Iq_stable= mach.signals(6).values; Id_stable= mach.signals(7).values;
    %     figure(i); subplot(211);plot(mach.time,mach.signals(1).values);grid;subplot(212);
2276 plot(mach.time,mach.signals(2).values);grid
2277 %     title(['CCT=',num2str(CT)]);
2278 % stable
2279     if g==e
2280         g1=1;

```

```

2281     Vitesse_stable(1:601,nmach*(g1-1)+1:nmach*g1)=s2828.signals(2).values(1:601,1:10);
2282     status_stable(1:601,(g1-1)+1:g1)=s2828.signals(3).values(1:601,1);
2283     CCT_stable(1,(g1-1)+1:g1)=CT1;
2284     Time_stable(1:601,(g1-1)+1:g1)=s2828.time(1:601,1);gg=2;
2285     end
2286     if g~=e
2287         Vitesse_stable(1:601,nmach*(gg-1)+1:nmach*gg)=s2828.signals(2).values(1:601,1:10);
2288         status_stable(1:601,(gg-1)+1:gg)=s2828.signals(3).values(1:601,1);
2289         CCT_stable(1,(gg-1)+1:gg)=CT1;
2290         Time_stable(1:601,(gg-1)+1:gg)=s2828.time(1:601,1); gg=gg+1;
2291     end
2292     else
2293         CT2=CT;
2294
2295     end
2296 end
2297     CCT=CT2; e=e+Dncy;
2298 end
2299     s28_L28=CT2; s28=mach;
2300     A1=mach.time; Id_unstable= mach.signals(7).values;
2301     w_unstable= mach.signals(2).values; Vd_unstable= mach.signals(5).values;
2302     Vq_unstable= mach.signals(4).values; Iq_unstable= mach.signals(6).values;
2303 %     figure(i+34);
2304 %     subplot(211);plot(mach.time,mach.signals(1).values);grid;subplot(212);
2305 plot(mach.time,mach.signals(2).values);grid
2306 %     title(['CCT=',num2str(CT)])
2307 % unstable
2308     f=1;sss28=size(s28.time,1);
2309     Vitesse_unstable(1:sss28,nmach*(f-1)+1:nmach*f)=s28.signals(2).values(1:sss28,1:10);
2310     status_unstable(1:sss28,(f-1)+1:f)=s28.signals(3).values(1:sss28,1);
2311     Time_unstable(1:sss28,(f-1)+1:f)=s28.time(1:sss28,1); CCT_unstable=CT2;
2312     fname = strcat('Data','_',char(txt{i,12}));
2313     unstable=s28;stable=s2828;
2314     save(fname,'unstable','stable','CCT_stable','CCT_unstable');
2315     save('.\Line28.mat','Vitesse_unstable','Vitesse_stable','status_stable','status_unstable',
2316     'Time_unstable','Time_stable','CCT_stable','CCT_unstable');%%L29
2317     clear Time_unstable; clear Time_stable; clear CCT_stable; clear CCT_unstable;
2318     clear Vitesse_unstable; clear Vitesse_stable;
2319
2320     case 30
2321     %% Line 29
2322     Dncy=1/2; stab=0; e=Dncy; nmach=10;
2323     mdl='NE39bus2_PQ'; load_system(mdl);
2324     [~,txt,~] = xlsread('CB names.xlsx');
2325     while stab==0

```

```

2325 CT2=2*e-1; CT1=1;
2326 while (CT2-CT1>e)
2327     CT=(CT1+CT2)/2; x=1; y=CT;
2328     % Fixer le switching time du breaker premier breaker
2329     pathname = strcat(mdl, '/', char(txt{i,18}), '/', char(txt{i,13}));
2330     set_param(pathname, 'SnubberResistance', 'inf', 'External', 'off', 'SwitchTimes', [x+y/60 10]);
2331     % Fixer le switching time du breaker deuxième breaker
2332     pathname = strcat(mdl, '/', char(txt{i,18}), '/', char(txt{i,14}));
2333     set_param(pathname, 'SnubberResistance', 'inf', 'External', 'off', 'SwitchTimes', [x+y/60 10]);
2334     % Fixer le temps de simulation du défaut.
2335     pathname = strcat(mdl, '/', char(txt{i,18}), '/', char(txt{i,17}));
2336     set_param(pathname, 'External', 'off', 'SwitchTimes', [x x+y/60]);
2337     % Simulation du défaut 2PhAB-G
2338     set_param(pathname, 'FaultA', 1); set_param(pathname, 'FaultB', 1);
2339     set_param(pathname, 'FaultC', 1); set_param(pathname, 'GroundFault', 1);
2340     LF = power_loadflow(mdl); sim(mdl);
2341     % Fixer à une valeur supérieure à la simulation le temps de commutation des disjoncteurs
2342     pathname = strcat(mdl, '/', char(txt{i,18}), '/', char(txt{i,13}));
2343     set_param(pathname, 'SnubberResistance', 'inf', 'External', 'off', 'SwitchTimes', '11');
2344     % Fixer à une valeur supérieure à la simulation le temps de commutation des disjoncteurs
2345     pathname = strcat(mdl, '/', char(txt{i,18}), '/', char(txt{i,14}));
2346     set_param(pathname, 'SnubberResistance', 'inf', 'External', 'off', 'SwitchTimes', '11');
2347     % Fixer à une valeur supérieure à la simulation le temps de commutation des défauts
2348     pathname = strcat(mdl, '/', char(txt{i,18}), '/', char(txt{i,17}));
2349     set_param(pathname, 'External', 'off', 'SwitchTimes', '11');
2350     pathname = strcat(mdl, '/', char(txt{i,18}), '/', char(txt{i,17}));
2351     set_param(pathname, 'FaultA', 0); set_param(pathname, 'FaultB', 0);
2352     set_param(pathname, 'FaultC', 0); set_param(pathname, 'GroundFault', 0);
2353     stab=max(stability);g=2.5;
2354     if stab==0
2355         CT1=CT; s29_L2929=CT1;
2356         s2929=mach;
2357         A=mach.time; w_stable= mach.signals(2).values;
2358         Vd_stable= mach.signals(5).values; Vq_stable= mach.signals(4).values;
2359         Iq_stable= mach.signals(6).values; Id_stable= mach.signals(7).values;
2360     % figure(i);subplot(211);plot(mach.time,mach.signals(1).values);grid;subplot(212);
2361     plot(mach.time,mach.signals(2).values);grid
2362     % title(['CCT=',num2str(CT)]);
2363     % stable
2364     if g==e
2365         g1=1;
2366         Vitesse_stable(1:601,nmach*(g1-1)+1:nmach*g1)=s2929.signals(2).values(1:601,1:10);
2367         status_stable(1:601,(g1-1)+1:g1)=s2929.signals(3).values(1:601,1);
2368         CCT_stable(1,(g1-1)+1:g1)=CT1;
2369         Time_stable(1:601,(g1-1)+1:g1)=s2929.time(1:601,1);gg=2;

```

```

2369     end
2370     if g~=e
2371         Vitesse_stable(1:601,nmach*(gg-1)+1:nmach*gg)=s2929.signals(2).values(1:601,1:10);
2372         status_stable(1:601,(gg-1)+1:gg)=s2929.signals(3).values(1:601,1);
2373         CCT_stable(1,(gg-1)+1:gg)=CT1;
2374         Time_stable(1:601,(gg-1)+1:gg)=s2929.time(1:601,1); gg=gg+1;
2375     end
2376     else
2377         CT2=CT;
2378     end
2379 end
2380     CCT=CT2; e=e+Dncy;
2381 end
2382     s29_L29=CT2;s29=mach;
2383     A1=mach.time; Id_unstable= mach.signals(7).values;
2384     w_unstable= mach.signals(2).values; Vd_unstable= mach.signals(5).values;
2385     Vq_unstable= mach.signals(4).values; Iq_unstable= mach.signals(6).values;
2386 %     figure(i+34);
2387 %     subplot(211);plot(mach.time,mach.signals(1).values);grid;subplot(212);
2387 plot(mach.time,mach.signals(2).values);grid
2388 %     title(['CCT=',num2str(CT)])
2389     % unstable
2390     f=1;sss29=size(s29.time,1);
2391     Vitesse_unstable(1:sss29,nmach*(f-1)+1:nmach*f)=s29.signals(2).values(1:sss29,1:10);
2392     status_unstable(1:sss29,(f-1)+1:f)=s29.signals(3).values(1:sss29,1);
2393     Time_unstable(1:sss29,(f-1)+1:f)=s29.time(1:sss29,1); CCT_unstable=CT2;
2394     fname = strcat('Data','_',char(txt{i,12}));
2395     unstable=s29;stable=s2929;
2396     save(fname,'unstable','stable','CCT_stable','CCT_unstable');
2397     save(['Line29.mat','Vitesse_unstable','Vitesse_stable','status_stable','status_unstable',
2398 'Time_unstable','Time_stable','CCT_stable','CCT_unstable'];%%L29
2399     clear Vitesse_unstable; clear Vitesse_stable; clear status_stable; clear status_unstable;
2399     clear Time_unstable, clear Time_stable; clear CCT_stable; clear CCT_unstable;
2400     case 31
2401 %% Line 30
2402     Dncy=1/2; stab=0; e=Dncy; nmach=10;
2403     mdl='NE39bus2_PQ'; load_system(mdl);
2404     [~,txt,~] = xlsread('CB names.xlsx');
2405     while stab==0
2406     CT2=2*e-1; CT1=1;
2407     while (CT2-CT1>e)
2408         CT=(CT1+CT2)/2; x=1; y=CT;
2409         % Fixer le switching time du breaker premier breaker
2410         pathname = strcat(mdl,'/',char(txt{i,18}),'/',char(txt{i,13}));
2411         set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes',[x+y/60 10]);

```



```

2412 % Fixer le switching time du breaker deuxième breaker
2413 pathname = strcat(mdl,'/',char(txt{i,18}),'/',char(txt{i,14}));
2414 set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes',[x+y/60 10]);
2415 % Fixer le temps de simulation du défaut.
2416 pathname = strcat(mdl,'/',char(txt{i,18}),'/',char(txt{i,17}));
2417 set_param(pathname,'External','off','SwitchTimes',[x x+y/60]);
2418 % Simulation du défaut 2PhAB-G
2419 set_param(pathname,'FaultA',1); set_param(pathname,'FaultB',1);
2420 set_param(pathname,'FaultC',1); set_param(pathname,'GroundFault',1);
2421 LF = power_loadflow(mdl); sim(mdl);
2422 % Fixer à une valeur supérieure à la simulation le temps de commutation des disjoncteurs
2423 pathname = strcat(mdl,'/',char(txt{i,18}),'/',char(txt{i,13}));
2424 set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes','11');
2425 % Fixer à une valeur supérieure à la simulation le temps de commutation des disjoncteurs
2426 pathname = strcat(mdl,'/',char(txt{i,18}),'/',char(txt{i,14}));
2427 set_param(pathname,'SnubberResistance','inf','External','off','SwitchTimes','11');
2428 % Fixer à une valeur supérieure à la simulation le temps de commutation des défauts
2429 pathname = strcat(mdl,'/',char(txt{i,18}),'/',char(txt{i,17}));
2430 set_param(pathname,'External','off','SwitchTimes','11');
2431 pathname = strcat(mdl,'/',char(txt{i,18}),'/',char(txt{i,17}));
2432 set_param(pathname,'FaultA',0); set_param(pathname,'FaultB',0);
2433 set_param(pathname,'FaultC',0); set_param(pathname,'GroundFault',0);
2434 stab=max(stability);g=2.5;
2435 if stab==0
2436     CT1=CT; s30_L3030=CT1;
2437     s3030=mach;
2438     A=mach.time; w_stable= mach.signals(2).values;
2439     Vd_stable= mach.signals(5).values; Vq_stable= mach.signals(4).values;
2440     Iq_stable= mach.signals(6).values; Id_stable= mach.signals(7).values;
    % figure(i); subplot(211);plot(mach.time,mach.signals(1).values);grid;subplot(212);
2441 plot(mach.time,mach.signals(2).values);grid
2442 % title(['CCT=',num2str(CT)]);
2443 % stable
2444     if g==e
2445         g1=1;
2446         Vitesse_stable(1:601,nmach*(g1-1)+1:nmach*g1)=s3030.signals(2).values(1:601,1:10);
2447         status_stable(1:601,(g1-1)+1:g1)=s3030.signals(3).values(1:601,1);
2448         CCT_stable(1,(g1-1)+1:g1)=CT1;
2449         Time_stable(1:601,(g1-1)+1:g1)=s3030.time(1:601,1);gg=2;
2450     end
2451     if g~=e
2452         Vitesse_stable(1:601,nmach*(gg-1)+1:nmach*gg)=s3030.signals(2).values(1:601,1:10);
2453         status_stable(1:601,(gg-1)+1:gg)=s3030.signals(3).values(1:601,1);
2454         CCT_stable(1,(gg-1)+1:gg)=CT1;
2455         Time_stable(1:601,(gg-1)+1:gg)=s3030.time(1:601,1); gg=gg+1;

```

```

2456     end
2457     else
2458         CT2=CT;
2459     end
2460 end
2461     CCT=CT2; e=e+Dncy;
2462 end
2463     s30_L30=CT2; s30=mach;
2464     A1=mach.time; Id_unstable= mach.signals(7).values;
2465     w_unstable= mach.signals(2).values; Vd_unstable= mach.signals(5).values;
2466     Vq_unstable= mach.signals(4).values; Iq_unstable= mach.signals(6).values;
2467 %     figure(i+34);
2468 %     subplot(211);plot(mach.time,mach.signals(1).values);grid;subplot(212);
2468 plot(mach.time,mach.signals(2).values);grid
2469 %     title(['CCT=',num2str(CT)])
2470 % unstable
2471     f=1;sss30=size(s30.time,1);
2472     Vitesse_unstable(1:sss30,nmach*(f-1)+1:nmach*f)=s30.signals(2).values(1:sss30,1:10);
2473     status_unstable(1:sss30,(f-1)+1:f)=s30.signals(3).values(1:sss30,1);
2474     Time_unstable(1:sss30,(f-1)+1:f)=s30.time(1:sss30,1); CCT_unstable=CT2;
2475     fname = strcat('Data','_',char(txt{i,12}));
2476     unstable=s30;stable=s3030;
2477     save(fname,'unstable','stable','CCT_stable','CCT_unstable');
2478     save(['.Line30.mat','Vitesse_unstable','Vitesse_stable','status_stable','status_unstable',
2478 'Time_unstable','Time_stable','CCT_stable','CCT_unstable']);
2479     end
2480 end

```

A.4 Code Matlab d'extraction des indices de stabilité à haute dimension

(HDSI)

```

1 close all; clear all
2 %Test de la structure Time Series pour HCTSA
3 StabMode=1;Dic=0;
4 nameG=['W_r'];dt=.0083;fs=1/dt;
5 fault=['P3N'];
6 % for l=1:9, fault(l,1)=num2str(l);end
7 nig=[1];nflt=5510;
8 sd= dir('simul1') ;ij=1;nl=length(sd)-2;
9 for il=1:nl
10 load(['simul1\',sd(il+2).name]);
11 [ndd,mdd]=size(cas1);ng=1;nv=1;nsg=nv*ng;iflt=1;ig=1;

```

```

12 stab_ok=max(cas2);
13 dt=.0083;fs=1/dt;nx=[1:75]; nd=length(nx);% figure(100);plot(nx*dt,cas1(nx,1));grid;axis tight;
14 dictionary = {'RnIdent','dct','sin'};[mpdict,nbvect] = wmpdictionary(nd,'lstcpt',dictionary);
15 for i=1:nsg*nflt;
16     if(rem(i,1)==0)
17         if(Dic==1)
18             [yfit,r,coef,iopt,qual,X] = wmpalg('OMP',cas1(nx,i),mpdict,'itermax',16); %35 vecteurs non nuls
19             id=find(iopt<=nbvect(1));
20             ic=find(iopt>=nbvect(1)+1 & iopt<=nbvect(1)+nbvect(2));
21             is=find(iopt>=nbvect(1)+nbvect(2)+1 & iopt<=sum(nbvect));
22             %Express signal as a sparse sum of Gabors
23             Xd=zeros(nd,1);Xc=Xd;Xs=Xd;Xa=Xd;jopt=[id ic is];
24             for l=1:length(id), Xd=Xd+coef(id(l))*X(:,iopt(id(l)));end
25             for l=1:length(ic), Xc=Xc+coef(ic(l))*X(:,iopt(ic(l)));end
26             for l=1:length(is), Xs=Xs+coef(is(l))*X(:,iopt(is(l)));end
27             Xa=Xc+Xs;
28             timeSeriesData{ij}=[Xc; Xs; Xd]; %cas1(nx,i)';
29         else
30             timeSeriesData{ij}=cas1(nx,i)';
31         end
32         labels{ij}=['Line',num2str(il),'G',num2str(ig),'',nameG(nv,:),'.fault',num2str(iflt)];
33         % keywords{ij} = ['fault',fault(iflt,:)];
34         if(StabMode==0) %Classification du type de défaut en 4 groupes
35             keywords{ij} = ['fault',fault(iflt,:)];
36         else %Classification de la stabilité en 2 groupes
37             keywords{ij} = ['stab',num2str(stab_ok(iflt))];
38         end
39         ij=ij+1;ig=ig+1;if(rem(i,nsg))==0, iflt=iflt+1; end;
40     end;
41     if(ig==17),ig=1;end;
42 end
43 end%% %% %% %% %% %% %% %% id
44 save('INP_test_stab.mat','timeSeriesData','labels','keywords');
45 TS_init('INP_test_stab.mat','INP_mops.txt','INP_ops_ik.txt');
46 TS_LabelGroups('raw',[]);
47 % Compute all missing values in HCTSA.mat:
48 TS_compute(true);
49 distanceMetricRow = 'euclidean'; % time-series feature distance
50 linkageMethodRow = 'average'; % linkage method
51 distanceMetricCol = 'corr_fast'; % a (poor) approximation of correlations with NaNs
52 linkageMethodCol = 'average'; % linkage method
53 TS_normalize();
54 TS_cluster(distanceMetricRow, linkageMethodRow, distanceMetricCol, linkageMethodCol);
55 TS_plot_DataMatrix('norm')
56 TS_plot_DataMatrix('cl')

```

```

57 TS_plot_DataMatrix('cl','colorGroups',true);
58 TS_PlotLowDim('norm','pca');
59 TS_TopFeatures;
60 whatData = 'norm'; % Get data from HCTSA_N.mat
61   plotWhatTimeSeries = 'all'; % plot examples from all time series
62   plotHowMany = 20; % how many to plot
63   maxLength = 75; % maximum number of samples to plot for each time series
64   TS_PlotTimeSeries(whatData,plotHowMany,plotWhatTimeSeries,maxLength);
65   % Plot as a freeform plot without labeling time series:
66   plotOptions = struct('plotFreeForm',true,'displayTitles',false);
67   TS_PlotTimeSeries('norm',40,'all',75,plotOptions);
68   numPerGroup = 5; % plot this many examples of each group of time series
69   plotHow = 'grouped'; % plot examples of each assigned group of time series
70   TS_PlotTimeSeries('norm',numPerGroup,plotHow,75);
71 t = cputime;
72 TS_classify('norm');
73 time_features = cputime-t;
74 load('HCTSA_N.mat')
75 T = array2table(TS_DataMat,'VariableNames',Operations.Name,'RowNames',TimeSeries.Name);
76 T.Group=TimeSeries.Keywords;
77 load('HCTSA_Nf.mat', 'TimeSeries')
78 T.Fault=TimeSeries.Keywords;
79 writetable(T,'test_stab','FileType','text')
80 eval(['!copy HCTSA_Nf.mat HCTSA_N',num2str(Dic),num2str(StabMode),'.mat']);

```

A.5 Code Matlab d'implémentation du dictionnaire joint au classificateur pour la DSP: DK-SVD

```

1 % =====
2 % Label KSVD algorithm
3 % USAGE: [D,X,T,W]=labelconsistentksvd1(Y,Dinit,Q_train,Tinit,H_train,...
4 %           iterations,sparsitythres,sqrt_alpha)
5 % Inputs
6 %   Y           -training features
7 %   Dinit       -initialized dictionary
8 %   Q_train     -optimal code matrix for training feature
9 %   Tinit       -initialized transform matrix
10 %   H_train    -labels matrix for training feature
11 %   iterations -iterations for KSVD
12 %   sparsitythres -sparsity threshold for KSVD
13 %   sqrt_alpha -contribution factor
14 % Outputs

```

```

15 % D -learned dictionary
16 % X -sparsed codes
17 % T -learned transform matrix
18 % W -learned classifier parameters
19 % =====
20 %% KSVD dictionary learning process
21 % K-singular value decomposition: K-SVD + OMP
22 addpath(genpath('\ksvdbox')); % add K-SVD box
23 addpath(genpath('\OMPbox')); % add sparse coding algorithm OMP
24 load('\trainingdata\matlab.mat','Label','w_stab');
25 [nl,nc]=size(w_stab); X=w_stab; state=Label;
26 load mat;
27 % % Partitionner les données
28 num_points = size(X,2); split_point = round(num_points*0.7); seq = randperm(num_points);
29 Training_dataset = X(:,seq(1:split_point)); Testing_dataset = X(:,seq(split_point+1:end));
30 Label_train = state(:,seq(1:split_point)); Label_Test = state(:,seq(split_point+1:end));
31 % Fixer les parametres du dictionnaire
32 sparsitythres = 10; % sparsity prior
33 sqrt_alpha = 0.4; % weights for label constraint term
34 sqrt_beta = 0.2; % weights for classification err term
35 dictsize = size(Label_Test,2); % dictionary size
36 iterations = 50; % iteration number
37 iterations4ini = 50; % iteration number for initialization
38 disp('#####');
39 disp('-----');
40 disp('***** Power Stability Dictionary Learning *****');
41 disp('-----');
42 %% Stability BD: KSVD + OMP
43 %Initialisation du dictionnaire d'apprentissage crée à partir de la BD des signaux de stabilité
44 fprintf('\nInitialization Signal PSDL-KSVD... ');
45 [Dinit_S,Tinit_S,Winit_S,Q_train_S] = initializationLCKSVD(Training_dataset,Label_train,
46 dictsize,iterations4ini,sparsitythres);fprintf('done!');
47 [Dinit_L,Tinit_L,Winit_L,Q_train_L] = initializationLCKSVD(Label_train,Training_dataset,
48 dictsize,iterations4ini,sparsitythres);fprintf('done!');
49 t = cputime;
50 % Dictionnaire d'apprentissage crée à partir de la BD des signaux de stabilité
51 fprintf('\nTrain Signal PSDL-KSVD...');
52 [D_KSVD_S,X_KSVD_S,T_KSVD_S,W_KSVD_S] = labelconsistentksvd1
53 (Training_dataset,Dinit_S,Q_train_S,Tinit_S,Label_train,iterations,sparsitythres,sqrt_alpha);fprintf('done!');
54 time_train = cputime-t;
55 fprintf('\nTrain Label PSDL-KSVD...');
56 [D_KSVD_L,X_KSVD_L,T_KSVD_L,W_KSVD_L] = labelconsistentksvd1(Label_train,Dinit_L,
57 Q_train_L,Tinit_L,Training_dataset,iterations,sparsitythres,sqrt_alpha);fprintf('done!');
58 t1 = cputime;
59 % learning linear classifier parameters

```

```

57 [PREDICTION] = Classifier_LSCD(D_KSVD_S,D_KSVD_L,Label_Test,Testing_dataset,sparsitythres);
    save('\trainingdata\Data.mat','Label_train','Label_Test','Training_dataset','Testing_dataset',
58 'PREDICTION','D_KSVD_S','D_KSVD_L');
59 time_online = cputime-t1;

```

A.6 Code Matlab d'implémentation des dictionnaires séparés des classificateurs: DCT, DHT, DWT, K-SVD, [DHT|DWT], [DCT|DWT], [DHT|DCT], [DST|DHT|DCT], [ID|DCT|DST], [DWT|DHT|DCT] pour la DSP.

```

1  % =====
2  % An example code for Train and Test single and hybrid Dictionaries Learning.
3  % Dataset: Implemented using bisection algorithm with IEEE 39-bus
4  % Author: Raoult Teukam Dabou
5  % Power Electronics and Industrial Control Laboratory (LEEPCI), Laval University
6  % Date: 08-08-2021.
7  % =====
8  % Load Dataset
9  clear all; clc
10 load matlab; nd=size(SPEED_TRAIN,2); L= Label_TRAIN;V= SPEED_TRAIN; H=size(SPEED_TRAIN,2);
11 for isg=1:nd
12     sig_v=V(:,isg)-0*V(1,isg);ns = length(sig_v);
13     sig_l=L(:,isg)-0*L(1,isg);
14     for k=1:ns
15         if(k>nd), y=sig_v(k-nd+1:k); end
16     %% Single (1 Atom)
17     % DCT
18     [mpdict_v_dct,nbvect_v_dct] = wmpdictionary(length(SPEED_TRAIN(:,k)), 'lstcpt', {'dct'});
19     [yfit_dct,r_v_dct,coef_v_dct,iopt_v_dct,qual_v_dct,X_v_dct] =
20     wmpalg('OMP',sig_v(1:size(SPEED_TRAIN,1),1),mpdict_v_dct);
21     Xdcs=zeros(ns,1); id_v_dct=find(iopt_v_dct<=nbvect_v_dct(1));
22     for l=1:length(id_v_dct); Xdcs=Xdcs + coef_v_dct(id_v_dct(l))*X_v_dct
23     (:,iopt_v_dct(id_v_dct(l))); end
24     DICO_COEF_S_DCT(:,isg)=coef_v_dct;
25     id_v_dct1(isg,:)=id_v_dct;
26     iopt_v_dct1(isg,:)=iopt_v_dct;
27     % DWT
28     [mpdict_v_dwt,nbvect_v_dwt] = wmpdictionary(length(SPEED_TRAIN(:,k)), 'lstcpt', {'db4',4});
29     [yfit_dwt,r_v_dwt,coef_v_dwt,iopt_v_dwt,qual_v_dwt,X_v_dwt] =
30     wmpalg('OMP',sig_v(1:size(SPEED_TRAIN,1),1),mpdict_v_dwt);
31     Xdws=zeros(ns,1); id_v_dwt=find(iopt_v_dwt<=nbvect_v_dwt(1));
32     for l=1:length(id_v_dwt); Xdws=Xdws + coef_v_dwt(id_v_dwt(l))*
33     X_v_dwt(:,iopt_v_dwt(id_v_dwt(l))); end

```

```

30 DICO_COEF_S_DWT(:,isg)=coef_v_dwt;
31 % DHT
32 [mpdict_v_dht,nbvect_v_dht] = wmpdictionary(length(SPEED_TRAIN(:,k)), 'lstcpt', {'wphaar',3});
[yfit_dht,r_v_dht,coef_v_dht,iopt_v_dht,qual_v_dht,X_v_dht] =
33 wmpalg('OMP',sig_v(1:size(SPEED_TRAIN,1),1),mpdict_v_dht);
34 Xdhs=zeros(ns,1); id_v_dht=find(iopt_v_dht<=nbvect_v_dht(1));
for l=1:length(id_v_dht); Xdhs=Xdhs + coef_v_dht(id_v_dht(l))*
35 X_v_dht(:,iopt_v_dht(id_v_dht(l))); end
36 DICO_COEF_S_DHT(:,isg)=coef_v_dht;
37 %% Hybride (2 Atoms)
38 % DHT+DCT
[mpdict_v_dctdht,nbvect_v_dctdht] = wmpdictionary(length(SPEED_TRAIN(:,k)),
39 'lstcpt',{'wphaar',4},'dct');
[yfit_dctdht,r_v_dctdht,coef_v_dctdht,iopt_v_dctdht,qual_v_dctdht,X_v_dctdht] =
40 wmpalg('OMP',sig_v(1:size(SPEED_TRAIN,1),1),mpdict_v_dctdht);
41 Xdchs=zeros(ns,1); id_v_dctdht=find(iopt_v_dctdht<=nbvect_v_dctdht(1));
for l=1:length(id_v_dctdht); Xdchs=Xdchs +
42 coef_v_dctdht(id_v_dctdht(l))*X_v_dctdht(:,iopt_v_dctdht(id_v_dctdht(l))); end
43 DICO_COEF_S_DCTDHT(:,isg)=coef_v_dctdht;
44 % DWT+DCT
[mpdict_v_dctdwt,nbvect_v_dctdwt] = wmpdictionary(length(SPEED_TRAIN(:,k)),
45 'lstcpt',{'db4',4},'dct');
[yfit_dctdwt,r_v_dctdwt,coef_v_dctdwt,iopt_v_dctdwt,qual_v_dctdwt,X_v_dctdwt] =
46 wmpalg('OMP',sig_v(1:size(SPEED_TRAIN,1),1),mpdict_v_dctdwt);
47 Xdcws=zeros(ns,1); id_v_dctdwt=find(iopt_v_dctdwt<=nbvect_v_dctdwt(1));
for l=1:length(id_v_dctdwt); Xdcws=Xdcws +
48 coef_v_dctdwt(id_v_dctdwt(l))*X_v_dctdwt(:,iopt_v_dctdwt(id_v_dctdwt(l))); end
49 DICO_COEF_S_DCTDWT(:,isg)=coef_v_dctdwt;
50 % DWT+DST
[mpdict_v_dwtdst,nbvect_v_dwtdst] = wmpdictionary(length(SPEED_TRAIN(:,k)),
51 'lstcpt',{'sin',{'db4',4}});
% [mpdict_v_dwt,dht,nbvect_v_dwt] = wmpdictionary(length(ROCOTE(:,k)),
52 'lstcpt',{'wphaar',4},{'db4',4});
[yfit_dwtdst,r_v_dwtdst,coef_v_dwtdst,iopt_v_dwtdst,qual_v_dwtdst,X_v_dwtdst] =
53 wmpalg('OMP',sig_v(1:size(SPEED_TRAIN,1),1),mpdict_v_dwtdst);
54 Xdwss=zeros(ns,1); id_v_dwtdst=find(iopt_v_dwtdst<=nbvect_v_dwtdst(1));
for l=1:length(id_v_dwtdst); Xdwss=Xdwss + coef_v_dwtdst(id_v_dwtdst(l))*
55 X_v_dwtdst(:,iopt_v_dwtdst(id_v_dwtdst(l))); end
56 DICO_COEF_S_DWTDST(:,isg)=coef_v_dwtdst;
57 %% Hybride (3 Atoms)
58 % DI+DCT+DST
[mpdict_v_didctdst,nbvect_v_didctdst] = wmpdictionary(length(SPEED_TRAIN(:,k)),
59 'lstcpt',{'RnIdent','dct','sin'});
[yfit_didctdst,r_v_didctdst,coef_v_didctdst,iopt_v_didctdst,qual_v_didctdst,X_v_didctdst] =
60 wmpalg('OMP',sig_v(1:size(SPEED_TRAIN,1),1),mpdict_v_didctdst);
61 Xdicss=zeros(ns,1); id_v_didctdst=find(iopt_v_didctdst<=nbvect_v_didctdst(1));
for l=1:length(id_v_didctdst); Xdicss=Xdicss + coef_v_didctdst(id_v_didctdst(l))*
62 X_v_dwtdst(:,iopt_v_didctdst(id_v_didctdst(l))); end
63 DICO_COEF_S_DIDCTDST(:,isg)=coef_v_didctdst;
64 % DCT+DHT+DWT

```

```

[mpdict_v_dctdhtdwt,nbvect_v_dctdhtdwt] =
65 wmpdictionary(length(SPEED_TRAIN(:,k)), 'lstept', {'dct', {'wphaar',4}, {'db4',4}});
[yfit_dctdhtdwt,r_v_dctdhtdwt,coef_v_dctdhtdwt,iopt_v_dctdhtdwt,
qual_v_dctdhtdwt,X_v_dctdhtdwt] = wmpalg('OMP',sig_v(1:size(SPEED_TRAIN,1),1),
66 mpdict_v_dctdhtdwt);
67 Xdchws=zeros(ns,1); id_v_dctdhtdwt=find(iopt_v_dctdhtdwt<=nbvect_v_dctdhtdwt(1));
for l=1:length(id_v_dctdhtdwt); Xdchws=Xdchws + coef_v_dctdhtdwt(id_v_dctdhtdwt(l))*
68 X_v_dctdhtdwt(:,iopt_v_dctdhtdwt(id_v_dctdhtdwt(l))); end
69 DICO_COEF_S_DCTDHTDWT(:,isg)=coef_v_dctdhtdwt;
SP_X_L_DCTDHTDWT(:,size(X_1_dctdhtdwt,2))*
70 (isg-1)+1:size(X_1_dctdhtdwt,2)*isg=X_1_dctdhtdwt;
71 % DST+DHT+DWT
[mpdict_v_dstdhtdwt,nbvect_v_dstdhtdwt] =
72 wmpdictionary(length(SPEED_TRAIN(:,k)), 'lstept', {'sin', {'wphaar',4}, {'db4',4}});
[yfit_dstdhtdwt,r_v_dstdhtdwt,coef_v_dstdhtdwt,iopt_v_dstdhtdwt,
qual_v_dstdhtdwt,X_v_dstdhtdwt] = wmpalg('OMP',
73 sig_v(1:size(SPEED_TRAIN,1),1),mpdict_v_dctdhtdwt);
74 Xdshws=zeros(ns,1); id_v_dstdhtdwt=find(iopt_v_dstdhtdwt<=nbvect_v_dstdhtdwt(1));
for l=1:length(id_v_dstdhtdwt); Xdshws=Xdshws + coef_v_dstdhtdwt(id_v_dstdhtdwt(l))*
75 X_v_dstdhtdwt(:,iopt_v_dstdhtdwt(id_v_dstdhtdwt(l))); end
76 DICO_COEF_S_DSTDHTDWT(:,isg)=coef_v_dstdhtdwt;
77 end
78 end
79 % erreur de reconstruction
80 ERR_dctdhtdwt=SPEED(:,isg)-yfit_dctdhtdwt;
81 ERR_ldctdhtdwt=Label(:,isg)-yfit_l_dctdhtdwt;
82 % erreur de reconstruction
83 ERR_dctdwt=SPEED(:,isg)-yfit_dctdwt;
84 ERR_ldctdwt=Label(:,isg)-yfit_l_dctdwt;
85 % erreur de reconstruction
86 ERR_dctdht=SPEED(:,isg)-yfit_dctdht;
87 ERR_ldctdht=Label(:,isg)-yfit_l_dctdht;
88 % erreur de reconstruction
89 ERR_dwt=SPEED(:,isg)-yfit_dwt;
90 ERR_ldwt=Label(:,isg)-yfit_l_dwt;
91 % erreur de reconstruction
92 ERR_dct=SPEED(:,isg)-yfit_dct; ERR_ldct=Label(:,isg)-yfit_l_dct;
93 % erreur de reconstruction
94 ERR_dht=SPEED(:,isg)-yfit_dht;
95 ERR_ldht=Label(:,isg)-yfit_l_dht;
96 % erreur de reconstruction
97 ERR_dwdht=SPEED(:,isg)-yfit_dwdht;
98 ERR_ldwdht=Label(:,isg)-yfit_l_dwdht;
99 % erreur de reconstruction
100 ERR_didctdst=SPEED(:,isg)-yfit_didctdst;
101 ERR_ldidctdst=Label(:,isg)-yfit_l_didctdst;
102 % erreur de reconstruction
103 ERR_dstdhtdwt=SPEED(:,isg)-yfit_dstdhtdwt;

```



```

104 ERR_ldstdhtdwt=Label(:,isg)-yfit_1_dstdhtdwt;
105 % KSVD
106 yfit_ksvd=PREDICTION.Signal_Pred(:,isg);
107 ERR_sksvd=PREDICTION.err_S(:,isg);
108 status_lksvd=PREDICTION.Status_Pred(:,isg);
109 ERR_lksvd=PREDICTION.err_L(:,isg);
110 figure(isg)
111 % 1 atom
112 subplot(3,4,1); plot(t,SPEED(:,isg),'b',t,yfit_dct,'g',t,yfit_dwt,'c',t,
113 yfit_dht,'m',t,yfit_ksvd,'k','linewidth',1.5); hold on;
114 legend('Orginal signal','DCT with OMP','DWT with OMP','DHT
115 with OMP','KSVD with OMP'); grid
116 title('Single (1 atom) & Adaptive ',['Sparse Decomposition #ID--',
117 num2str(isg) '(Size: 4281)'])
118 xlabel('Post-contingency Time (s)');ylabel('Rotor Speed (p.u)');
119 subplot(3,4,2); plot(t,ERR_dct,'g',t,ERR_dwt,'c',t,ERR_dht,'m',t,ERR_sksvd,
120 'k','linewidth',1.5); hold on;
121 legend('DCT with OMP','DWT with OMP','DHT with OMP','KSVD with OMP'); grid
122 title('Single (1 atom) & Adaptive ',['RMSE #ID--',num2str(isg) '(Size: 4281)'])
123 xlabel('Post-contingency Time (s)');ylabel('Reconstruction Error (p.u)');
124 subplot(3,4,3); plot(t,Label(:,isg),'b',t,yfit_1_dct,'g',t,yfit_1_dwt,'c',t,yfit_1_dht,'m',t,
125 status_lksvd,'k','linewidth',1.5);
126 legend('Orginal stability','DCT with OMP','DWT with OMP','DHT with OMP','KSVD
127 with OMP'); grid
128 title('Single (1 atom) & Adaptive',['Stability status #ID--',num2str(isg) '(Size: 4281)'])
129 xlabel('Post-contingency Time (s)');ylabel('Binary(S=0 or U=1)');
130 subplot(3,4,4); plot(t,ERR_ldct,'g',t,ERR_ldwt,'c',t,ERR_ldht,'m',t,ERR_lksvd,
131 'k','linewidth',1.5); hold on;
132 legend('DCT with OMP','DWT with OMP','DHT with OMP','KSVD with OMP'); grid
133 title('Single (1 atom) & Adaptive ',['RMSE #ID--',num2str(isg) '(Size: 4281)'])
134 xlabel('Post-contingency Time (s)');ylabel('Classification Error');
135 % 2 Atoms
136 subplot(3,4,5); plot(t,SPEED(:,isg),'b',t,yfit_dctdht,'g',t,yfit_dctdwt,'c',t,
137 yfit_dwtdht,'m',t,yfit_ksvd,'k','linewidth',1.5);
138 legend('Orginal signal','[DCT DHT] with OMP','[DCT DWT] with OMP','[DST DWT]
139 with OMP','KSVD with OMP');
140 title('Hybrid (2 atoms) & Adaptive ',['Sparse Decomposition #ID--',num2str(isg)
141 '(Size: 4281)'])
142 xlabel('Post-contingency Time (s)');ylabel('Rotor Speed (p.u)');
143 subplot(3,4,6); plot(t,ERR_dctdht,'g',t,ERR_dctdwt,'c',t,ERR_dwtdht,'m',t,ERR_sksvd,
144 'k','linewidth',1.5);
145 legend('[DCT DHT] with OMP','[DCT DWT] with OMP','[DST DWT] with OMP','KSVD
146 with OMP'); grid
147 title('Hybrid (2 atoms) & Adaptive ',['RMSE #ID--',num2str(isg) '(Size: 4281)'])
148 xlabel('Post-contingency Time (s)');ylabel('Reconstruction Error (p.u)');
149 subplot(3,4,7); plot(t,Label(:,isg),'b',t,yfit_1_dctdht,'g',t,yfit_1_dctdwt,'c',t,yfit_1_dwtdht,'m',t,
150 status_lksvd,'k','linewidth',1.5);
151 legend('Orginal stability','[DCT DHT] with OMP','[DCT DWT] with OMP','[DST DWT]
152 with OMP','KSVD with OMP');
153 title('Hybrid (2 atoms) & Adaptive',['Stability status #ID--',num2str(isg) '(Size: 4281)'])

```

```

140 xlabel('Post-contingency Time (s);ylabel('Binary(S=0 or U=1));
    subplot(3,4,8); plot(t,ERR_ldctdht,'g',t,ERR_ldctdwt,'c',t,ERR_ldwdht,'m',t,ERR_lksvd,'k',
141 linewidth',1.5); hold on;
    legend(['DCT DHT] with OMP','[DCT DWT] with OMP','[DST DWT] with OMP','KSVD
142 with OMP'); grid
143 title('Hybrid (2 atoms) & Adaptive ',['RMSE #ID--',num2str(ig) '(Size: 4281)'])
144 xlabel('Post-contingency Time (s);ylabel('Classification Error');
145 % 3 Atoms
    subplot(3,4,9); plot(t,SPEED(:,ig),'b',t,yfit_didctdst,'g',t,yfit_dctdhtdwt,'c',t,yfit_dstdhtdwt,'m',
146 t,yfit_ksvd,'k',linewidth',1.5);
    legend('Orginal signal','[DI DCT DST] with OMP','[DCT DHT DWT]
    with OMP','[DST DHT DWT]
147 with OMP','KSVD with OMP');
148 title('Hybrid (3 atoms) & Adaptive ',['Sparse Decomposition #ID--',num2str(ig) '(Size: 4281)'])
149 xlabel('Post-contingency Time (s);ylabel('Rotor Speed (p.u)');
    subplot(3,4,10); plot(t,ERR_didctdst,'g',t,ERR_dctdhtdwt,'c',t,ERR_dstdhtdwt,'m',t,
150 ERR_sksvd,'k',linewidth',1.5); hold on;
    legend(['DI DCT DST] with OMP','[DCT DHT DWT] with OMP','[DST DHT DWT]
151 with OMP','KSVD with OMP'); grid
152 title('Hybrid (3 atoms) & Adaptive ',['RMSE #ID--',num2str(ig) '(Size: 4281)'])
153 xlabel('Post-contingency Time (s);ylabel('Reconstruction Error');
    subplot(3,4,11); plot(t,Label(:,ig),'b',t,yfit_1_didctdst,'g',t,yfit_1_dctdhtdwt,'c',t,
154 yfit_1_dstdhtdwt,'m',t,status_lksvd,'k',linewidth',1.5);
    legend('Orginal stability','[DI DCT DST] with OMP','[DCT DHT DWT] with
    OMP','[DST DHT DWT]
155 with OMP','KSVD with OMP');
156 title('Hybrid (3 atoms) & Adaptive ',['Stability status #ID--',num2str(ig) '(Size: 4281)'])
157 xlabel('Post-contingency Time (s);ylabel('Binary(S=0 or U=1));
    subplot(3,4,12); plot(t,ERR_ldidctdst,'g',t,ERR_ldctdhtdwt,'c',t,ERR_ldstdhtdwt,'m',t,
158 ERR_lksvd,'k',linewidth',1.5); hold on;
    legend(['DI DCT DST] with OMP','[DCT DHT DWT] with OMP','[DST DHT DWT]
159 with OMP','KSVD with OMP'); grid
160 title('Hybrid (3 atoms) & Adaptive ',['RMSE #ID--',num2str(ig) '(Size: 4281)'])
161 xlabel('Post-contingency Time (s);ylabel('Classification Error');
162 set(gcf,'Position',[1 1 2000 2000]);pause(9); close(figure(ig))

```

A.7 Code Matlab d'implémentation de l'algorithme SVM pour la DSP

```

1 % =====
2 % An example code for SVM Learning.
3 % Dataset: Implemented using bisection algorithm with IEEE 39-bus
4 % Author: Raoult Teukam Dabou
5 % Power Electronics and Industrial Control Laboratory (LEEPCI), Laval University
6 % Date: 08-10-2021.
7 % =====
8 load matlab;
9 %% SVM

```

```

10 load matlab;t1 = cputime;
11 Train=[Training_dataset Testing_dataset];Train=Train';Train=Train(:,1);
    Label=[Label_train Label_Test];Label=Label';Label=Label(:,1); Testing_dataset=
12 Testing_dataset(1,:);Testing_dataset=Testing_dataset';
13 SVMModel = fitsvm(Train,Label);% Fit a SVM classifier model on the training samples.
14 Learning_Time = cputime-t1;
15 t2 = cputime;
    YPredicted = predict(SVMModel,Testing_dataset);%Predict labels for the test samples by
16 using the trained model.
    Label_test=Label_Test(1,:);YPredicted=YPredicted';
17 Confusion_S_V_M=plotconfusion(Label_test,YPredicted);
18 Classification_Time = cputime-t2;
19 title('Confusion_S_V_M IEEE 39 bus')
    save('\SVM.mat','Learning_Time','Classification_Time','SVMModel',
20 'Confusion_S_V_M','YPredicted');

```

A.8 Code Matlab d'implémentation de l'algorithme DT pour la DSP

```

1 % =====
2 % An example code for DT Learning.
3 % Dataset: Implemented using bisection algorithm with IEEE 39-bus
4 % Author: Raoult Teukam Dabou
5 % Power Electronics and Industrial Control Laboratory (LEEPCI), Laval University
6 % Date: 08-10-2021.
7 % =====
8 %% Decision Trees
9 load matlab;t1 = cputime;
    Train=[Training_dataset Testing_dataset];Train=Train';Train=Train(:,1);%
10 Testing_dataset1=Testing_dataset(1:507,:);
    Label=[Label_train Label_Test];Label=Label';Label=Label(:,1);
11 Testing_dataset=Testing_dataset(1,:);Testing_dataset=Testing_dataset';
    Mdl = fitctree(Train,Label,'MinLeafSize',1);% Fit a decision tree classifier
12 model on the training samples.
13 Learning_Time = cputime-t1;
14 view(Mdl,'Mode','Graph')% Fit a decision tree classifier model on the training samples.
15 t2 = cputime;
    YPredicted = predict(Mdl,Testing_dataset);%Predict labels for the test samples
16 by using the trained model.
    Label_test=Label_Test(1,:);YPredicted=YPredicted';
17 Confusion_D_T=plotconfusion(Label_test,YPredicted);
18 Classification_Time = cputime-t2;
19 title('Confusion_D_T IEEE 39 bus')
20 save('\DT.mat','Learning_Time','Classification_Time','Mdl','Confusion_D_T','YPredicted');

```

A.9 Code Matlab d'implémentation de l'algorithme RF pour la DSP

```
1 % =====
2 % An example code for RF Learning
3 % Dataset: Implemented using bisection algorithm with IEEE 39-bus
4 % Author: Raoult Teukam Dabou
5 % Power Electronics and Industrial Control Laboratory (LEEPCI), Laval University
6 % Date: 08-10-2021.
7 % =====
8 %% Random Forest
9 load matlab;t1 = cputime;
10 Train=[Training_dataset Testing_dataset];Train=Train';Train=Train(:,1);
    Label=[Label_train Label_Test];Label=Label';Label=Label(:,1);
11 Testing_dataset=Testing_dataset(1,:);Testing_dataset=Testing_dataset';
12 nTrees=14; B=TreeBagger(nTrees,Train,Label,'Method','classification'); Learning_Time = cputime-t1;
13 t2 = cputime;
14 predChar1=B.predict(Testing_dataset); % Predictions is a char though. We want it to be a number.
15 YPredicted=str2double(predChar1); Label_test=Label_Test(1,:);YPredicted=YPredicted';
16 Confusion_R_F=plotconfusion(Label_test,YPredicted);
17 Classification_Time = cputime-t2;
18 title('Confusion_R_F')
19 save('\RF.mat','Learning_Time','Classification_Time','B','Confusion_R_F','YPredicted');
```

A.10 Code Matlab d'implémentation de l'algorithme ANN pour la DSP

```
1 % =====
2 % An example code for ANN Learning.
3 % Dataset: Implemented using bisection algorithm with IEEE 39-bus
4 % Author: Raoult Teukam Dabou
5 % Power Electronics and Industrial Control Laboratory (LEEPCI), Laval University
6 % Date: 08-10-2021.
7 % =====
8 %% Neural Network
9 load matlab;
10 Train=[Training_dataset Testing_dataset];Train=Train';Train=Train(:,1);
    Label=[Label_train Label_Test];Label=Label';Label=Label(:,1);
11 Testing_dataset=Testing_dataset(1,:);Testing_dataset=Testing_dataset';
12 t1 = cputime;
13 net = fitnet(92);% Fit a ANN classifier model on the training samples.
14 Train_data=Train';Label=Label';
15 net = train(net,Train_data,Label);% Train the network net using the training data.
16 Learning_Time = cputime-t1;t2 = cputime;
17 Testing_dataset2=Testing_dataset';Label_test=Label_Test(1,:);
18 YPredicted= net(Testing_dataset2);%Estimate the targets using the trained network.
```

```

19 Confusion_A_N_N=plotconfusion(Label_test,YPredicted); Classification_Time = cputime-t2;
20 title('Confusion_A_N_N IEEE 39 bus')
21 save('\ANNS.mat','Learning_Time','Classification_Time','net','Confusion_A_N_N','YPredicted');

```

A.11 Code Matlab d'implémentation de l'algorithme AdaBoost pour la DSP

```

1 % =====
2 % An example code for AdaBoost Learning.
3 % Dataset: Implemented using bisection algorithm with IEEE 39-bus
4 % Author: Raoult Teukam Dabou
5 % Power Electronics and Industrial Control Laboratory (LEEPCI), Laval University
6 % Date: 08-10-2021.
7 % =====
8 %% Adaptive Boosting
9 load matlab; t1 = cputime;
10 Train=[Training_dataset Testing_dataset];Train=Train';Train=Train(:,1);Label=[Label_train Label_Test];
11 Label=Label';Label=Label(:,1); Testing_dataset=Testing_dataset(1,:);Testing_dataset=Testing_dataset';
12 ClassTreeEns = fitensemble(Train,Label,'AdaBoostM1',104,'Tree');
13 Learning_Time = cputime-t1; t2 = cputime;
14 YPredicted = predict(ClassTreeEns,Testing_dataset);
15 Label_test=Label_Test(1,:);YPredicted=YPredicted';
16 Confusion_A_d_a_B_o_o_s_t=plotconfusion(Label_test,YPredicted);
17 Classification_Time = cputime-t2;
18 title('Confusion_A_d_a_B_o_o_s_t IEEE 39 bus')
19 save('\Adaboost.mat','Learning_Time','Classification_Time','ClassTreeEns',
19 'Confusion_A_d_a_B_o_o_s_t','YPredicted');

```

A.12 Code Matlab d'implémentation de l'algorithme CNN pour la DSP

```

1 % =====
2 % An example code for CNN Learning.
3 % Dataset: Implemented using bisection algorithm with IEEE 39-bus
4 % Author: Raoult Teukam Dabou
5 % Power Electronics and Industrial Control Laboratory (LEEPCI), Laval University
6 % Date: 08-10-2021.
7 % =====
8 load matlab;
9 %% CNN
10 t1 = cputime; XTrain=[Training_dataset Testing_dataset];XTrain=XTrain(1:441,:);
11 XTrain = reshape(XTrain,21,21,1,7871);
12 YTrain=[Label_train Label_Test];YTrain=YTrain(1:441,:);YTrain=categorical(YTrain(1,:));

```

```

    Testing_dataset=Testing_dataset(1:441,:);XTest=reshape(Testing_dataset,21,21,1,2361);
12 XTest=reshape(Testing_dataset,3,169,1,324);
13 YTest=categorical(Label_Test(1,:));
14 layers = [
15     imageInputLayer([21 21 1], 'Name', 'input')
16     convolution2dLayer(5, 20, 'Name', 'conv_1')
17     reluLayer('Name', 'relu_1')
18     convolution2dLayer(3, 20, 'Padding', 1, 'Name', 'conv_2')
19     reluLayer('Name', 'relu_2')
20     convolution2dLayer(3, 20, 'Padding', 1, 'Name', 'conv_3')
21     reluLayer('Name', 'relu_3')
22     additionLayer(2,'Name', 'add')
23     fullyConnectedLayer(2, 'Name', 'fc')
24     softmaxLayer('Name', 'softmax')
25     classificationLayer('Name', 'classoutput')];
26 lgraph = layerGraph(layers);
27 lgraph = connectLayers(lgraph, 'relu_1', 'add/in2');
28 plot(lgraph);
29 options = trainingOptions('sgdm', 'Plots', 'training-progress');
30 [net,info] = trainNetwork(XTrain, YTrain, lgraph, options);
31 Learning_Time = cputime-t1; t2 = cputime;
32 YPred = classify(net, XTest);
33 accuracy = sum(YTest == YPred)/numel(YTest);
34 Confusion_C_N_N=plotconfusion(YTest,YPred); Classification_Time = cputime-t2;
35 title('Confusion_C_N_N IEEE 39 bus')
36 save('\CNN.mat','Learning_Time','Classification_Time','net','Confusion_C_N_N','YPred');

```

A.13 Code Matlab d'implémentation de l'algorithme LSTM pour la DSP

```

1 % =====
2 % An example code for LSTM Learning.
3 % Dataset: Implemented using bisection algorithm with IEEE 39-bus
4 % Author: Raoult Teukam Dabou
5 % Power Electronics and Industrial Control Laboratory (LEEPCI), Laval University
6 % Date: 08-10-2021.
7 % =====
8 load matlab;
9 %% LSTM
10 t1 = cputime;
11 XXXTrain=[Training_dataset Testing_dataset];YYYTrain=[Label_train Label_Test];
12 XTrain=num2cell(XXXTrain); YTrain=categorical(YYYTrain); YTrain=YTrain(1,:);YTrain=YTrain';
13 for i=1:size(YYYTrain,2)
14     XXTrain{i} = cell2mat(XTrain(:,i));
15 end

```

```

16 inputSize =size(Label_train,1);numHiddenUnits1 = 85;numHiddenUnits2 = 85;numClasses = 2;
17 maxEpochs = 100; miniBatchSize = 27;
18 layers = [ ...
19     sequenceInputLayer(inputSize , 'Name','InputLayer' )
20     lstmLayer(numHiddenUnits1,'OutputMode','sequence' , 'Name','LSTM_Layer1')
21     lstmLayer(numHiddenUnits2,'OutputMode','last' , 'Name','LSTM_Layer2')
22     fullyConnectedLayer( numClasses , 'Name','Fully_Connected_Layer')
23     softmaxLayer('Name','Softmax_Layer')
24     classificationLayer('Name','Classification_Layer')
25 ];
26 options = trainingOptions('sgdm', ...
27     'ExecutionEnvironment','auto', ...
28     'InitialLearnRate',0.01, ...
29     'LearnRateSchedule','piecewise', ...
30     'LearnRateDropPeriod',20, ...
31     'MaxEpochs',maxEpochs, ...
32     'MiniBatchSize',miniBatchSize, ...
33     'SequenceLength','longest', ...
34     'Shuffle','never', ...
35     'Verbose',0, ...
36     'Plots','training-progress');
37 net = trainNetwork(XXTrain,YTrain,layers,options);
38 Learning_Time = cputime-t1;t2 = cputime;
39 Testing_dataset=Testing_dataset;Label_Test=Label_Test;
40 XTest=num2cell(Testing_dataset);YTest=categorical(Label_Test);YTest=YTest(1,:);YTest=YTest';
41 for ii=1:size(Label_Test,2)
42     XXTest{ii} = cell2mat(XTest(:,ii));
43 end
44 miniBatchSize = 27;
45 YPred = classify(net,XXTest, ...
46     'MiniBatchSize',miniBatchSize, ...
47     'SequenceLength','longest');
48 Confusion_L_S_T_M=plotconfusion(YTest,YPred); Classification_Time = cputime-t2;
49 title('Confusion_L_S_T_M IEEE 39 bus')
50 save('.\LSTM.mat','Learning_Time','Classification_Time','net','Confusion_L_S_T_M','YPred');

```