

Titre: Nouvelle approche de détermination des paramètres de la machine asynchrone pour les transitoires électromagnétiques
Title: asynchrone pour les transitoires électromagnétiques

Auteur: Steven Joubert
Author:

Date: 2022

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Joubert, S. (2022). Nouvelle approche de détermination des paramètres de la machine asynchrone pour les transitoires électromagnétiques [Mémoire de maîtrise, Polytechnique Montréal]. PolyPublie.
Citation: <https://publications.polymtl.ca/10587/>

 **Document en libre accès dans PolyPublie**
Open Access document in PolyPublie

URL de PolyPublie: <https://publications.polymtl.ca/10587/>
PolyPublie URL:

Directeurs de recherche: Jean Mahseredjian, & Mircea Fratila
Advisors:

Programme: Génie électrique
Program:

POLYTECHNIQUE MONTRÉAL

affiliée à l'Université de Montréal

**Nouvelle approche de détermination des paramètres de la machine asynchrone
pour les transitoires électromagnétiques**

STEVEN JOUBERT

Département de génie électrique

Mémoire présenté en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*

Génie électrique

Novembre 2022

POLYTECHNIQUE MONTRÉAL

affiliée à l'Université de Montréal

Ce mémoire intitulé :

**Nouvelle approche de détermination des paramètres de la machine asynchrone
pour les transitoires électromagnétiques**

présenté par **Steven JOUBERT**

en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*

a été dûment accepté par le jury d'examen constitué de :

Antoine LESAGE-LANDRY, président

Jean MAHSEREDJIAN, membre et directeur de recherche

Mircea FRATILA, membre et codirecteur de recherche

Saad OMAR, membre

REMERCIEMENTS

J'aimerais d'abord remercier mon directeur de recherche Jean Mahseredjian pour sa confiance et ses conseils tout au long du projet. J'aimerais également remercier Mircea Fratila et Paul Akiki de EDF ainsi que mes collègues de la chaire pour leur aide. Finalement, une mention spéciale à ma famille pour leur support inconditionnel depuis le début de mes études.

RÉSUMÉ

Le calculateur des paramètres de la machine asynchrone précédemment développé pour le logiciel EMTP (logiciel utilisé pour la simulation des transitoires électromagnétiques) utilise un modèle à rotor complexe qui inclut la saturation variable des inductances de fuite. Malgré l'ajout de l'algorithme *Particle Swarm Optimization* (PSO), ce calculateur ne parvient pas à trouver des paramètres pour toutes les machines, particulièrement pour celles de faible puissance. Les résultats n'ont également pas été validées directement en simulation.

L'objectif de ce projet de maîtrise est donc, dans un premier temps, d'améliorer le calcul des paramètres en se basant sur les données du manufacturier, puis dans un deuxième temps de comparer les résultats analytiques avec les résultats de simulation et d'apporter des modifications aux paramètres si nécessaire.

Tout d'abord, un modèle à cage double qui conserve la saturation variable des inductances de fuite et qui considère les pertes fer sous la forme d'une résistance à l'entrée du modèle est utilisé, alors que la fonction objectif qui permet de calculer l'erreur sur les spécifications est ajustée afin de prendre en compte un glissement nominal variable. De plus, l'algorithme PSO est conservé, mais l'ajustement dynamique des bornes est ajouté. Toutes ces modifications résultent en un nouveau calculateur nommé calculateur à glissement variable (CGV). Le CGV présente un taux de convergence de 95%, soit une nette amélioration par rapport au calculateur précédent qui présentait un taux de convergence de 57%.

Ensuite, les résultats analytiques sont comparés avec les résultats de simulation afin de valider le modèle de la machine asynchrone dans EMTP. Les résultats sont parfaitement identiques lorsque la partie saturable des inductances de fuite est fixe, alors qu'il y a une erreur non négligeable lorsque ce n'est pas le cas. Puisque le CGV n'est pas aussi performant lorsque ce paramètre est fixe et qu'il est présentement impossible de le modifier dans EMTP, deux méthodes distinctes sont proposées afin de corriger ce problème. La première méthode analyse l'influence de chaque paramètre sur les spécifications de la machine dans le but de modifier les paramètres calculés à partir du modèle analytique, alors que la deuxième méthode incorpore l'erreur entre le modèle analytique et la simulation dans le calcul des paramètres. Les deux méthodes utilisent un API afin de lancer les simulations dans EMTP à partir de MATLAB sans avoir à modifier manuellement le fichier de simulation.

Finalement, le régime transitoire de la machine est analysé, autant à travers les paramètres de la simulation que les paramètres électromécaniques. Les oscillations sont réduites en laissant suffisamment de temps à la machine pour se stabiliser avant de modifier la commande de vitesse, alors qu'une formule est utilisée pour calculer le moment d'inertie de la machine à partir de l'essai de démarrage à vide. Ce paramètre est nécessaire afin de bien représenter le régime transitoire de la machine lorsqu'elle n'est pas contrôlée en vitesse.

La principale contribution de ce projet de maîtrise est une amélioration de la détermination des paramètres de la machine asynchrone à l'aide, entre autres, du calcul de l'erreur qui considère un glissement nominal variable ainsi que de l'ajout de bornes dynamiques sur les paramètres. Une preuve de concept pour deux méthodes distinctes qui permettent d'ajuster les paramètres de la machine de manière dynamique en fonction des résultats de simulation est également présentée.

ABSTRACT

The asynchronous machine parameters calculator previously developed for the EMTP (Electromagnetic Transients Program) software uses a complex rotor model that includes variable saturation of the leakage inductances. Despite the addition of the particle swarm optimization (PSO) algorithm, this calculator fails to find parameters for all machines, especially for low power machines. Also, the results have not been directly validated in simulation.

The objective of this Master's thesis is first to improve the calculation of parameters based on manufacturer's data, and then to compare analytical results with simulation results and to make modifications to parameters if necessary.

This work starts with the development of a double-cage model that preserves the variable saturation of the leakage inductances and considers iron losses as a resistance at the input of the model, while the objective function that calculates the error on the specifications is adjusted to consider a variable nominal slip. In addition, the PSO algorithm is kept, but dynamic boundary adjustment is added. All these modifications result in a new calculator called variable slip calculator (CGV). The CGV presents a convergence rate of 95%, which is a clear improvement compared to the previous calculator which presented a convergence rate of 57%.

Then, analytical results are compared with simulation results to validate the machine model in EMTP. The results are perfectly identical when the saturable part of the leakage inductance is fixed, while there is a non-negligible error when this is not the case. Since the CGV does not perform as well when this parameter is fixed and since it is currently impossible to modify it in EMTP, two distinct methods are proposed to correct this problem. The first method analyzes the influence of each parameter on the machine specifications to modify the parameters calculated from the analytical model, while the second method incorporates the error between the analytical model and the simulation in the calculation of the parameters. Both methods use an API to run the simulations in EMTP from MATLAB without having to manually modify the simulation file.

Finally, the transient regime of the machine is analyzed, both through the simulation parameters and the electromechanical parameters. Oscillations are reduced by allowing sufficient time for the machine to stabilize before changing the speed command, while a formula is developed to calculate the moment of inertia of the machine from the no-load start-up test. This parameter is necessary to properly represent the transient regime of the machine when it is not speed controlled.

The main contribution of this Master's thesis is an improvement in the determination of asynchronous machine parameters using, among others, the calculation of error considering a variable nominal slip as well as addition of dynamic bounds on parameters. A proof of concept for two separate methods that allow the machine parameters to be adjusted dynamically according to the simulation results is also presented.

TABLE DES MATIÈRES

REMERCIEMENTS	III
RÉSUMÉ.....	IV
ABSTRACT	VI
TABLE DES MATIÈRES	VIII
LISTE DES TABLEAUX.....	X
LISTE DES FIGURES	XI
LISTE DES SIGLES ET ABRÉVIATIONS	XIII
LISTE DES ANNEXES.....	XIV
CHAPITRE 1 INTRODUCTION.....	1
CHAPITRE 2 CADRE THÉORIQUE	5
2.1 Description de la machine asynchrone.....	5
2.2 Modèle électrique et paramètres	7
2.2.1 Calcul des spécifications	8
2.2.2 Saturation	9
2.3 Fonction objectif	11
2.3.1 Spécifications de démarrage.....	13
2.3.2 Spécification de décrochage.....	13
2.3.3 Spécifications nominales.....	14
2.4 Algorithme d'optimisation.....	16
2.4.1 Initialisation.....	16
2.4.2 PSO.....	18
2.4.3 Recherche adaptative.....	19
2.4.4 Critères d'arrêt.....	21

2.4.5 Bornes dynamiques	22
CHAPITRE 3 MÉTHODE ANALYTIQUE	25
3.1 Code MATLAB.....	25
3.2 Performance du CGV	28
3.3 Analyse de convergence	31
3.3.1 Cas de convergence	31
3.3.2 Cas de divergence.....	36
CHAPITRE 4 MÉTHODE DYNAMIQUE.....	42
4.1 API MATLAB-EMTP.....	42
4.2 Écart entre le modèle simulé et le modèle analytique	45
4.3 Calcul dynamique des paramètres	48
4.3.1 Méthode 1 – Calcul basé sur l'influence des paramètres	48
4.3.2 Méthode 2 – Calcul basé sur l'erreur	53
4.3.3 Comparaison entre les deux méthodes	55
CHAPITRE 5 RÉGIME TRANSITOIRE	57
5.1 Paramètres de simulation	57
5.2 Modèle électromécanique	63
CHAPITRE 6 CONCLUSION	69
RÉFÉRENCES	71
ANNEXES	73

LISTE DES TABLEAUX

Tableau 2.1 : Valeurs numériques des coefficients de l'optimisation.....	20
Tableau 3.1 : Temps de calcul approximatif selon les paramètres de l'optimisation	28
Tableau 3.2 : Performance du CGV	29
Tableau 3.3 : Spécifications de la machine 1	31
Tableau 3.4 : Paramètres de la machine 1	31
Tableau 3.5 : Erreurs sur les spécifications de la machine 1.....	33
Tableau 3.6 : Spécifications de la machine 10	35
Tableau 3.7 : Paramètres de la machine 10.....	35
Tableau 3.8 : Erreurs sur les spécifications de la machine 10.....	36
Tableau 3.9 : Relation entre le couple de décrochage et le couple de démarrage.....	37
Tableau 3.10 : Paramètres des cages du rotor et erreur maximale pour les cas divergents	38
Tableau 4.1 : Paramètres de la machine 10 pour PS = 0.5	47
Tableau 4.2 : Influence sur les spécifications pour un facteur de correction de 0.8	49
Tableau 4.3 : Influence sur les spécifications pour un facteur de correction de 1.2	49
Tableau 4.4 : Erreur sur les spécifications en simulation.....	50
Tableau 4.5 : Erreur sur les spécifications en simulation après correction – Méthode 1.....	51
Tableau 4.6 : Erreur sur les spécifications en simulation après correction – Méthode 2.....	54
Tableau 5.1 : Spécifications de la machine de 37 kW	64
Tableau 5.2 : Paramètres de la machine de 37 kW	64

LISTE DES FIGURES

Figure 2.1 : Modèle à cage double de la machine asynchrone.....	7
Figure 2.2 : Courbes analytiques de la machine asynchrone	13
Figure 2.3 : Décalage du couple nominal de la machine 1	14
Figure 2.4 : Agrandissement du décalage du couple nominal de la machine 1	15
Figure 2.5 : Schéma de l'algorithme PSO	24
Figure 3.1 : Structure du code MATLAB	26
Figure 3.2 : Erreur maximale pour chaque machine de la base de données	30
Figure 3.3 : Courbes analytiques de la machine 1.....	32
Figure 3.4 : Agrandissement des courbes analytiques de la machine 1	34
Figure 3.5 : Courbes analytiques de la machine 10.....	36
Figure 3.6 : Courbe du couple pour les cas divergents – Modifications acceptables	39
Figure 3.7 : Courbe du couple pour les cas divergents – Modifications non acceptables	40
Figure 4.1 : Schéma EMTP	43
Figure 4.2 : Schéma MATLAB-EMTP	44
Figure 4.3 : Courbes simulées et analytiques de la machine 10.....	46
Figure 4.4 : Courbes simulées et analytiques du couple de la machine 10 pour PS = 0.5	47
Figure 4.5 : Courbes simulées et analytiques après correction – Méthode 1	52
Figure 4.6 : Courbes simulées et analytiques après correction – Méthode 2	55
Figure 5.1 : Oscillations pour une simulation de 10s	58
Figure 5.2 : Oscillations pour une simulation de 100s	58
Figure 5.3 : Oscillations pour une simulation de 1000s	59
Figure 5.4 : Oscillations pour une simulation de 10s et un filtre à 50Hz.....	60
Figure 5.5 : Transitoire du couple à vitesse nulle	61

Figure 5.6 : Transitoire du couple à vitesse nominale.....	61
Figure 5.7 : Courbe discrète du couple	62
Figure 5.8 : Démarrage à vide	65
Figure 5.9 : Comparaison du temps de démarrage dynamique pour $D = 0.0307$	67
Figure 5.10 : Comparaison du temps de démarrage dynamique pour $D = 0$	68

LISTE DES SIGLES ET ABRÉVIATIONS

CGV	Calculateur à glissement variable
I_n	<i>Nominal Current</i> – Courant nominal
I_{st}	<i>Starting Current</i> – Courant de démarrage
LB	<i>Lower Bound</i> – Borne inférieure
pf_n	<i>Nominal Power Factor</i> – Facteur de puissance nominal
PSO	<i>Particle Swarm Optimization</i> – Optimisation par essaims particulaires
RPM	Rotation par minute
T_{br}	<i>Breaking Torque</i> – Couple de décrochage
T_n	<i>Nominal Torque</i> – Couple nominal
T_{st}	<i>Starting Torque</i> – Couple de démarrage
UB	<i>Upper bound</i> – Borne supérieure

LISTE DES ANNEXES

ANNEXE A – Paramètres des figures	73
---	----

CHAPITRE 1 INTRODUCTION

La machine asynchrone, aussi appelée machine à induction, représente de 60% à 70% de l'énergie consommée sur le réseau électrique, principalement dans le secteur industriel. Elle est robuste et son coût est relativement faible en comparaison avec d'autres types de machines, ce qui en fait un choix populaire. Les applications qui nécessitent une machine asynchrone sont nombreuses et souvent critiques.

Les logiciels de simulation pour les réseaux électriques, comme EMTP, utilisent des modèles équivalents afin de représenter la machine asynchrone. Ces logiciels sont utilisés pour effectuer des analyses de défaillance ou de stabilité sur les réseaux électriques, d'où l'utilité d'un modèle qui représente le plus fidèlement possible le comportement de la machine sous diverses conditions d'opération. Le remplacement d'une machine déjà existante par une nouvelle machine peut également être étudié à l'aide de ce type de logiciel afin de prévenir des problèmes d'intégration.

La plupart du temps, les manufacturiers ne fournissent pas les paramètres du modèle équivalent de la machine. Ils fournissent plutôt les spécifications de la machine à différents points d'opération, comme le couple, le courant ou encore la puissance. Ces données sont disponibles dans des catalogues ou directement sur la plaque signalétique de la machine. Il est également possible d'effectuer des essais standardisés afin d'obtenir plus d'informations sur le comportement de la machine ou même pour mesurer directement certains paramètres comme les résistances statoriques. Il faut toutefois avoir un accès à la machine, ce qui n'est pas toujours le cas pour des machines déjà en opération ou encore en fabrication. Ainsi, les paramètres du modèle équivalent doivent bien souvent être obtenus à partir des spécifications du manufacturier seulement.

Le problème réside dans le fait que les calculateurs de paramètres, notamment celui intégré dans EMTP, ne permettent pas toujours de trouver des paramètres qui sont adéquats. Les ingénieurs en charge d'une étude doivent alors modifier manuellement les paramètres du modèle afin d'obtenir des résultats satisfaisants.

Le projet de recherche précédemment mené par François Gauthier [2] puis poursuivi par Younes Seniani [3], établit les bases du projet présenté dans ce mémoire. Un nouveau modèle à rotor complexe qui inclut la saturation variable des inductances de fuite a d'abord été avancé, puis les formules mathématiques et le code du calculateur ont été développés. Le modèle proposé présente

toutefois des lacunes, notamment pour les machines de faibles puissances. L'auteur laisse aussi entendre que le modèle n'est pas adéquat pour toutes les machines. Il suggère d'avoir recours à la simulation afin de calculer les paramètres de manière dynamique plutôt que de se limiter au calcul analytique. Ensuite, le nouveau calculateur a été intégré dans EMTP et un algorithme d'optimisation plus performant a été ajouté sans toutefois régler le problème de convergence du calculateur. La recherche d'un nouveau modèle, de même que l'amélioration du calculateur, d'abord au niveau du calcul analytique puis au niveau du calcul dynamique, en fonction des résultats de simulation, constituent donc les objectifs principaux du projet de recherche actuel.

La plupart des modèles proposés dans la littérature sont simples. C'est notamment le cas des modèles dans les normes IEEE [4] et IEC [5] qui possèdent un rotor à cage simple. En comparaison, le modèle avec un rotor à cage double est plus précis. Ce modèle permet de représenter le comportement de la machine autant aux conditions de démarrage que nominales, ce qui est souvent un problème avec les modèles plus simples. Il s'agit donc d'un modèle robuste qui peut être utilisé pour tous les types de machines, même dans le cas où la machine est à cage simple par construction [6]. Il existe différentes configurations pour représenter le modèle à cage double. Cependant, il a été démontré que ces configurations sont équivalentes en autant qu'elles contiennent un certain nombre de variables indépendantes [7]. Dans la littérature, la résistance représentant les pertes fer est souvent négligée, ce qui affecte l'exactitude du modèle au niveau de la puissance à vide et du facteur de puissance. Finalement, la saturation des inductances de fuite n'est rarement, voire jamais considérée dans les modèles rencontrés.

L'un des modèles présentement disponibles dans EMTP est un modèle à cage double. La résistance des pertes fer n'est toutefois pas présente, alors que les inductances du stator et de la première branche du rotor suivent la fonction de saturation définie dans [8]. En théorie, il s'agit donc d'un modèle adéquat qui peut toutefois être amélioré de manière relativement simple en intégrant la résistance des pertes fer. Peu importe le modèle choisi, il faut vérifier que les paramètres respectent les spécifications du manufacturier. La fonction objectif est ce qui permet de comparer les spécifications théoriques de la machine qui proviennent des données du manufacturier avec les spécifications du modèle obtenues à partir des formules analytiques. Pour chacune des spécifications considérées, l'erreur entre la spécification théorique et analytique est calculée. Plus simplement, la fonction objectif permet d'évaluer si un ensemble de paramètres est bon ou mauvais.

Les spécifications qui sont comparées, sont généralement le couple et le courant de démarrage, ainsi que le courant et le facteur de puissance aux conditions nominales. Une autre approche est de considérer seulement les puissances active et réactive aux conditions nominales, ce qui permet d'éviter des cas de divergence [9]. Les deux méthodes sont valides, tout dépend de la précision et des critères de comparaison qui sont considérés. Le couple de décrochage est une spécification qui n'est pas toujours prise en compte, même s'il s'agit d'un critère important dans plusieurs études. Il est à noter que plus la fonction objectif contient de contraintes, plus le calcul a des chances de diverger ou de ne pas trouver un ensemble de paramètres satisfaisants, ce qui peut expliquer l'omission de cette spécification dans certains cas.

Une fois que la fonction objectif est définie, il faut un algorithme d'optimisation afin de déterminer les paramètres de la machine. L'algorithme calcule l'ensemble de paramètres qui minimise les erreurs en se basant sur la fonction objectif. Certains algorithmes calculent directement les paramètres à partir des données du manufacturier [10-12]. Ces techniques sont plutôt utiles pour trouver une estimation initiale des paramètres qui est relativement près de la solution finale.

Les algorithmes plus complexes ont généralement recours à un calcul itératif qui permet de réduire progressivement l'erreur sur les spécifications. D'abord, il y a les algorithmes basés sur le calcul numérique de la dérivée [6, 13, 14], comme la méthode de Newton. C'est d'ailleurs ce type d'algorithme qui est utilisé dans EMTP. Ces algorithmes sont relativement rapides, mais ils ont tendance à rester coincés près d'un optimum local et ils dépendent grandement de l'estimation initiale des paramètres. Ensuite, les algorithmes de type métaheuristiques font récemment l'objet de plusieurs articles sur le calcul des paramètres de la machine asynchrone [15-22]. On retrouve dans cette catégorie les algorithmes évolutionnaires comme les algorithmes génétiques ou encore ceux inspirés par la nature, comme l'algorithme PSO, le nouvel algorithme ajouté dans [3]. Ces algorithmes sont généralement plus lents et sont de nature aléatoire. Ils permettent toutefois de trouver une solution à des problèmes complexes où l'espace de solution est vaste. Plus récemment, les algorithmes reliés à l'intelligence artificielle et aux réseaux de neurones commencent aussi à faire leur apparition [23].

La plupart des auteurs dans la littérature présentent un algorithme d'optimisation appliqué au calcul des paramètres de la machine asynchrone qui est soi-disant plus rapide et efficace que les autres. Peu d'auteurs se concentrent sur l'espace solution en tant que tel, soit les bornes qui doivent être

définies sur chacun des paramètres. En ce sens, [24] laisse entendre que l'important est de bien cibler l'espace solution de la machine afin d'optimiser le calcul des paramètres. Ainsi, n'importe lequel des algorithmes présentés plus haut peut en théorie obtenir des résultats satisfaisants en autant qu'il soit bien paramétré pour le problème à l'étude.

Même si tous les éléments mentionnés plus haut sont pris en considération et qu'un ensemble de paramètres satisfaisants est déterminé, il faut tout de même vérifier que le comportement de la machine est bien simulé une fois que le modèle est intégré à un logiciel de simulation comme EMTP, tel que suggéré dans [2]. Cet aspect cause des problèmes lorsque vient le temps de simuler les caractéristiques de la machine asynchrone alors que le comportement analytique ne se reflète pas toujours dans la simulation. Ce mémoire tente donc de présenter une solution qui est acceptable autant au niveau analytique que dans le domaine de la simulation.

Tout d'abord, le chapitre 2 contient le cadre théorique. Le modèle complet de la machine asynchrone à cage double de même que les équations analytiques qui décrivent son comportement sont détaillés, en plus de la fonction objectif et de l'algorithme d'optimisation. Le chapitre 3 adresse la méthode analytique pour déterminer les paramètres du modèle. Une description du code MATLAB ainsi que les performances du CGV sont présentées, notamment à travers l'exemple d'un cas convergent et d'un cas divergent. Le chapitre 4 développe sur la méthode dynamique pour déterminer les paramètres du modèle dans le logiciel de simulation EMTP. L'utilisation d'un API afin de communiquer entre MATLAB et EMTP est d'abord proposé et un exemple concret qui démontre la nécessité de cette méthode est ensuite mis de l'avant. Ensuite, deux méthodes distinctes qui permettent de calculer les paramètres du modèle de manière dynamique sont suggérées, soit une méthode basée sur les paramètres et une autre sur l'erreur. Étant donné que les résultats précédents sont valides dans le régime statique, le chapitre 5 est consacré à l'étude du régime transitoire de la machine. L'influence des paramètres de simulation sur les oscillations est d'abord analysée, puis le moment d'inertie est calculé à partir des données d'essais. Finalement, une conclusion et des recommandations sont fournies au chapitre 6.

CHAPITRE 2 CADRE THÉORIQUE

La machine asynchrone, qu'elle soit utilisée en mode moteur ou en mode générateur, possède des caractéristiques qui lui sont propres et qui la distingue des autres machines tournantes électriques. Une description tirée de [25] est d'abord donnée afin de bien comprendre son fonctionnement. Ensuite, le modèle à cage double et ses paramètres sont clairement explicités, de même que les équations qui permettent de calculer les spécifications de la machine à partir du modèle analytique. Finalement, la fonction objectif utilisée pour évaluer l'erreur entre les spécifications théoriques et analytiques ainsi que l'algorithme PSO sont décrits en détails.

2.1 Description de la machine asynchrone

Alors que les premiers moteurs à induction font leur apparition dès 1885-1886 grâce à Galileo Ferraris et Nikola Tesla, le fonctionnement est demeuré le même depuis. La machine asynchrone est composée de deux parties, soit le stator et le rotor. En mode moteur, une source de tension alternative alimente le stator. Un champ magnétique tournant à la fréquence de la source d'alimentation est produit dans l'entrefer entre le stator et le rotor. Le champ magnétique induit une tension et ultimement un courant au rotor, que ce soit à travers les barres qui sont court-circuitées dans le cas d'un rotor à cage ou à travers une impédance externe dans le cas d'un rotor bobiné. Au fur et à mesure que le champ magnétique produit par le stator se déplace et que le courant induit au rotor varie, le rotor se met en rotation dans le même sens que le champ tournant afin de s'opposer à cette variation selon le principe de la loi de Lenz. Le rotor ne peut toutefois pas atteindre la vitesse du champ tournant puisqu'il n'y aurait alors plus de variation du champ magnétique vue par le rotor. C'est donc la différence de vitesse entre le stator et le rotor qui force le rotor à tourner et qui engendre un couple électromagnétique. Le mode générateur fonctionne de manière opposée, c'est-à-dire que la rotation du rotor par un couple externe engendre une tension et un courant au stator.

Pour le reste de ce mémoire, on considère seulement le mode moteur de la machine asynchrone. Les termes machines et moteurs sont donc interchangeables et désignent tous les deux la machine asynchrone en mode moteur. On considère également que la machine est alimentée par une source triphasée, ce qui est généralement le cas sur un réseau électrique.

Il est possible d'établir les équations générales qui décrivent le fonctionnement de la machine asynchrone. La vitesse du champ tournant au niveau du stator, aussi appelée vitesse synchrone, est donnée par l'équation suivante :

$$N_s = \frac{60f}{p} \quad (2.1)$$

où

N_s : vitesse de rotation synchrone [RPM]

f : fréquence de la source d'alimentation [Hz]

p : nombre de paire(s) de pôles

Il est à noter qu'il est également possible d'exprimer la vitesse de rotation en Hz, en rad/s ou dans tout autre unité. Étant donné que les manufacturiers fournissent généralement la vitesse en RPM, cette unité est choisie pour le reste de ce mémoire.

Tel que mentionné précédemment, la vitesse de rotation du rotor est nécessairement différente de la vitesse synchrone afin de générer un couple électromagnétique. La différence de vitesse entre le stator et le rotor est exprimée par le glissement :

$$s = \frac{N_s - N_r}{N_s} \quad (2.2)$$

où

s : glissement

N_r : vitesse de rotation du rotor [RPM]

Généralement, le glissement correspond à un faible pourcentage de la vitesse synchrone et est proportionnel à la puissance de la machine. Une machine de faible puissance aura donc un glissement plus petit qu'une machine de puissance plus élevée.

2.2 Modèle électrique et paramètres

Le modèle développé est présenté à la Figure 2.1.

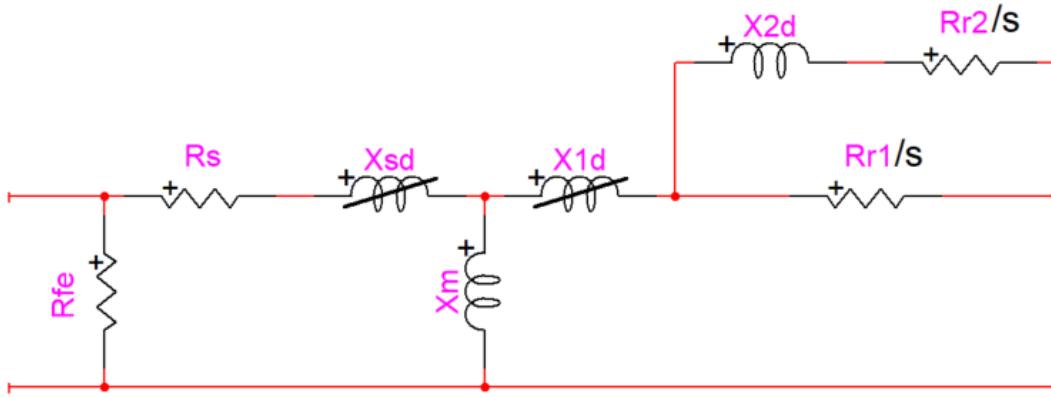


Figure 2.1 : Modèle à cage double de la machine asynchrone

où

R_{fe} : résistance des pertes fer [Ω]

R_s : résistance du stator [Ω]

R_{r1} : résistance de la première branche du rotor [Ω]

R_{r2} : résistance de la deuxième branche du rotor [Ω]

X_{sd} : réactance du stator [Ω]

X_m : réactance de magnétisation [Ω]

X_{1d} : réactance de la première branche du rotor [Ω]

X_{2d} : réactance de la deuxième branche du rotor [Ω]

Peu importe le type de machine, le même modèle à cage double est utilisé. Il est plus précis que le modèle à cage simple puisqu'il permet de mieux représenter le comportement de la machine au démarrage ainsi qu'aux conditions nominales. La caractéristique du couple de décrochage est également mieux approximée avec ce modèle. Par construction, la résistance de la première branche du rotor est nécessairement plus petite que celle de la deuxième branche, alors que la relation est inversée entre les inductances des branches du rotor. Afin de considérer les pertes fer, une résistance est ajoutée. L'ajout de R_{fe} devant le modèle plutôt qu'en parallèle avec X_m comme dans le cas d'un transformateur facilite l'intégration avec le modèle existant dans EMTP et permet d'isoler les pertes du reste du circuit [6].

2.2.1 Calcul des spécifications

Afin de calculer les spécifications de manière analytique, il faut d'abord définir les équations pour les impédances et les courants de la machine. Les équations proviennent de l'analyse du circuit de la Figure 2.1 et sont présentées plus bas.

$$Z_{re} = R_{re} + jX_{re} = jX_{1d} + \frac{1}{\frac{1}{R_{r1}} + \frac{1}{\frac{R_{r2}}{s} + jX_{2d}}} \quad (2.3)$$

$$Z_p = \frac{1}{\frac{1}{jX_m} + \frac{1}{Z_{re}}} \quad (2.4)$$

où

Z_{re} : impédance équivalente au rotor [Ω]

R_{re} : résistance équivalente au rotor [Ω]

X_{re} : réactance équivalente au rotor [Ω]

Z_p : impédance parallèle [Ω]

$$I_s = \frac{V_{in}}{R_s + jX_{sd} + Z_p} \quad (2.5)$$

$$I_r = I_s \frac{Z_p}{Z_{re}} \quad (2.6)$$

$$I_{fe} = \frac{V_{in}}{R_{fe}} \quad (2.7)$$

$$I_{in} = I_s + I_{fe} \quad (2.8)$$

où

I_s : courant au stator [A]

V_{in} : tension d'entrée [V]

I_r : courant au rotor [A]

I_{fe} : courant des pertes fer [A]

I_{in} : courant d'entrée [A]

Connaissant le courant dans les différentes parties du moteur, il est possible de calculer les autres spécifications à partir des équations suivantes :

$$T = 3 \frac{p}{2\pi f} |I_r|^2 \frac{R_{re}}{s} \quad (2.9)$$

$$S = P + jQ = 3V_{in}I_{in}^* \quad (2.10)$$

$$pf = \cos(\angle I_{in}) = \cos(\phi) \quad (2.11)$$

où

T : couple [Nm]

S : puissance apparente [VA]

P : puissance active [W]

Q : puissance réactive [VAR]

pf : facteur de puissance (entre 0 et 1)

2.2.2 Saturation

La saturation des inductances de fuite permet de représenter la caractéristique non-linéaire de la machine asynchrone. La saturation de l'inductance de magnétisation peut également être représentée, mais il a été décidé dans ce mémoire de représenter seulement la saturation des inductances de fuite. Lorsque le courant qui circule dans une inductance devient trop important, celle-ci entre en saturation. La fonction de saturation qui est utilisée est la même fonction qui est utilisée dans le logiciel EMTP. [8] présente d'ailleurs une comparaison entre une inductance de fuite saturée et linéaire, où il est possible d'observer un décalage de phase.

Seule l'inductance de fuite du stator et de la première branche du rotor est saturable. La saturation est définie de la manière suivante :

$$SAT = \begin{cases} \frac{2}{\pi} [\arcsin(\alpha) + \alpha\sqrt{1 - \alpha^2}], & \text{si } \alpha \leq 1 \\ 1, & \text{sinon} \end{cases} \quad (2.12)$$

$$\alpha = \frac{I_{sat}}{I_L} \quad (2.13)$$

où

SAT : fonction de saturation

I_{sat} : courant de saturation [A]

I_L : courant dans l'inductance [A]

La réactance saturée est alors définie selon la fonction de saturation et selon sa partie saturable.

$$X_{sat} = X[(1 - PS) + PS SAT] \quad (2.14)$$

où

X_{sat} : réactance saturée [Ω]

X : réactance non saturée [Ω]

PS : partie saturable de l'inductance de fuite (entre 0 et 1)

La saturation est présente seulement lorsque le courant qui circule dans l'inductance est plus grand que la limite définie par I_{sat} . Dans EMTP, la partie saturable est fixée à 50% de la réactance, mais il est possible de faire varier cette valeur entre 0% et 100% pour une plus grande flexibilité. De plus, étant donné que l'inductance dépend du courant et que le courant dépend de l'inductance, la saturation est calculée de manière itérative jusqu'à l'obtention d'une convergence.

En incluant la saturation, les paramètres du modèle de la machine asynchrone sont les suivants :

$$x = [R_{fe}, R_s, R_{r1}, R_{r2}, X_{sd}, X_m, X_{1d}, X_{2d}, I_{sat}, PS] \quad (2.15)$$

2.3 Fonction objectif

La fonction objectif permet à l'algorithme de déterminer si les paramètres calculés approximent bien le comportement de la machine. Pour chaque ensemble de paramètres, les spécifications sont calculées à partir des équations analytiques. La solution est jugée en fonction de l'erreur entre la spécification donnée par le manufacturier et celle obtenue analytiquement.

Les spécifications qui sont importantes à considérer sont surtout au démarrage et aux conditions nominales. Différentes approches sont proposées dans la littérature afin d'évaluer les conditions nominales, mais celle qui est exploitée dans ce mémoire utilise le couple, le courant et le facteur de puissance. La puissance peut être obtenue à partir de ces caractéristiques, ce qui la rend redondante. Le couple de décrochage, soit le couple maximal que peut fournir la machine en condition de fonctionnement, est également une spécification très importante à considérer pour représenter adéquatement un moteur. Elle est incluse dans le CGV. Il serait possible d'inclure plus de spécifications dans la fonction objectif, mais la convergence de l'algorithme serait encore plus difficile. Plus il y a de points à faire correspondre, plus il y a de chances de faire diverger le calcul. La fonction objectif $f(x)$ compare donc seulement six spécifications distinctes :

$$\begin{aligned} f_1 &= \Delta T_n (\%) \\ f_2 &= \Delta T_{st} (\%) \\ f_3 &= \Delta T_{br} (\%) \\ f_4 &= \Delta I_n (\%) \\ f_5 &= \Delta I_{st} (\%) \\ f_6 &= \Delta p f_n (\%) \end{aligned} \tag{2.16}$$

où

f_i : i ème fonction objectif

Δ : opérateur de différence

T_n : couple nominal [Nm]

T_{st} : couple de démarrage [Nm]

T_{br} : couple de décrochage [Nm]

I_n : courant nominal [A]

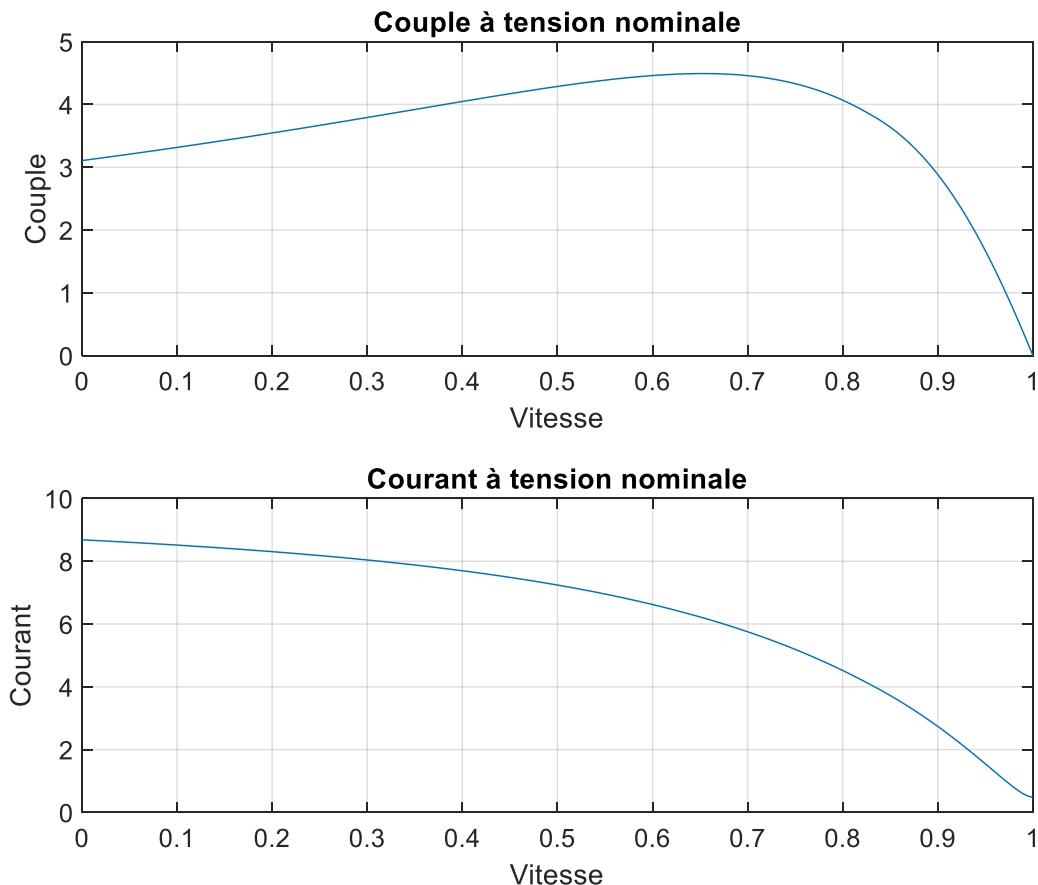
I_{st} : courant de démarrage [A]

$p f_n$: facteur de puissance nominal

La fonction objectif $f(x)$ représente l'évaluation de ces six sous-fonctions pour un ensemble de paramètres. Il est important de ramener ces valeurs en pourcentage afin de pondérer chacune des erreurs entre elles. Ainsi, l'erreur est calculée de manière relative et non en absolue.

Les caractéristiques au démarrage sont relativement simples à calculer. Il suffit d'évaluer le couple et le courant pour un glissement unitaire, soit lorsque la machine possède une vitesse nulle. Le couple de décrochage se calcule plus difficilement étant donné que le glissement à ce point n'est pas fourni par le manufacturier. Pour ce qui des caractéristiques nominales, bien que le glissement nominal soit fourni par le manufacturier, il est également possible de vérifier les valeurs de glissement avoisinante en considérant que la valeur fournie par le manufacturier n'est pas nécessairement bonne. Ces concepts sont explicités davantage dans les sections suivantes.

Avant de continuer plus loin, il est important de comprendre que les équations du couple, du courant et du facteur de puissance dépendent du glissement s en raison des deux résistances rotoriques. Il est donc possible de tracer une courbe en fonction de la vitesse de rotation de la machine pour chacune de ces caractéristiques, telles que présentées à la Figure 2.2.



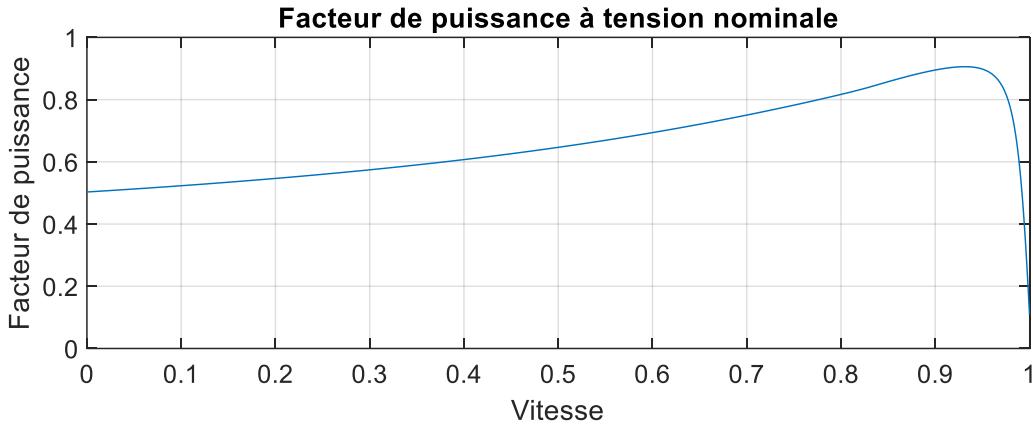


Figure 2.2 : Courbes analytiques de la machine asynchrone

2.3.1 Spécifications de démarrage

Tel que mentionné plus haut, le couple et le courant de démarrage sont obtenues directement à partir des équations analytiques en utilisant un glissement unitaire. Le calcul est direct et ne pose aucun problème, à l'exception de la vitesse nulle qui est remplacée par une valeur très petite de $1e^{-5}$ afin d'éviter une division par zéro. Le manufacturier fournit généralement les valeurs au démarrage comme un multiple de la valeur nominale.

2.3.2 Spécification de décrochage

Le calcul du couple de décrochage nécessite d'évaluer le couple sur un certain intervalle de valeurs de glissement étant donné que le glissement est inconnu à ce point d'opération. Le manufacturier fournit le couple de décrochage comme un multiple de la valeur nominale sans toutefois mentionner le glissement correspondant. Pour rappel, le couple de décrochage correspond au couple maximal que peut fournir la machine avant d'accélérer ou de décélérer brusquement.

Instinctivement, le couple de décrochage est une fonction de la tension d'opération et correspond simplement au couple maximal que peut fournir la machine. Cela presuppose toutefois que le couple de décrochage est supérieur au couple de démarrage, ce qui n'est pas toujours le cas. Dans le cas où le couple de démarrage est plus élevé, il faut plutôt trouver un maximum local près de la vitesse nominale en cherchant la valeur pour laquelle la dérivée du couple est nulle. Dans le cas où la dérivée ne s'annule jamais, le couple maximal correspond alors à la valeur la plus faible de la

dérivée. Cette situation est susceptible de survenir puisque le glissement est représenté par un ensemble de valeurs discrètes et non continues. Ainsi, la méthode proposée considère le couple de démarrage afin de calculer le couple de décrochage de la machine.

$$T_{br} = \begin{cases} T \text{ pour } T' = 0 \text{ ou pour } \min(T'), & \text{si } T_{br} < T_{st} \\ \max(T), & \text{sinon} \end{cases} \quad (2.17)$$

où

T' : dérivée du couple [Nm/RPM]

2.3.3 Spécifications nominales

Le couple, le courant et le facteur de puissance nominaux semblent relativement simples à calculer, mais ce n'est pas tout à fait le cas. En théorie, il suffit d'évaluer les équations analytiques au glissement nominal fourni par le manufacturier. En pratique, on ne parvient pas toujours à trouver une solution satisfaisante en utilisant cette technique.

Un exemple simple avec le couple nominal est présenté à la Figure 2.3 et à la Figure 2.4. Pour référence, les courbes du courant et du facteur de puissance pour cette même machine sont présentées à la Figure 3.3 et à la Figure 3.4 au chapitre suivant, de même que les données du manufacturier et les paramètres du modèle.

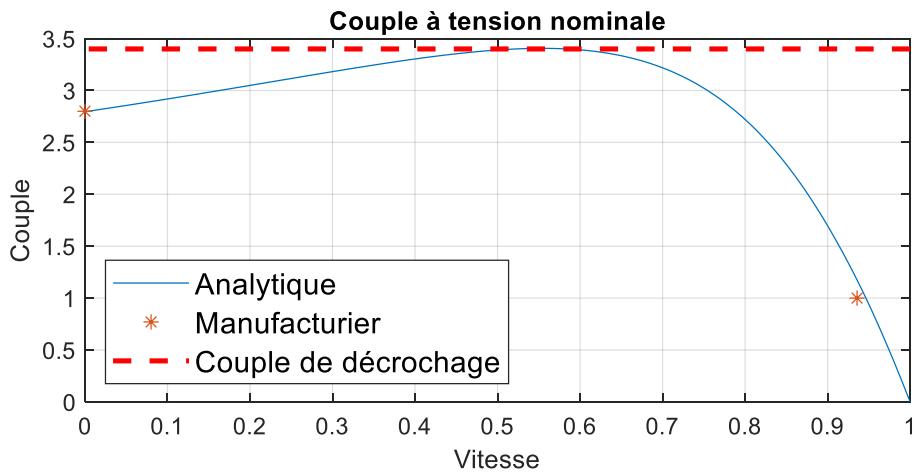


Figure 2.3 : Décalage du couple nominal de la machine 1

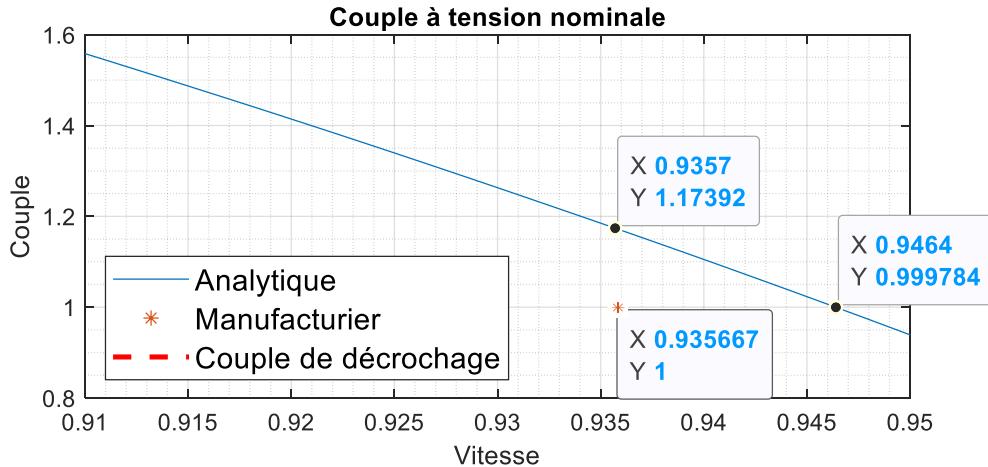


Figure 2.4 : Agrandissement du décalage du couple nominal de la machine 1

En analysant la courbe du couple à la Figure 2.3, on peut observer que la courbe bleue du modèle passe pratiquement par la valeur spécifiée par le manufacturier, soit. Un agrandissement de cette région à la Figure 2.4 permet toutefois de voir que l'erreur est en réalité assez grande, soit 17.39%. En comparaison, l'erreur sur la valeur de la vitesse ou de manière équivalente sur le glissement est de seulement 1.14%. Bien que la courbe semble bien approximer le comportement de la machine, la fonction objectif considère plutôt qu'il s'agit d'une solution inacceptable étant donné que seul le couple est analysé.

$$e_{couple} = \left| \frac{1 - 1.1739}{1} \right| = 17.39\% \quad (2.18)$$

$$e_{glissement} = \left| \frac{0.9357 - 0.9464}{0.9357} \right| = 1.14\% \quad (2.19)$$

Une autre manière d'analyser le problème est de considérer qu'il y a potentiellement une erreur sur la valeur du glissement nominal, ou du moins qu'il faudrait calculer cette erreur au même titre que l'erreur sur la spécification elle-même. Tolérer une erreur sur le couple de démarrage ou de décrochage n'est pas différent que de tolérer une erreur sur le glissement nominal. Il s'agit dans les deux cas d'une valeur fournie par le manufacturier que le modèle tente de reproduire.

L'approche évoquée précédemment est utilisée dans [26]. Au lieu de mesurer l'erreur sur le couple seulement, l'erreur sur le glissement est aussi incluse. Cela revient à calculer la distance la plus courte entre la courbe et le point spécifié par le manufacturier. D'un point de vue physique, cette méthode présente toutefois une lacune importante. Il ne suffit pas trouver une courbe qui passe près d'un point : il faut être en mesure d'identifier une valeur de glissement réelle. Il faut également que le glissement nominal soit le même peu importe la spécification qui est étudiée. Par exemple, le glissement nominal du couple ne pourrait pas être différent de celui du courant puisqu'on ne pourrait pas spécifier les conditions nominales correctement.

La méthode proposée dans ce mémoire est donc de calculer l'erreur sur les spécifications nominales du couple, du courant et du facteur de puissance pour les valeurs de glissement autour du glissement nominal. Un nouveau glissement nominal, qui se trouve à plus ou moins 2% de la valeur donnée par le manufacturier, est calculé en se basant sur l'erreur cumulative la plus petite.

2.4 Algorithme d'optimisation

L'algorithme d'optimisation permet de calculer un ensemble de paramètres qui minimise la fonction objectif. L'initialisation se fait à partir des spécifications du manufacturier et de relations empiriques. L'algorithme en tant que tel se base sur l'estimation initiale et calcule ensuite de manière itérative une solution qui diminue l'erreur au fur et à mesure. Bien que plusieurs algorithmes soient présentés dans la littérature, celui qui est utilisé pour le CGV est l'algorithme PSO, soit *Particle Swarm Optimization*. Il est présenté dans les prochaines sections.

2.4.1 Initialisation

L'initialisation permet de fournir une première estimation des paramètres. Lorsque les paramètres sont totalement inconnus, il faut se baser sur les spécifications du manufacturier ou encore sur des données empiriques afin d'établir des relations générales entre les paramètres. Chacun des paramètres est initialisé de la manière suivante :

$$R_r = \frac{3V_{in}^2 s_n}{P_{mfl}} \quad (2.20)$$

$$R_{r1} = R_r \quad (2.21)$$

$$R_{r2} = 5R_r \quad (2.22)$$

$$R_s = 1.5R_r \quad (2.23)$$

$$X_m = 0.8 \frac{3V_{in}^2}{Q_{fl}} \quad (2.24)$$

$$X_{sd} = 0.05X_m \quad (2.25)$$

$$X_{1d} = 1.2X_{sd} \quad (2.26)$$

$$X_{2d} = 0.5X_{sd} \quad (2.27)$$

$$R_{fe} = \frac{V_{in}^2}{P_{core}} \quad (2.28)$$

$$I_{sat} = 3 \quad (2.29)$$

$$PS = 0.5 \quad (2.30)$$

avec

$$P_{fl} = 3V_{in}I_n \cos(\phi) \quad (2.31)$$

$$P_{mfl} = P_{fl}\eta_{100} = P_n \quad (2.32)$$

$$P_{core} = P_{fl} - P_{mfl} \quad (2.33)$$

$$Q_{fl} = 3V_{in}I_n \sqrt{(1 - \cos^2(\phi))} \quad (2.34)$$

où

P_{fl} : puissance active à pleine charge [W]

P_{mfl} : puissance mécanique à pleine charge [W]

P_{core} : pertes du noyau [W]

Q_{fl} : puissance réactive à pleine charge [VAR]

Étant donné que les caractéristiques du moteur changent énormément en fonction des données nominales de tension, de courant, de couple et de puissance, l'estimation initiale n'est pas nécessairement très précise. Elle sert tout de même de point d'initialisation pour l'algorithme d'optimisation.

2.4.2 PSO

PSO est un algorithme qui s'inspire de la nature, plus précisément du déplacement des oiseaux et des poissons lorsqu'ils sont en groupe. L'algorithme a été développé en 1995 par Kennedy et Eberhart [27]. L'idée générale est d'explorer un espace solution à l'aide de plusieurs individus et de trouver une bonne solution en forçant un déplacement dans l'espace solution selon la meilleure position trouvée par chaque individu et par le groupe en entier.

Cet algorithme est une métaheuristique, un algorithme d'optimisation qui permet de trouver une solution à des problèmes complexes et non-linéaires. L'algorithme fait intervenir une certaine part de hasard en cherchant de manière aléatoire l'espace solution afin d'y trouver une solution.

Le but de l'algorithme est d'identifier les paramètres qui minimisent la fonction objectif $f(x)$ (2.16). Chaque individu, aussi appelé particule, est donc un ensemble de paramètres x (2.15(2.15)). De manière équivalente, une particule correspond à une position x_i dans l'espace solution. Mathématiquement, l'algorithme peut être décrit de la manière suivante :

$$v_i = c_{mémoire} v_i + c_{exploration} (P_{best,i} - x_i) + c_{exploitation} (G_{best} - x_i) \quad (2.35)$$

$$x_i = x_i + v_i \quad (2.36)$$

où

x_i : ième particule

v_i : déplacement de la i ème particule

$P_{best,i}$: meilleure position personnelle trouvée par la i ème particule

G_{best} : meilleure position globale trouvée par le groupe de particules

$c_{mémoire}$: coefficient de mémoire

$c_{exploration}$: coefficient d'exploration

$c_{exploitation}$: coefficient d'exploitation

Lors de la première itération, chaque particule x_i est initialisée à partir de l'estimation initiale. Plus précisément, deux paramètres parmi les 10 paramètres qui composent le vecteur x sont modifiés au hasard afin de distribuer les particules dans l'espace solution.

$$x_i = x0 \quad (2.37)$$

avec

$$\begin{aligned} x_i(rnd_1) &= x0(rnd_1) [1 + normrnd(0,1)] \\ x_i(rnd_2) &= x0(rnd_2) [1 + normrnd(0,1)] \end{aligned}$$

où

rnd_i : nombre aléatoire d'une distribution uniforme entre 1 et 10

$normrnd(0,1)$: nombre aléatoire d'une distribution normale de moyenne 0 et d'écart-type 1

Le déplacement v_i est aussi initialisé à ce moment alors que la position optimale de chaque particule et du groupe en entier, soit l'erreur la plus petite, est calculée grâce à la fonction objectif $f(x)$. Pour les itérations subséquentes, la prochaine position des particules est obtenue à l'aide des équations (2.35) et (2.36) alors que les nouvelles valeurs pour $P_{best,i}$ et G_{best} sont calculées à nouveau. L'algorithme se termine lorsque l'un des critères d'arrêt est atteint. Le nombre d'itérations k et le nombre de particules n influencent grandement le temps de calcul et la convergence de l'algorithme. Cet aspect est détaillé davantage au chapitre 3.

2.4.3 Recherche adaptative

À chaque itération, la nouvelle position de la particule est définie en fonctions de trois critères distincts : la mémoire, l'exploration et l'exploitation. La mémoire correspond au dernier déplacement effectué par la particule, soit son inertie. L'exploration est ce qui permet à la particule d'explorer l'espace solution autour de sa propre position. L'exploitation est ce qui permet à toutes les particules de converger vers la meilleure solution qui a été trouvée par le groupe.

Il est souhaitable de pouvoir donner plus ou moins d'importance à chacun de ces critères au fur et à mesure que le nombre d'itérations augmente. Effectivement, il est plus utile de laisser les particules explorer l'espace solution au début afin de couvrir la plus grande superficie possible, alors qu'il est préférable de les regrouper autour de la solution optimale ensuite. La formule qui décrit la variation de chacun des coefficients s'exprime de la manière suivante :

$$c = c_{initial} + (c_{final} - c_{initial}) \frac{k}{k_{max}} \quad (2.38)$$

où

c : coefficient de mémoire, d'exploration ou d'exploitation

$c_{initial}$: valeur initiale du coefficient

c_{final} : valeur finale du coefficient

k : itération

k_{max} : nombre d'itérations maximal

Pour obtenir l'effet souhaité, il faut que le coefficient d'exploration soit d'abord plus grand que celui d'exploitation, puis le contraire. Le coefficient de mémoire doit également diminuer progressivement avec le nombre d'itérations. Les relations suivantes peuvent donc être établies :

$$c_{exploration_{initial}} > c_{exploitation_{initial}} \quad (2.39)$$

$$c_{exploration_{final}} < c_{exploitation_{final}} \quad (2.40)$$

$$c_{mémoire} \downarrow \quad (2.41)$$

$$c_{exploration} \downarrow \quad (2.42)$$

$$c_{exploitation} \uparrow \quad (2.43)$$

Les valeurs numériques qui sont utilisées pour chacun des coefficients sont présentées au Tableau 2.1. Ces valeurs ont été obtenues de manière expérimentale afin d'accélérer la convergence et elles n'ont pas été changées par rapport à [3] étant donné que les résultats étaient satisfaisants.

Tableau 2.1 : Valeurs numériques des coefficients de l'optimisation

Coefficient	Mémoire	Exploration	Exploitation
Initial	0.9	2.5	0.5
Final	0.4	0.5	1.0

Les coefficients d'exploration et d'exploitation sont également multipliés par un nombre aléatoire compris entre 0 et 1, ce qui ajoute une autre composante arbitraire. L'exploration de l'espace solution est améliorée, mais un résultat différent est retourné à chaque appel de l'algorithme d'optimisation en contrepartie. Ainsi, il est intéressant de faire quelques itérations du calcul des paramètres afin de s'assurer que la solution trouvée est réellement optimale. Encore une fois, cet aspect est approfondi au chapitre suivant.

2.4.4 Critères d'arrêt

L'algorithme possède plusieurs critères d'arrêt, que ce soit l'atteinte d'une certaine erreur maximale ou encore du nombre d'itérations maximal. Dans le cas où les solutions convergent toutes vers les mêmes paramètres, l'algorithme est aussi arrêté.

Le critère de convergence principal est celui relié à l'erreur sur les spécifications. Il existe plusieurs façons d'évaluer l'erreur entre les spécifications pour une fonction à multi-objectifs comme celle définie dans ce mémoire. Chaque évaluation de $f(x)$ retourne six erreurs. Quelle est l'importance relative de ces erreurs entre elles? Est-il préférable d'obtenir la somme des erreurs la plus petite ou encore l'erreur maximale la plus petite, peu importe laquelle?

Dans le cas présent, il est considéré que les erreurs sont aussi importantes les unes que les autres. Il n'est donc pas utile de donner un poids à chaque erreur dans le calcul de l'erreur totale puisque les erreurs sont déjà relatives. Selon les essais expérimentaux effectués, l'algorithme a tendance à délaisser une spécification et à minimiser les autres lorsque la somme des erreurs est considérée. Cela résulte en un modèle qui n'est pas très précis dans son ensemble. En priorisant l'erreur maximale la plus petite, le modèle respecte chacune des spécifications, même si l'erreur totale est potentiellement légèrement plus élevée. Le critère de convergence qui est utilisé pour le CGV est donc de prendre l'erreur la plus élevée en valeur absolue. Ainsi, pour chaque évaluation de $f(x)$, une seule erreur est retournée et comparée avec les autres afin de déterminer l'ensemble de paramètres optimal. L'erreur maximale est fixée à 2%.

Le critère d'arrêt basé sur le nombre d'itération permet d'arrêter l'algorithme dans le cas où la convergence n'est pas atteinte rapidement. Le nombre d'itérations maximal est également utilisé pour modifier les coefficients de mémoire, d'exploration et d'exploitation. Aussi, plus le nombre d'itérations est élevé, plus l'algorithme prend de temps à s'exécuter.

Dans le cas où les particules convergent vers un optimum local, c'est-à-dire que l'erreur pour chacune des particules est pratiquement la même, l'algorithme est aussi arrêté. Cela permet de mettre fin au calcul avant l'atteinte du nombre maximal d'itérations lorsqu'une solution satisfaisante ne sera visiblement pas trouvée.

2.4.5 Bornes dynamiques

Une des notions qui n'a pas été discutée jusqu'à présent et qui est primordiale dans l'atteinte d'une solution optimale est celle des bornes sur les paramètres. Ces contraintes permettent de définir l'espace solution de l'algorithme, mais aussi d'établir des relations entre deux paramètres.

Tout d'abord, la taille de l'espace solution est très importante. Dans le cas où l'espace solution est trop petit, l'algorithme peut facilement trouver un minimum local, mais rien n'assure qu'il s'agit aussi du minimum global. Au contraire, un espace solution trop grand a plus de chance de contenir le minimum global, mais l'algorithme ne sera pas nécessairement en mesure de le trouver. Il doit y avoir un compromis entre la grosseur de l'espace solution et la probabilité de trouver un optimum dans cet espace.

Lors de l'initialisation, la borne inférieure LB est fixée à 0.02 fois la valeur du paramètre alors que la borne supérieure UB est fixée à 10 fois la valeur de ce même paramètre. Ces valeurs sont posées de manière arbitraire afin de donner une certaine marge de manœuvre autour de l'estimation initiale qui n'est pas toujours précise. Les bornes inférieures et supérieures sur I_{sat} sont fixées à 1 et à la valeur du courant de démarrage en pu respectivement, alors que les bornes sur SAT sont fixées à 0 et 1. Une borne inférieure plus petite sur le courant de saturation I_{sat} indiquerait que la machine est en saturation même en fonctionnement nominal, alors que le courant de démarrage correspond au courant maximal de la machine.

Les bornes permettent aussi de ramener les paramètres qui tentent de sortir des limites, par exemple lors de l'initialisation ou lorsque les particules se déplacent dans l'espace solution. Cependant, il arrive parfois qu'un paramètre se retrouve coincé près d'une borne, c'est-à-dire que la solution optimale se trouve proche d'une limite mais qu'une particule n'est pas en mesure d'aller explorer autour de cette position afin de trouver une solution encore meilleure. C'est à ce moment que l'utilisation de bornes dynamiques peut s'avérer utile.

Lorsque la solution optimale G_{best} est à plus ou moins 10% d'une borne sur un paramètre, la limite sur ce paramètre est augmentée afin de permettre aux particules d'explorer au-delà de l'espace solution initial. La capacité de l'algorithme à trouver un minimum n'est pas affectée étant donné qu'un seul paramètre et non l'ensemble des paramètres est modifié. Étant donné qu'il est difficile de définir à l'avance l'espace solution pour des machines de spécifications différentes, cette méthode permet de l'agrandir seulement en cas de besoin.

Finalement, les bornes sont également utiles pour forcer une relation entre les paramètres, notamment entre la première et la deuxième branche du rotor. Étant donné que R_{r2} est plus grand que R_{r1} et que X_{1d} est plus grand que X_{2d} par construction, le vecteur x est redéfinie par rapport à la déclaration initiale faite à l'équation (2.15).

$$x = [R_{fe}, R_s, R_{r1}, R_{r2} - R_{r1}, X_{sd}, X_m, X_{1d} - X_{2d}, X_{2d}, I_{sat}, PS] \quad (2.44)$$

Ainsi, en prenant la valeur absolue des paramètres, on s'assure que les relations définies précédemment sont toujours vraies.

$$R_{r2} - R_{r1} \geq 0 \Rightarrow R_{r2} \geq R_{r1} \quad (2.45)$$

$$X_{1d} - X_{2d} \geq 0 \Rightarrow X_{1d} \geq X_{2d} \quad (2.46)$$

En résumé, l'algorithme d'optimisation permet de passer d'une estimation initiale des paramètres à une solution qui minimise l'erreur maximale sur les spécifications de la machine de manière itérative. Un schéma de l'algorithme PSO comprenant la recherche adaptative, les critères d'arrêts ainsi que les bornes dynamiques est présenté à la Figure 2.5.

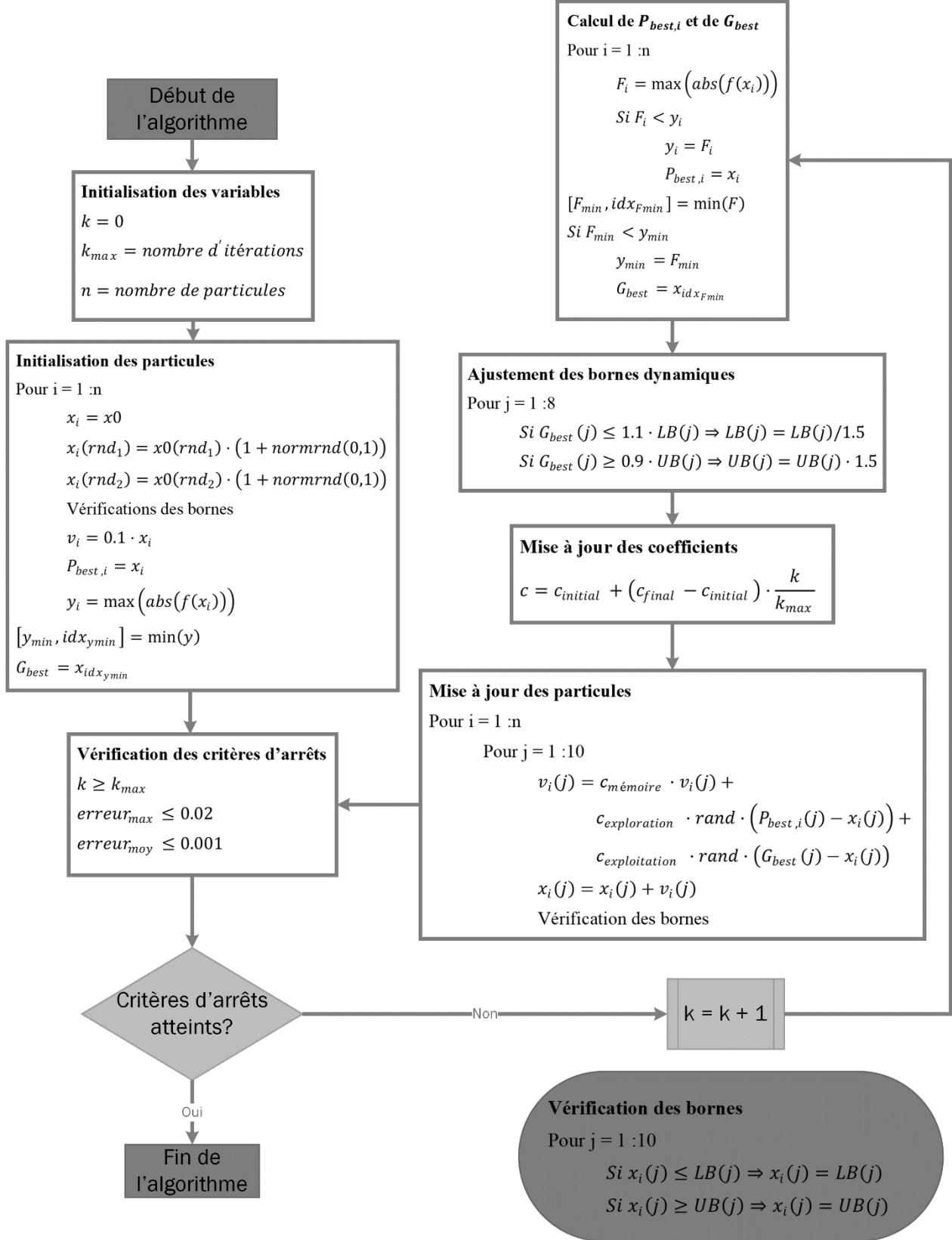


Figure 2.5 : Schéma de l'algorithme PSO

CHAPITRE 3 MÉTHODE ANALYTIQUE

À partir des équations analytiques qui décrivent le comportement de la machine asynchrone définie au chapitre précédent, il est possible de construire les courbes du couple, du courant et du facteur de puissance en fonction de la vitesse et ainsi d'évaluer l'erreur sur les spécifications de manière analytique. Une banque de donnée de 115 machines provenant de l'IREQ et une autre banque de données de 82 machines provenant du manufacturier ABB ont été analysées afin de déterminer les performances du modèle et du CGV. Le code MATLAB est d'abord introduit, puis les résultats pour les deux banques de données sont mis de l'avant et comparés avec le calculateur précédent. Finalement, des exemples de cas où la convergence est obtenue de même que des cas où la convergence n'est pas obtenue sont présentés.

3.1 Code MATLAB

Le langage de programmation MATLAB est utilisé afin de déterminer les paramètres recherchés. D'une part, il s'agit d'un langage simple et convivial à utiliser. De nombreuses librairies et fonctions prédéfinies sont fournies et l'environnement de programmation est adapté à la visualisation des données. D'une autre part, l'ancien calculateur développé par François Gauthier était déjà codé dans ce langage. Le choix du langage de programmation a donc été évident.

De manière générale, le code prend en entrée les spécifications de la machine et retourne les paramètres du modèle à cage double ainsi que les courbes du couple, du courant et du facteur de puissance en fonction de la vitesse. La structure du code est présentée à la Figure 3.1, alors qu'un survol des différentes fonctions appelées par la fonction principale *calculateParam.m* est effectué dans les paragraphes suivants. Un guide pour utiliser le code a également été réalisé. Ce guide contient une description plus approfondie du code et des fonctions.

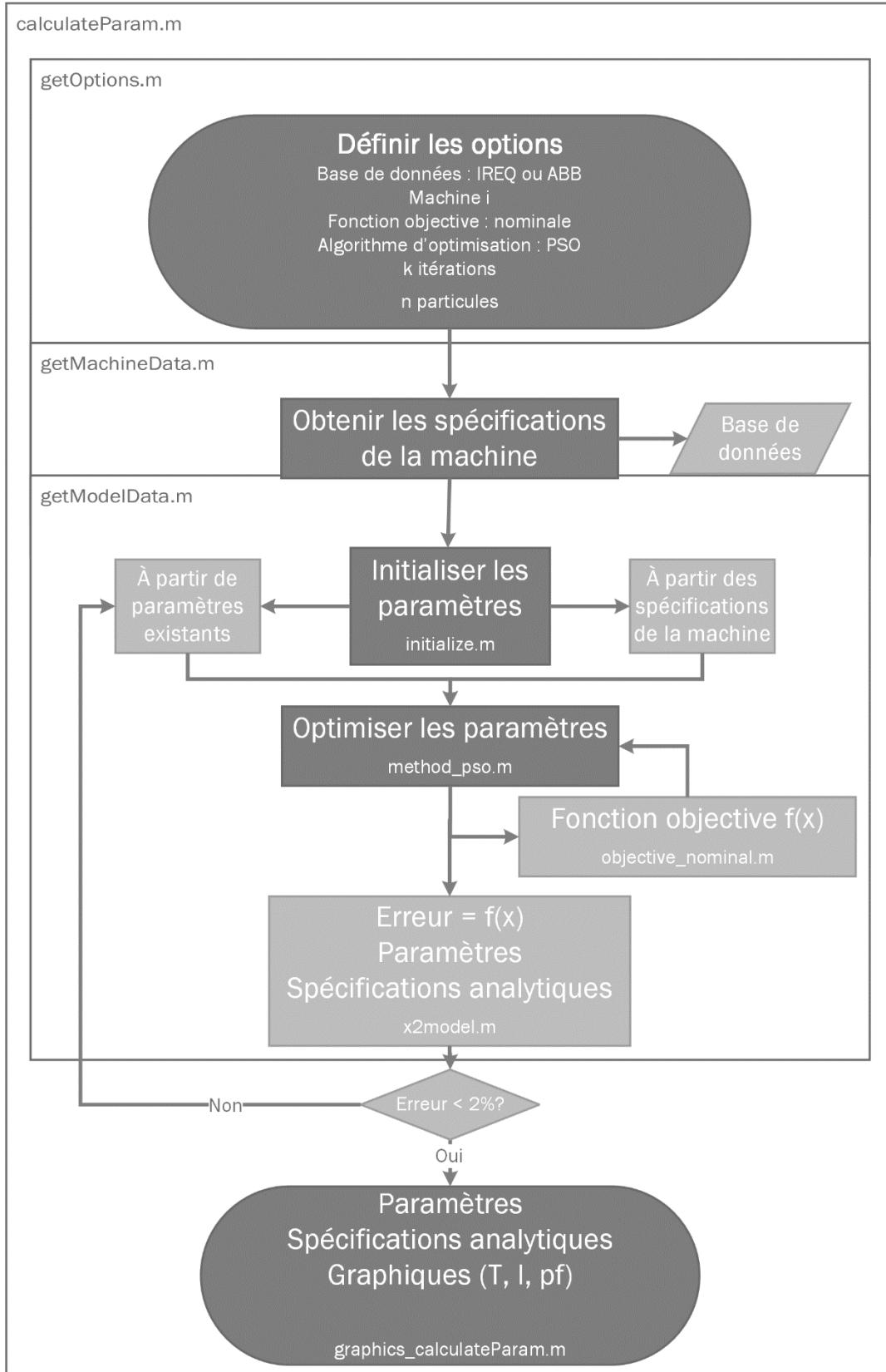


Figure 3.1 : Structure du code MATLAB

Tout d'abord, les options du CGV, notamment au niveau de la base de données, de la fonction objectif et de l'algorithme d'optimisation sont définies dans le fichier *getOptions.m*.

Le nom du fichier dans lequel les données des machines sont enregistrées de même que le numéro de la machine à l'étude est donné par l'utilisateur. Les données sont enregistrées en format .xml dans le cas des données provenant de l'IREQ, alors qu'elles sont au format .xlsx pour les données de ABB. La base de données de l'IREQ est d'ailleurs incluse directement dans le modèle de la machine asynchrone dans les versions plus récentes de EMTP. Il serait possible de créer sa propre base de données en respectant le format de l'un ou l'autre de ces fichiers. L'accès à la base de données se fait par *getMachineData.m*.

La fonction objectif avec le calcul des spécifications nominales est définie au chapitre précédent. Encore une fois, il serait possible d'utiliser une fonction objectif différente, par exemple une fonction qui se base sur la puissance nominale plutôt que sur les autres spécifications afin de comparer les résultats. La fonction objectif est définie dans le fichier *objective_nominal.m*.

Pour ce qui est de l'algorithme d'optimisation, le CGV utilise PSO. Évidemment, un algorithme différent pourrait être utilisé. L'algorithme se trouve dans *method_pso.m*.

Le nombre d'itérations et le nombre de particules définissent la rapidité de l'algorithme PSO. Plus ces nombres sont grands et plus le calcul prend du temps à s'effectuer, pouvant passer de quelques secondes à plusieurs dizaines de minutes par machine. De plus, étant donné la nature aléatoire de l'algorithme, il n'est pas assuré qu'une solution satisfaisante sera trouvée du premier coup. Ainsi, il est préférable de diminuer la valeur de ces paramètres et d'effectuer quelques calculs plutôt qu'un seul. Cela permet également d'utiliser les paramètres du calcul précédent lors de l'initialisation, ce qui donne généralement de meilleurs résultats que l'estimation à partir des spécifications de la machine. L'initialisation des paramètres se fait d'ailleurs dans le fichier *initialize.m*.

Aussi, il est préférable de faire un premier calcul rapide des paramètres puis d'augmenter progressivement la précision jusqu'à l'obtention d'une convergence. Cela permet d'optimiser le temps de calcul total. Pour donner une idée de grandeur, quelques paires de valeurs pour le nombre d'itérations et de particules de l'algorithme d'optimisation PSO et le temps de calcul correspondant sont présentées au Tableau 3.1.

Tableau 3.1 : Temps de calcul approximatif selon les paramètres de l'optimisation

Nombre d'itérations k	Nombre de particules n	Temps de calcul [s]
50	10	10
100	50	60
200	100	180
500	200	600

Ainsi, faire quelques calculs avec des paramètres faibles permet de trouver une première solution acceptable, puis refaire des calculs supplémentaires avec des paramètres plus forts donne une précision maximale.

La fonction *x2model.m* est utilisée afin de passer du vecteur de paramètres avec les relations entre les cages du rotor, soit le vecteur x , aux paramètres électriques et aux spécifications analytiques de la machine. Cette fonction est utilisée à chaque fois que la fonction objectif est évaluée sur une particule lors de l'optimisation.

Finalement, dans le cas où l'erreur est satisfaisante, les paramètres et les spécifications du modèle analytique sont renvoyées par le CGV. Les courbes du couple, du courant et de facteur de puissance en fonction de la vitesse peuvent ensuite être affichés à l'aide de la fonction *graphics_calculateParam.m*. Il est possible d'afficher les graphiques en pu ou en valeurs nominales à la convenance de l'utilisateur.

3.2 Performance du CGV

Les bases de données provenant de l'IREQ et de ABB ont été analysées avec le CGV. La base de données de l'IREQ avait été utilisée pour étudier les performances du calculateur dans [2], ce qui permet de faire une comparaison directe entre les deux. La base de données ABB a été ajoutée afin de valider que le CGV fonctionne pour un plus grand nombre de machines et que des résultats

similaires sont obtenus. De plus, la provenance des données de la base de données de l'IREQ est inconnue, c'est-à-dire que ce n'est pas clair s'il s'agit de spécifications provenant de la plaque signalétique ou encore d'essais effectuées sur les machines. Les données ABB proviennent directement de la fiche technique du manufacturier, ce qui enlève toute ambiguïté.

Les performances du CGV sont présentées au Tableau 3.2. Pour rappel, le critère de convergence considéré est une erreur maximale sur l'une ou l'autre des spécifications de 2% au maximum. De plus, le nouveau couple nominal calculé se trouve à plus ou moins 2% de la valeur donnée par le manufacturier.

Tableau 3.2 : Performance du CGV

Calculateur	Base de données IREQ 115 machines	Base de données ABB 82 machines
CGV	109 convergences 6 divergences 94.8%	78 convergences 4 divergences 95.1%
Calculateur précédent	66 convergences 49 divergences 57.4%	-

Pour la base de données de l'IREQ, on observe que le taux de convergence passe de 57.4% à 94.8%, ce qui représente une nette amélioration. Un taux de convergence pratiquement identique est obtenu avec la base de données ABB, soit 95.1%. Cela permet de confirmer que le CGV possède un meilleur taux de convergence que le calculateur précédent pour un large échantillon de machines.

Les erreurs en fonction du numéro de la machine dans la base de données sont présentées à la Figure 3.2. Il est à noter que le critère d'arrêt lors de l'optimisation a été diminué à 0.2% afin d'obtenir une plus grande exactitude.

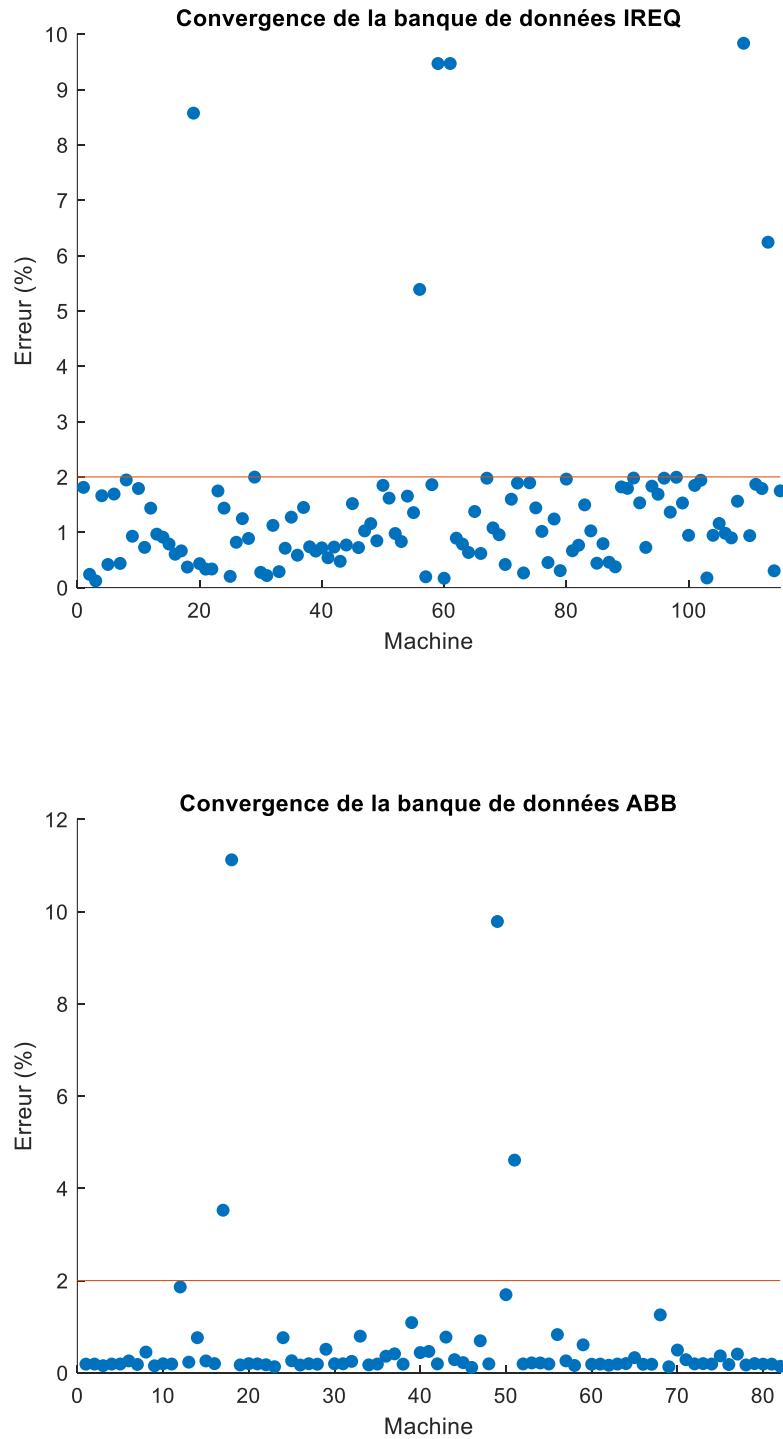


Figure 3.2 : Erreur maximale pour chaque machine de la base de données

3.3 Analyse de convergence

Quelques exemples de convergences et de divergences provenant de la base de données ABB sont présentés plus bas afin d'illustrer les performances du CGV. Les spécifications du manufacturier ainsi que les paramètres calculés sont indiqués pour chacune des machines. Les courbes du couple, du courant et du facteur de puissance en fonction de la vitesse de même que l'erreur sur chacune des spécifications sont également présentées. Différentes techniques afin de trouver une solution aux cas divergents sont finalement avancées.

3.3.1 Cas de convergence

Un premier exemple de convergence est donné pour la machine ABB – M2BAX 71MA 2, soit la machine 1 dans la base de données ABB. Les spécifications ainsi que les paramètres de la machine sont présentés au Tableau 3.3 et au Tableau 3.4 respectivement, alors que les courbes analytiques sont illustrées à la Figure 3.3. Finalement, les erreurs sur chacune des spécifications sont rassemblées au Tableau 3.5.

Tableau 3.3 : Spécifications de la machine 1

V_n [V]	P_n [W]	T_n [Nm]	T_{st} [pu]	T_{br} [pu]	I_n [A]	I_{st} [pu]	pf_n	N_n [RPM]	N_s [RPM]
400	370	1.26	2.8	3.4	0.91	5.1	0.79	2807	3000

Tableau 3.4 : Paramètres de la machine 1

R_{fe} [Ω]	R_s [Ω]	R_{r1} [Ω]	R_{r2} [Ω]	X_{sd} [Ω]	X_m [Ω]	X_{1d} [Ω]	X_{2d} [Ω]	I_{sat} [pu]	PS
2018.8	10.88	36.56	37.18	18.18	450.19	32.24	22.13	1.0384	0.3481

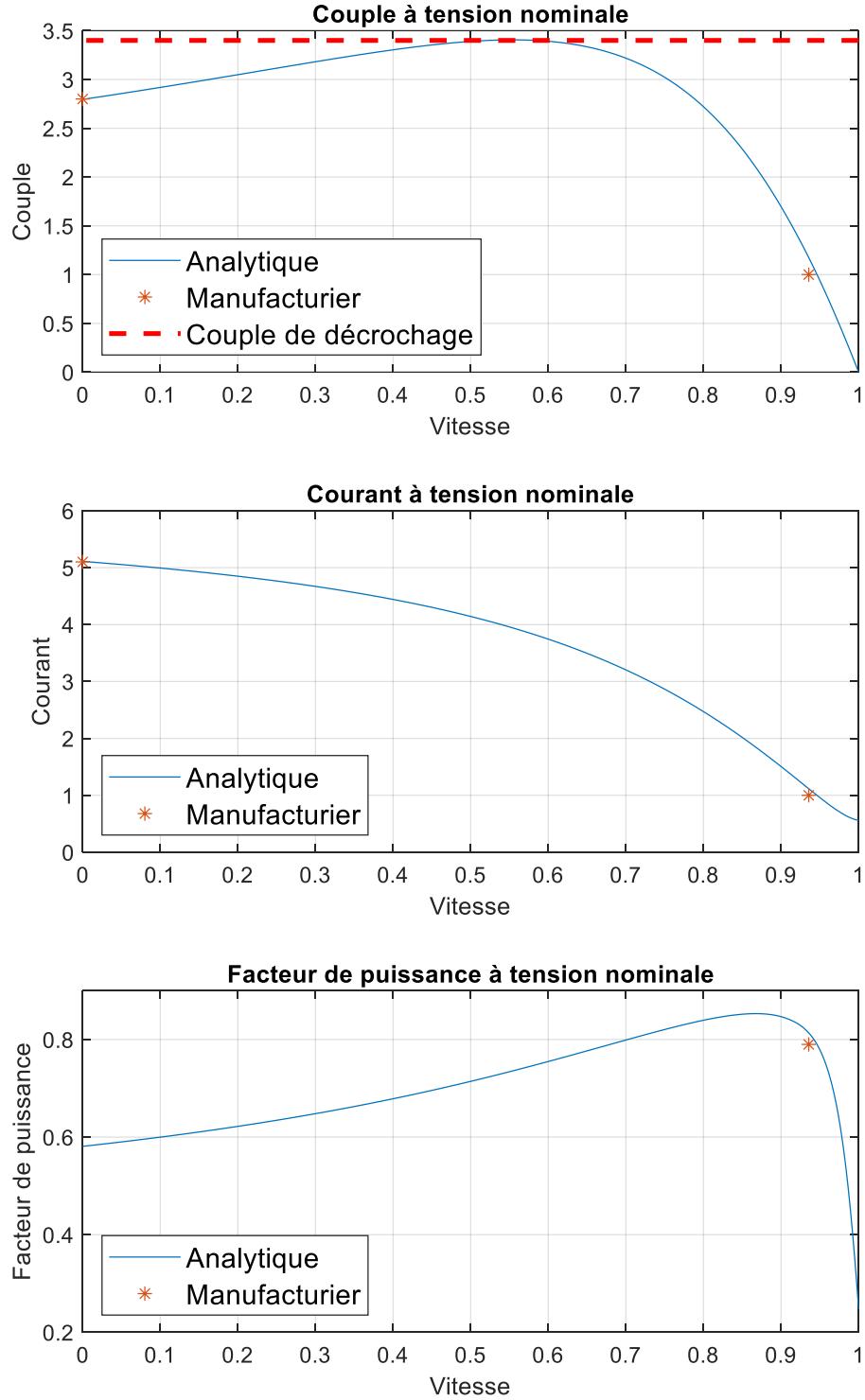


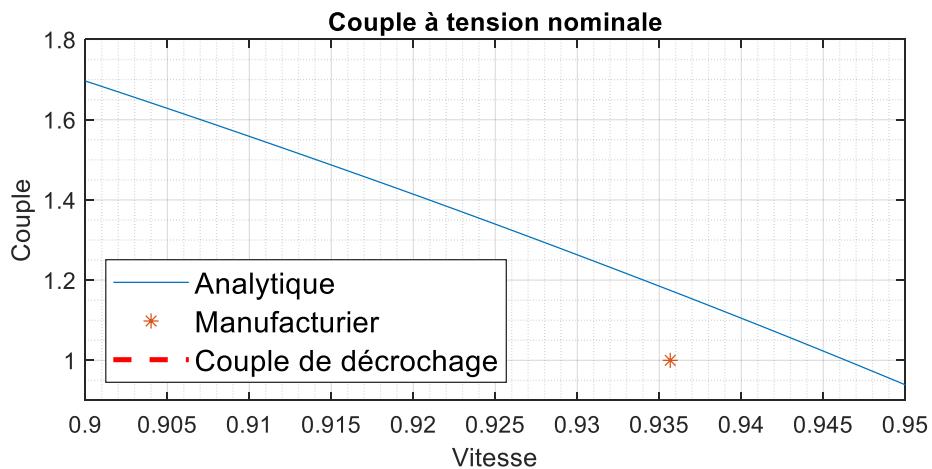
Figure 3.3 : Courbes analytiques de la machine 1

Tableau 3.5 : Erreurs sur les spécifications de la machine 1

Spécification	T_n	T_{st}	T_{br}	I_n	I_{st}	pf_n
Erreur (%)	0.15	-0.16	0.17	-0.06	0.19	-0.01

On observe que les courbes suivent parfaitement les spécifications de la machine, ce qui se reflète dans les erreurs qui sont toutes inférieures à 0.2%. Effectivement, la courbe analytique en bleu passe par les points du manufacturier en orange dans chacun des graphiques. On remarque aussi que le glissement nominal est légèrement décalé par rapport à la valeur donnée par le manufacturier. En agrandissant le graphique entre 0.90 pu et 0.95 pu sur l'axe de la vitesse, on peut s'assurer que le nouveau glissement nominal qui a été calculé pour cette machine respecte la limite de 2% qui a été fixée. Les graphiques avec un agrandissement autour du glissement nominal sont présentés à la Figure 3.4.

Pour chacune des courbes, on observe que la spécification nominale est obtenue pour une vitesse d'environ 0.946 pu, soit la vitesse pour laquelle la valeur analytique est égale à celle du manufacturier. En comparaison avec la vitesse nominale donnée par le manufacturier, on trouve une différence de seulement 1.14% telle que calculée au chapitre précédent à l'équation (2.19).



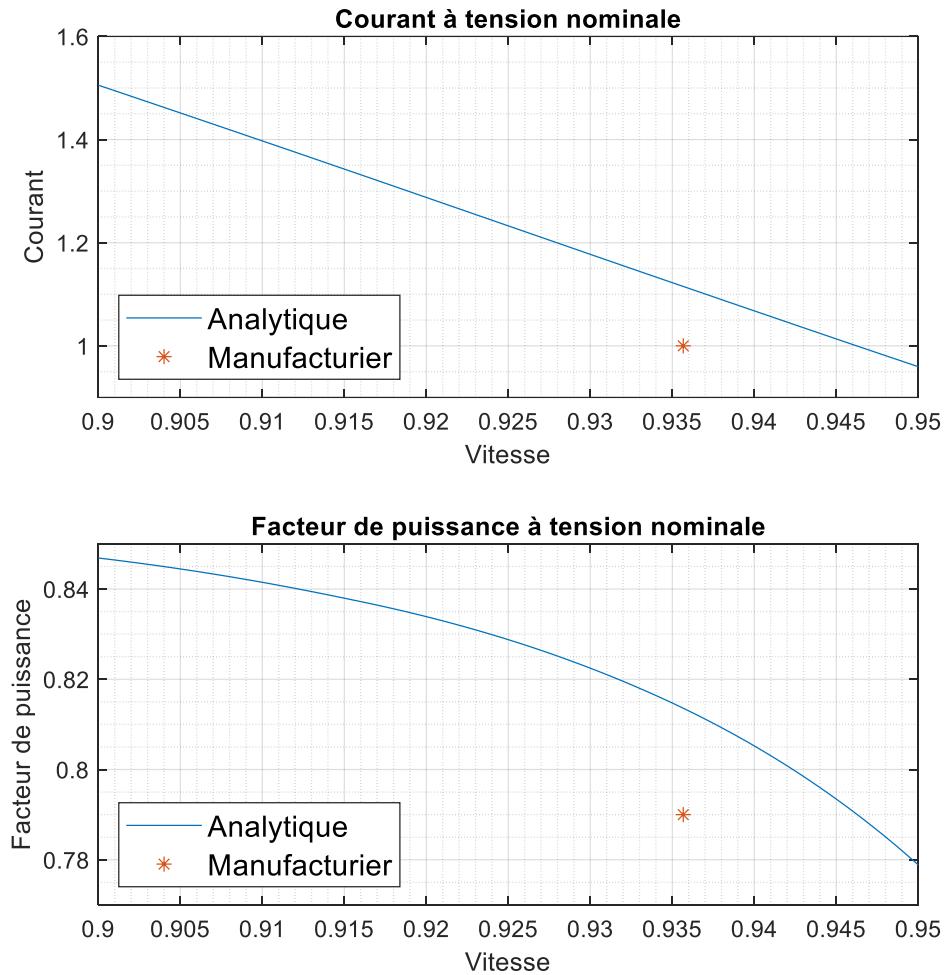


Figure 3.4 : Agrandissement des courbes analytiques de la machine 1

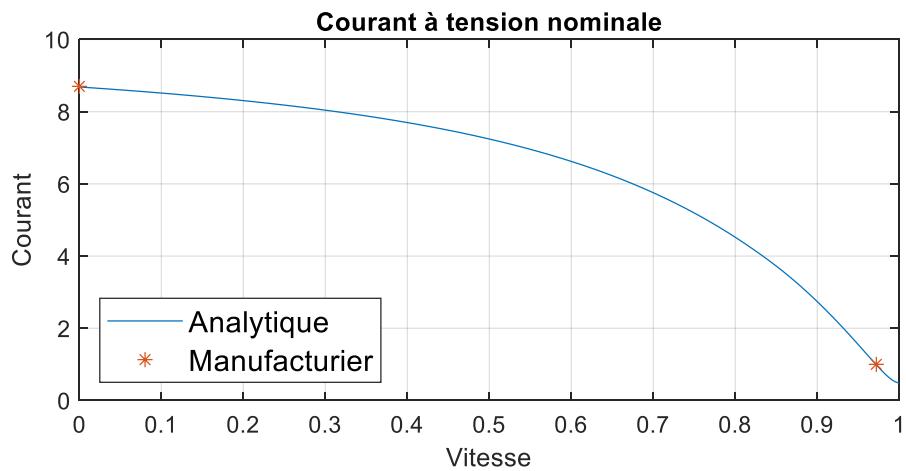
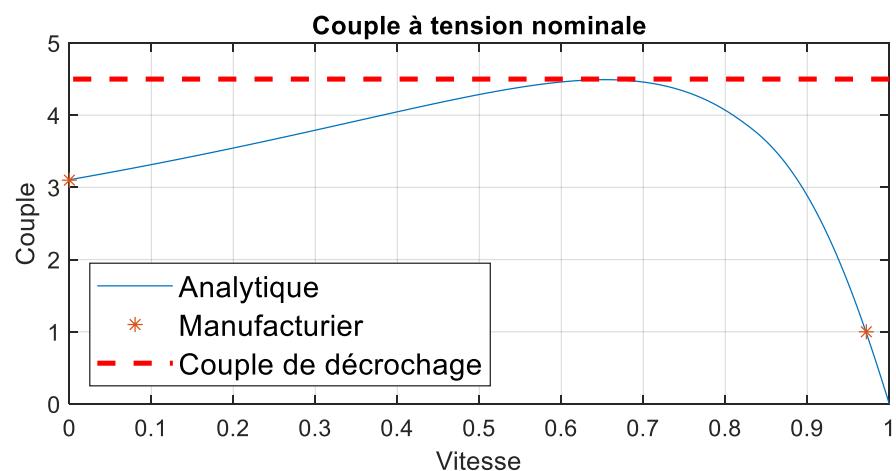
Un deuxième exemple de convergence est donné pour la machine ABB – M2BAX 132SB 2, soit la machine 10 dans la banque de données ABB. Le Tableau 3.6 contient les spécifications de la machine, alors que le Tableau 3.7 regroupe les paramètres calculés. Les courbes analytiques sont présentées à la Figure 3.5. Finalement, le Tableau 3.8 résume l'erreur sur chacune des spécifications.

Tableau 3.6 : Spécifications de la machine 10

V_n [V]	P_n [W]	T_n [Nm]	T_{st} [pu]	T_{br} [pu]	I_n [A]	I_{st} [pu]	pf_n	N_n [RPM]	N_s [RPM]
400	7500	24.57	3.1	4.5	14.5	8.7	0.84	2916	3000

Tableau 3.7 : Paramètres de la machine 10

R_{fe} [Ω]	R_s [Ω]	R_{r1} [Ω]	R_{r2} [Ω]	X_{sd} [Ω]	X_m [Ω]	X_{1d} [Ω]	X_{2d} [Ω]	I_{sat} [pu]	PS
336.4	0.41	0.56	40.42	0.25	32.80	1.74	0.83	3.8194	0.3916



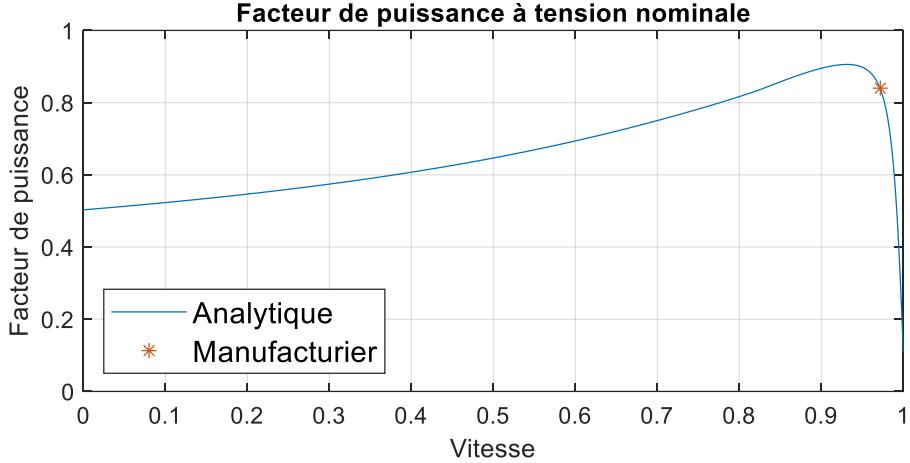


Figure 3.5 : Courbes analytiques de la machine 10

Tableau 3.8 : Erreurs sur les spécifications de la machine 10

Spécification	T_n	T_{st}	T_{br}	I_n	I_{st}	pf_n
Erreur (%)	0.03	0.18	-0.18	-0.10	-0.20	0.06

On observe que les erreurs sont encore une fois inférieures à 0.2% pour chacune des spécifications. Cette fois-ci, le nouveau glissement nominal qui a été calculé est pratiquement identique à celui fourni par le manufacturier, soit 0.0285 par rapport à 0.0280. Dans les deux cas présentés plus haut, la puissance nominale de la machine est inférieure à 10 kW, ce qui est une puissance relativement faible pour des moteurs industriels. Le calculateur dans [2] avait notamment de la difficulté à trouver des solutions pour les machines de faible puissance, ce qui n'est pas le cas du CGV. Il s'agit donc d'une amélioration notable par rapport au calculateur précédent.

3.3.2 Cas de divergence

Lorsque le CGV n'arrive pas à trouver une solution avec une erreur inférieure à 2%, le cas est considéré comme divergent. Une ou plusieurs des spécifications ne sont pas respectées, ce qui veut dire que les courbes ne représentent pas adéquatement le comportement de la machine. Il existe peut-être une solution à ce problème.

Avant de considérer qu'un cas est réellement divergent, il peut être intéressant de vérifier qu'il ne s'agit pas simplement d'une erreur de convergence au niveau de l'algorithme. Parfois, le CGV retourne la même erreur minimale pour chaque nouveau calcul des paramètres peu importe la valeur des paramètres k et n , c'est-à-dire que l'algorithme est incapable de se sortir d'un minimum local. Au lieu de refaire un calcul en se basant sur les paramètres calculés lors de l'itération précédente, il est alors préférable de refaire le calcul à partir du début, soit en utilisant les spécifications du manufacturier pour obtenir un premier estimé des paramètres. L'algorithme peut maintenant tenter de trouver une meilleure solution sans être coincé dans le minimum local qui l'empêchait d'explorer ailleurs dans l'espace solution. En d'autres mots, on donne une seconde chance au CGV de trouver une solution puisque la première fois n'a pas donné un résultat satisfaisant. Cela peut survenir en raison de la nature aléatoire de l'algorithme qui ne garantit en aucun temps que la meilleure solution sera trouvée.

Une autre observation intéressante est au niveau de la relation entre le couple de démarrage et le couple de décrochage. Effectivement, il est possible d'observer que la plupart des machines ont un couple de décrochage plus élevé que le couple de démarrage, alors que quelques-unes ont un couple de décrochage égal ou même inférieur au couple de démarrage. La classification des machines en fonction du couple est présentée au Tableau 3.9.

Tableau 3.9 : Relation entre le couple de décrochage et le couple de démarrage

Condition	$T_{br} > T_{st}$	$T_{br} = T_{st}$	$T_{br} < T_{st}$
Nombre de machines	73 machines (17, 45, 46, 59, 82)	5 machines (18, 39, 50, 51)	4 machines

Parmi les quatre cas divergents, soit les machines 17, 18, 49 et 51, trois d'entre elles possèdent un couple de décrochage égal ou inférieur au couple de démarrage. La machine 49 possède quant à elle un couple de démarrage de 2.9 pu et un couple de décrochage de 3.0 pu, ce qui est aussi relativement proche. Cela indique donc qu'il y a potentiellement un problème au niveau de l'évaluation du couple de décrochage lorsque celui-ci est similaire au couple de démarrage.

La méthode décrite au chapitre précédent pour évaluer le couple de décrochage est relativement simpliste. Les cas où le couple de décrochage est inférieur au couple de démarrage sont séparés du reste, sans qu'une attention particulière ne soit portée au glissement correspondant. Selon la classe NEMA de la machine [28], le couple peut avoir une allure décroissante monotone (classe C ou D), ou encore une courbe qui ressemble plus aux exemples précédents (classes A ou B). En identifiant la classe de la machine, il serait possible de définir des conditions plus strictes au niveau du glissement associé au couple de décrochage et ainsi de réduire le nombre de cas de divergence. Or, la classe n'est pas nécessairement connue, comme c'est le cas pour les machines de la base de données de l'IREQ où aucune information n'est fournie à ce sujet. Cette approche est donc difficilement utilisable en pratique.

Pour les quatre cas divergents de la base de données ABB mentionné plus haut, la relation entre les paramètres a été enlevée afin de vérifier si un résultat satisfaisant peut être trouvé. Aucune relation entre les paramètres indique que la valeur de la résistance et de la réactance de la première cage du rotor n'est pas influencée par celle de la deuxième cage et vice-versa. Pour rappel, la résistance de la première cage est théoriquement plus faible que celle de la deuxième cage, alors que la relation est inversée dans le cas des réactances. Étant donné que ces relations sont intrinsèquement reliées à la construction de la machine, que ce soit la forme des encoches ou encore les matériaux utilisés, il faut vérifier que les courbes obtenues pour chacune des spécifications ont du sens, c'est-à-dire qu'elles reproduisent le comportement typique d'un moteur asynchrone. Les paramètres des deux cages du rotor et l'erreur maximale sont présentés pour chacune des machines dans le Tableau 3.10.

Tableau 3.10 : Paramètres des cages du rotor et erreur maximale pour les cas divergents

Paramètres	$R_{r1} [\Omega]$	$R_{r2} [\Omega]$	$X_{1d} [\Omega]$	$X_{2d} [\Omega]$	Erreur maximale (%)
Machine 17	0.22	0.09	0.17	0.23	0.20
Machine 18	0.19	0.14	0.23	0.32	0.20
Machine 49	0.07	0.03	0.15	0.16	1.69
Machine 51	0.07	0.04	$5e^{-6}$	0.08	0.14

À la suite du nouveau calcul des paramètres en ne considérant pas la relation entre les paramètres, on observe que la relation entre les paramètres est effectivement inversée, alors que l'erreur la plus grande se trouve sous la limite de 2% pour toutes les machines. L'analyse des courbes du couple en fonction de la vitesse à la Figure 3.6 et à la Figure 3.7 permet toutefois de réaliser que les spécifications ne sont pas réellement satisfaites pour toutes les machines.

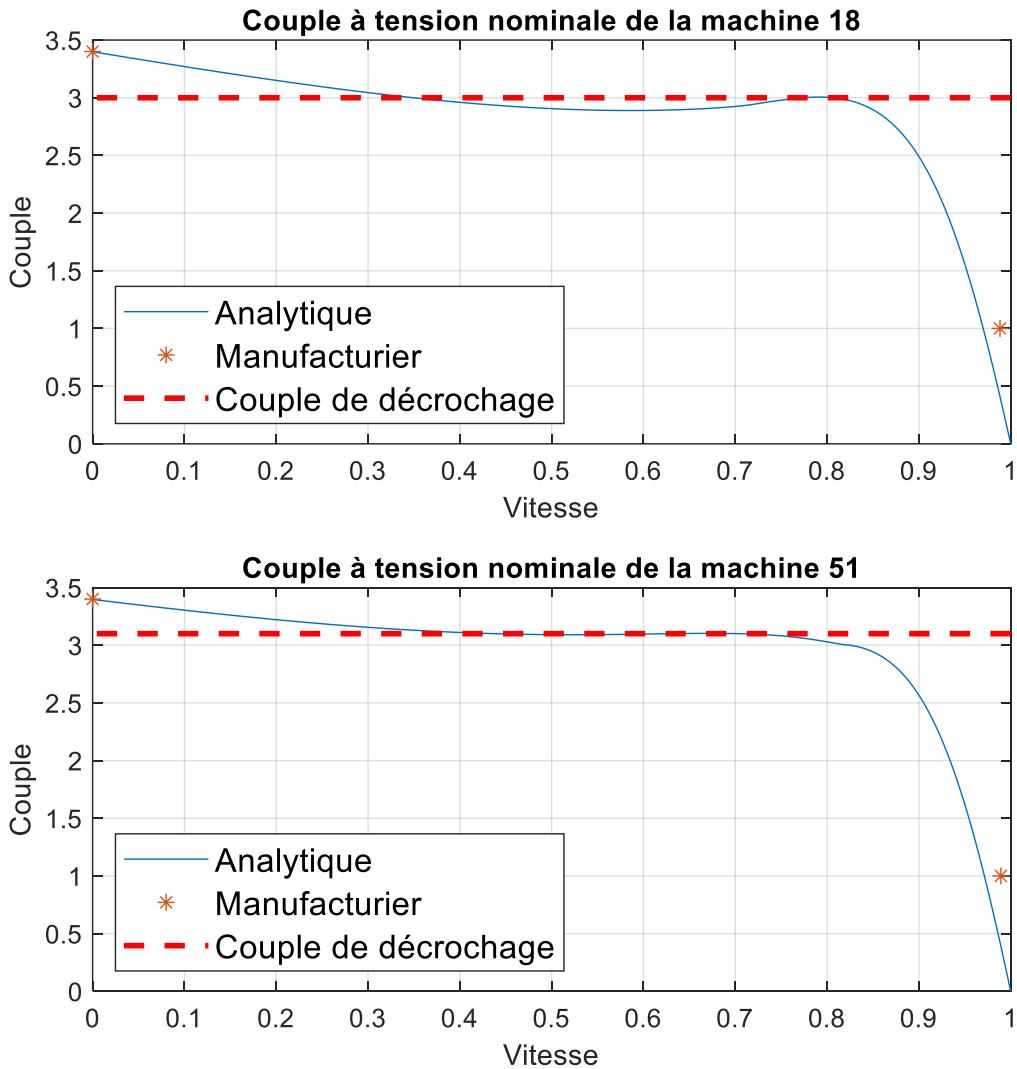


Figure 3.6 : Courbe du couple pour les cas divergents – Modifications acceptables

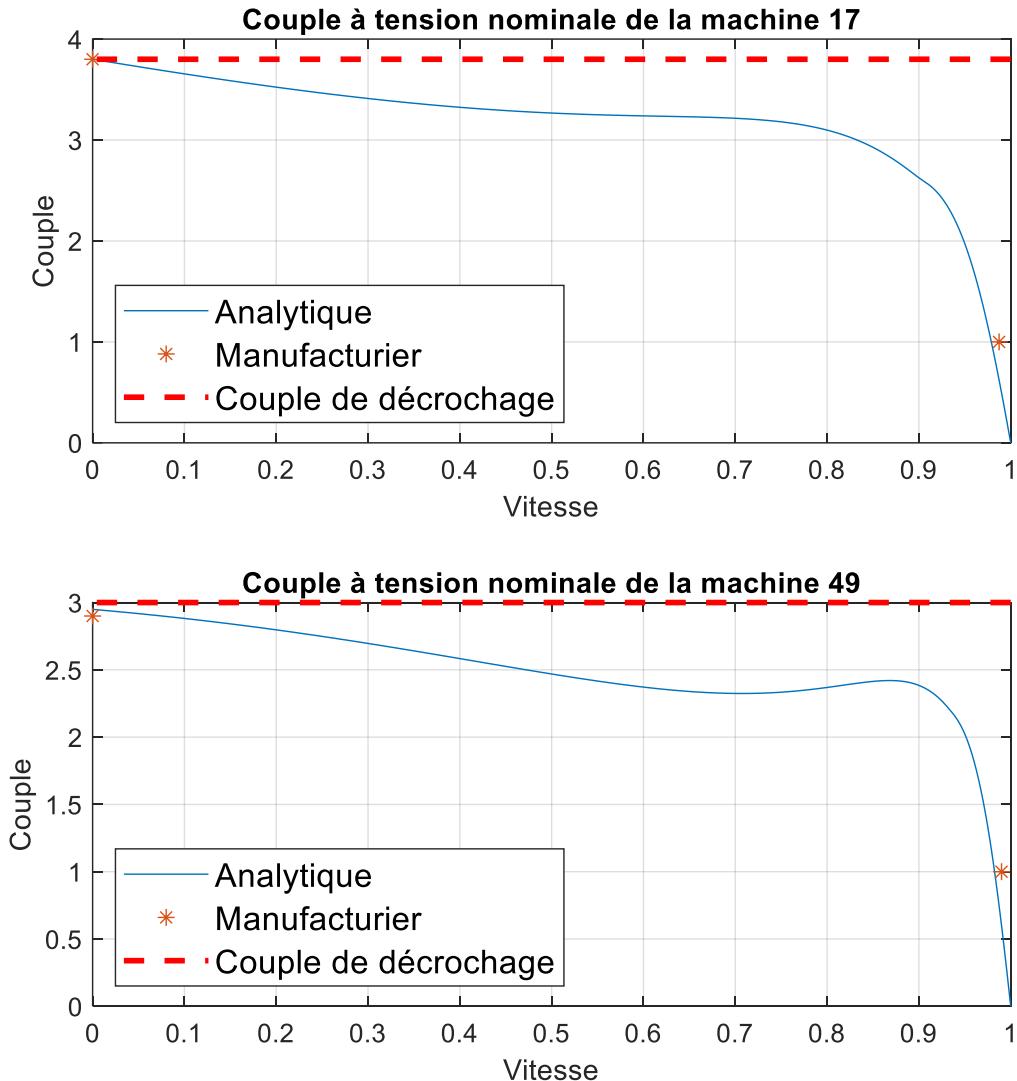


Figure 3.7 : Courbe du couple pour les cas divergents – Modifications non acceptables

On observe que les courbes des machines 18 et 51 sont acceptables. Effectivement, le couple de décrochage se trouve autour de 0.80 pu de vitesse et il s'agit d'un maximum local. Au contraire, les courbes des machines 17 et 49 ne correspondent pas vraiment à une courbe typique du couple. Le maximum local près de 0.80 pu de vitesse est beaucoup plus bas que la valeur du couple de décrochage du manufacturier, ce qui veut dire que le couple de décrochage considéré par le CGV correspond plutôt au couple de démarrage. Encore une fois, le calcul du couple de décrochage est un problème.

Les deux cas où la courbe du couple est valide laissent supposer que la relation entre les paramètres n'est pas réellement nécessaire. Il n'y a toutefois pas d'explications physiques qui justifient cette approche. Il est donc difficile de valider cette méthode même si elle semble prometteuse en théorie. Il faudrait valider à l'aide d'essais sur la machine pour voir si cette relation est effectivement réelle ou encore trouver une explication physique à ce phénomène, notamment pour les machines de faible puissance ou encore pour un rotor bobiné avec résistances externes au démarrage. Il ne serait pas recommandé d'enlever cette relation pour toutes les machines puisqu'elle mène certainement à des faux cas de convergence.

Ainsi, les divergences semblent être causées par le calcul du couple de décrochage. Ce calcul devrait être amélioré et corrélé avec la classe NEMA de la machine lorsque cette information est disponible. Dans tous les cas, le calcul devrait être différent en fonction du rapport entre le couple de démarrage et le couple de décrochage. Le retrait de la relation entre les paramètres permet de faire converger le calcul dans certains cas, mais rien ne justifie l'utilisation de cette approche d'un point de vue physique.

En somme, le CGV développé dans l'environnement MATLAB permet de déterminer des paramètres de la machine asynchrone avec un taux de convergence d'environ 95%. Ces résultats ont été démontrés à l'aide de deux banques de données distinctes, soit une provenant de l'IREQ et une autre de ABB. Certaines techniques ont aussi été présentées afin de trouver une solution aux cas divergents.

CHAPITRE 4 MÉTHODE DYNAMIQUE

Bien que de bons résultats soient obtenus avec la méthode analytique, rien n'assure que des résultats similaires seront obtenus avec un logiciel de simulation comme EMTP, ce qui est ultimement le but. Effectivement, il arrive que le comportement de la machine en simulation ne soit pas le même que celui du modèle analytique. Le schéma EMTP ainsi que les détails de la simulation sont d'abord présentés. L'utilisation d'un API qui permet de faire une simulation dans EMTP à partir de MATLAB sans avoir à modifier manuellement les paramètres de la machine est également proposée. Ensuite, un cas où un écart important est observé entre le modèle simulé et le modèle analytique est mis en évidence. L'influence de la partie saturable variable des inductances de fuite est d'ailleurs démontrée à ce moment. Finalement, deux méthodes dynamiques qui visent à régler ce problème sont introduites, soit une méthode qui se base sur l'influence de chaque paramètre sur les spécifications de la machine et une autre qui utilise l'erreur entre la simulation et les équations analytiques.

4.1 API MATLAB-EMTP

Le logiciel EMTP possède un API afin de communiquer avec MATLAB. Il permet de lancer des simulations directement à partir de MATLAB sans avoir à modifier quoi que ce soit manuellement dans EMTP, ce qui accélère grandement le processus d'optimisation et qui permet d'automatiser les différents essais et comparaisons. De la documentation est disponible sur l'outil et le lecteur est invité à la consulter afin d'en apprendre plus sur l'API en tant que tel.

Afin de s'interfacer avec le CGV, un fichier de simulation pour EMTP a été créé avec une machine asynchrone et une source de tension. La résistance qui représente les pertes fer a été ajoutée à l'extérieur de la machine étant donné que le modèle de base de EMTP n'en contient pas. Le schéma est présenté à la Figure 4.1 alors que le fichier EMTP se trouve dans le même répertoire que le code MATLAB.

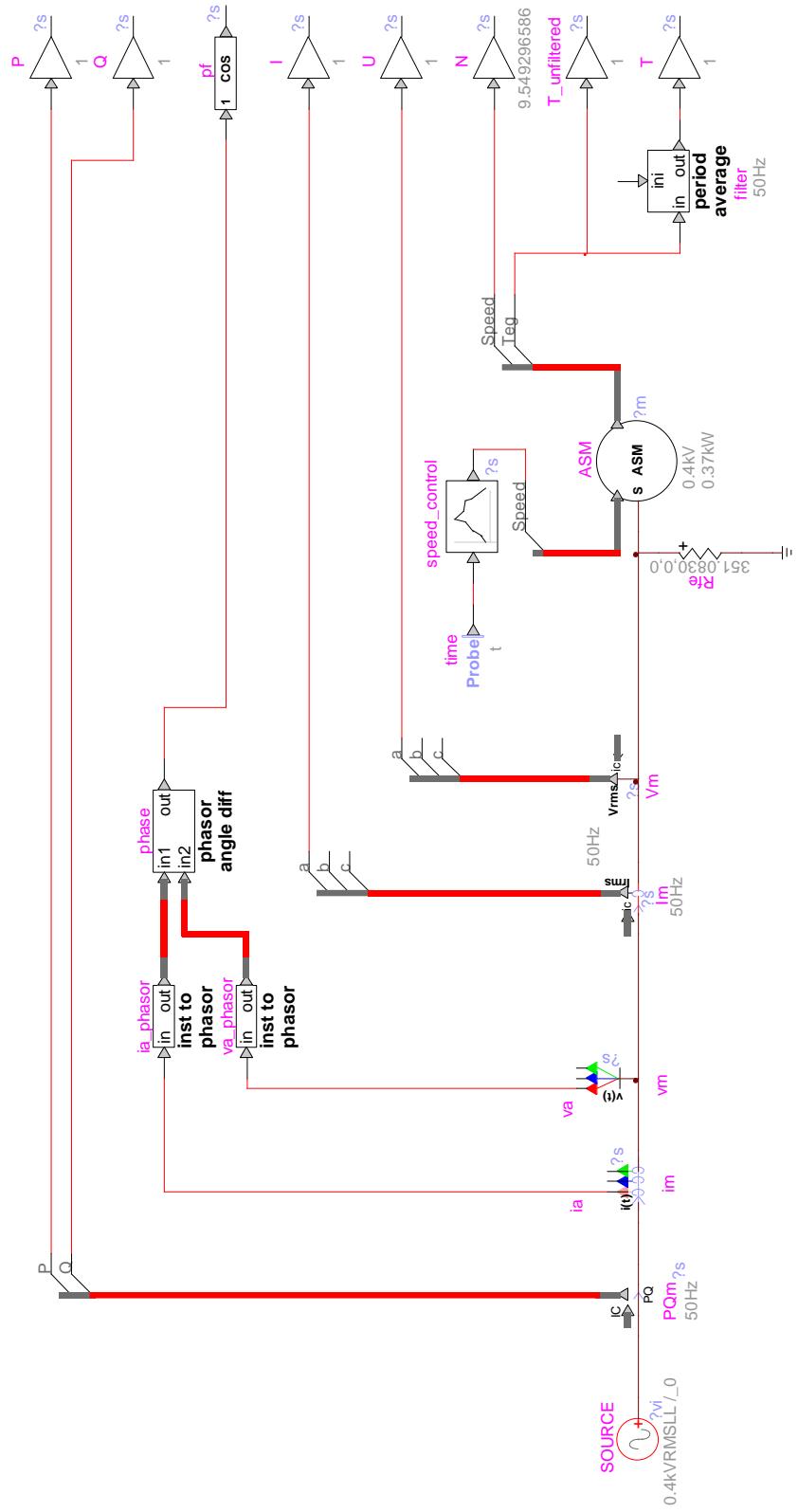


Figure 4.1 : Schéma EMTP

Pour simuler le comportement de la machine, les paramètres de la machine ainsi que la source de tension sont d'abord modifiées, puis le contrôle de la machine est ajusté afin de faire la varier la vitesse entre 0% et 100% de sa vitesse synchrone. Les caractéristiques du couple, du courant ainsi que du facteur de puissance sont finalement enregistrées afin d'être analysées. Le code MATLAB qui permet de communiquer entre les deux applications se trouve dans le fichier *emtp_nominal.m*. Un schéma est présenté à la Figure 4.2 afin de représenter la relation entre MATLAB et EMTP.

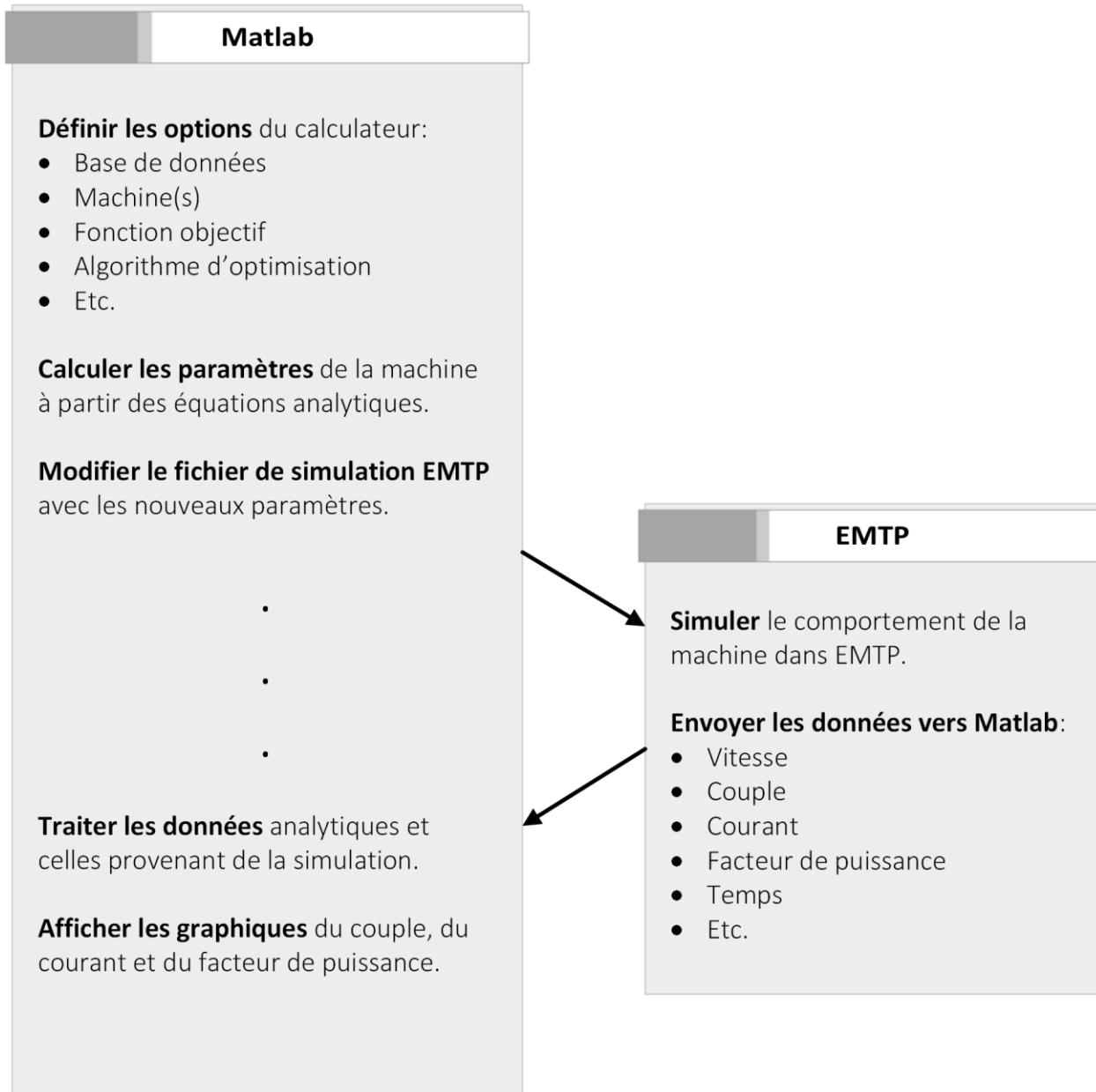


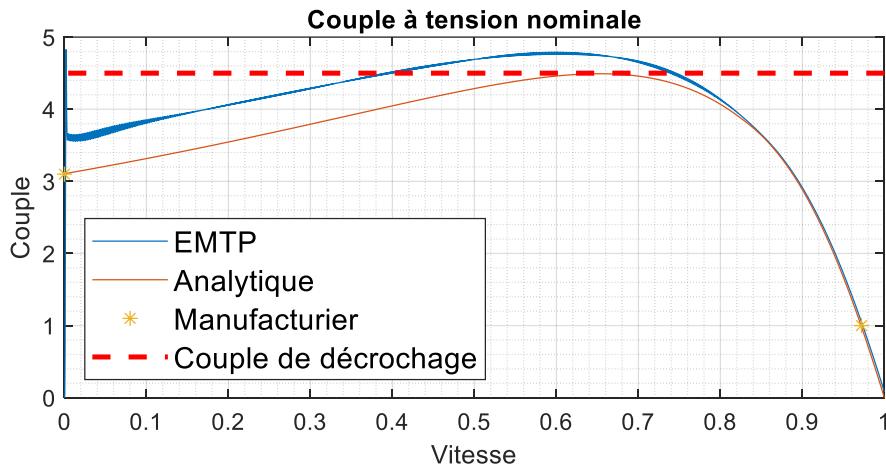
Figure 4.2 : Schéma MATLAB-EMTP

Dans le fichier EMTP, le pas de temps ainsi que la durée de la simulation doivent être soigneusement ajustés afin d'obtenir des résultats acceptables. Si le pas de temps est trop grand, il y aura des non-linéarités entre les points, alors qu'une pas de temps trop court augmente inutilement le temps d'exécution de la simulation. Le pas de temps est fixé à $50\mu\text{s}$, alors que la durée de la simulation est de 10s. Ces paramètres permettent d'obtenir des résultats assez précis en quelques secondes seulement. Une analyse plus approfondie de ces paramètres est présentée au chapitre suivant en lien avec les oscillations engendrées dans le régime transitoire.

Puisque le signal du couple oscille constamment, un filtre à la fréquence de la source peut être appliqué afin d'obtenir une courbe plus lisse. Ces oscillations forcent tout de même à revoir le calcul des spécifications. Ainsi, une moyenne sur quelques centaines de points est fait au démarrage, alors que le couple de décrochage est évalué au glissement de décrochage déterminé par la méthode analytique. Finalement, le même principe est appliqué aux signaux du courant et du facteur de puissance, même si l'oscillation est moins présente pour ceux-ci. Les courbes du couple, du courant et du facteur de puissance peuvent alors être comparées entre le modèle simulé et le modèle analytique.

4.2 Écart entre le modèle simulé et le modèle analytique

La machine 10 de la banque de données ABB est analysée afin de valider le modèle de la machine en simulation. Les courbes provenant de la simulation dans EMTP ainsi que les courbes analytiques sont juxtaposées dans la Figure 4.3.



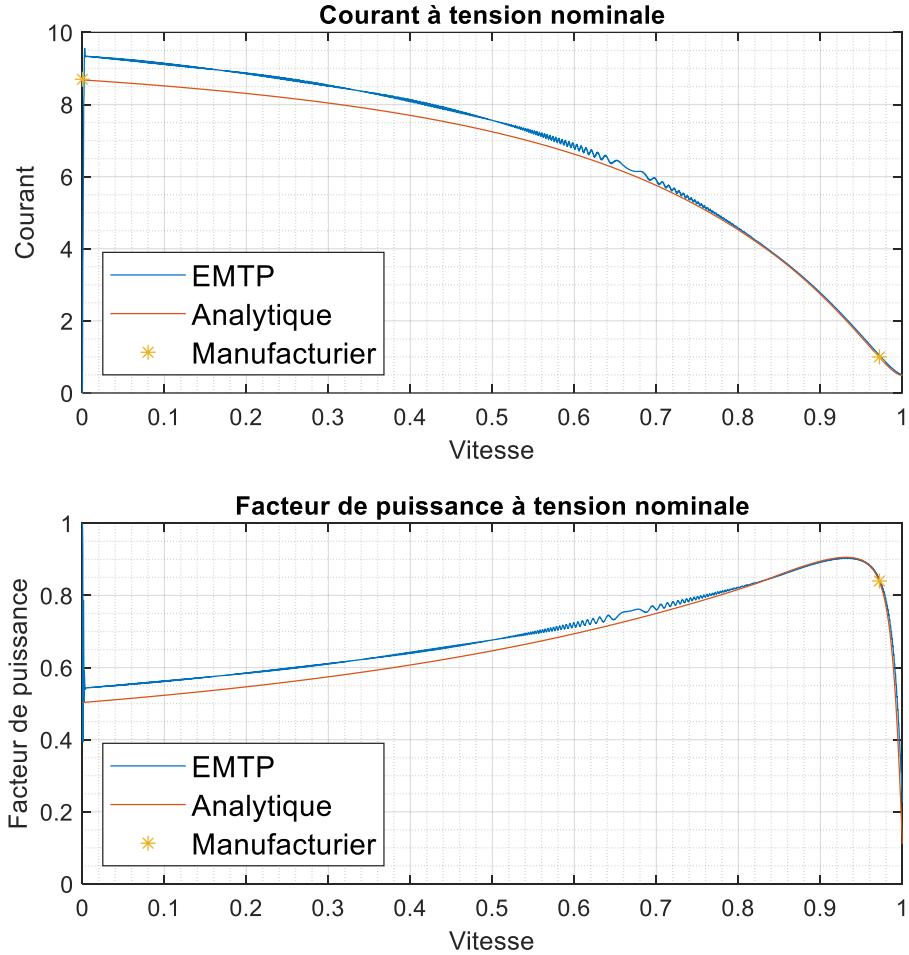


Figure 4.3 : Courbes simulées et analytiques de la machine 10

On observe que les courbes simulées ne correspondent pas aux spécifications de démarrage autant pour le couple que le courant, alors que le couple de décrochage n'est pas respecté. La simulation ne permet donc pas de représenter adéquatement le comportement de la machine même si le modèle analytique donne des résultats satisfaisants. L'erreur maximale est d'ailleurs de 16.25%, ce qui est clairement inacceptable.

Cette différence est due au fait que la partie saturable des inductances de fuite est fixée à 0.5 dans la simulation alors qu'elle est variable dans le CGV. La simulation ne retournera donc pas les mêmes résultats que le modèle analytique. Le fait que les courbes soient plus éloignées pour les vitesses faibles que pour les vitesses élevées donne du poids à cette explication, étant donné que la partie saturable possède une influence plus grande au démarrage lorsque le courant est élevé.

Pour vérifier l'affirmation précédente, il est possible de recalculer un nouvel ensemble de paramètres en fixant le paramètre PS à 0.5. Les nouveaux paramètres calculés ainsi que la courbe du couple pour la même machine 10 de la banque de données ABB sont présentés au Tableau 4.1 et à la Figure 4.4.

Tableau 4.1 : Paramètres de la machine 10 pour $PS = 0.5$

R_{fe} [Ω]	R_s [Ω]	R_{r1} [Ω]	R_{r2} [Ω]	X_{sd} [Ω]	X_m [Ω]	X_{1d} [Ω]	X_{2d} [Ω]	I_{sat} [pu]	PS
448.5	0.68	0.66	2.87	0.38	30.43	1.02	1.02	8.6987	0.5

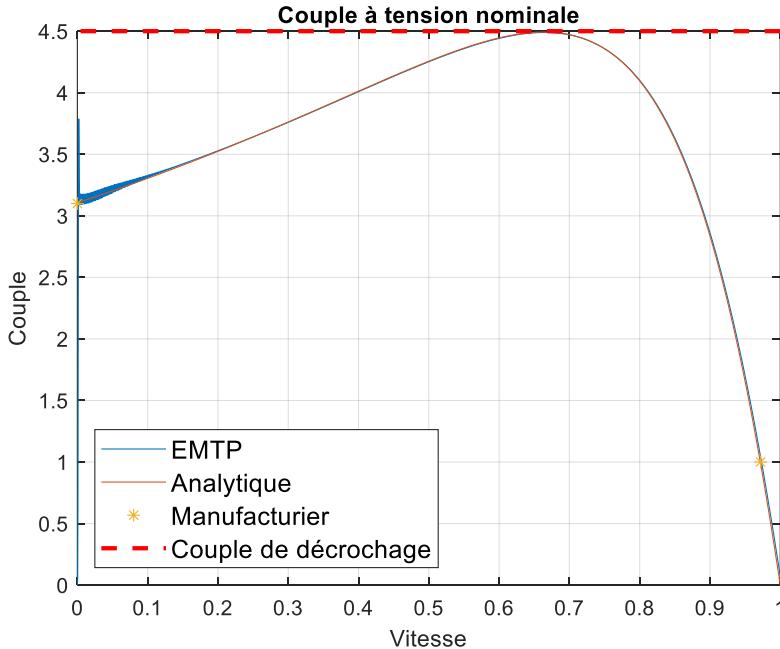


Figure 4.4 : Courbes simulées et analytiques du couple de la machine 10 pour $PS = 0.5$

On observe maintenant que les résultats sont identiques dans les deux cas, ce qui laisse croire que l'erreur est effectivement reliée à ce paramètre.

Afin de contourner ce problème, il faudrait calculer les paramètres en conservant PS fixe à 0.5, ce qui aurait comme désavantage de rendre le CGV moins performant. Une autre option serait de

modifier le modèle interne dans EMTP afin de rendre ce paramètre variable. Cette solution est présentement en attente d'être exécutée et ne peut donc pas être utilisée immédiatement. En considérant le modèle actuel dans EMTP, deux méthodes sont envisagées afin d'obtenir des courbes qui respectent les spécifications du manufacturier en simulation malgré un paramètre *PS* variable. Ces deux méthodes sont présentées à la section suivante.

4.3 Calcul dynamique des paramètres

Afin de combler l'écart entre les résultats obtenus dans la simulation et ceux obtenus à partir des équations analytiques, deux solutions distinctes sont envisagées. Ces méthodes sont dites dynamiques étant donné qu'elles se basent sur le comportement de la machine en simulation.

La première méthode évalue l'influence de chacun des paramètres sur les spécifications afin d'identifier les paramètres à modifier.

La deuxième méthode effectue plutôt un nouveau calcul des paramètres à l'aide de l'algorithme d'optimisation, à la différence que l'erreur considérée est celle par rapport aux résultats de la simulation.

Les deux méthodes ainsi que les résultats pour la machine 10 de la base de données ABB sont présentés dans les prochaines sections. Dans les deux cas, il s'agit plus d'une preuve de concept que d'une démonstration complète.

4.3.1 Méthode 1 – Calcul basé sur l'influence des paramètres

La première méthode se base sur l'influence des paramètres sur les spécifications de la machine. En faisant varier chaque paramètre de manière individuelle, il est possible de mesurer l'effet sur les spécifications et d'isoler les paramètres qui possèdent une influence significative. Ainsi, en modifiant seulement un ou quelques paramètres autour de leur valeur respective, on espère obtenir des résultats satisfaisants en simulation.

Pour connaître l'influence d'un paramètre sur les spécifications, on modifie sa valeur par un facteur de correction de 0.8 et de 1.2. Par exemple, on multiplie le paramètre R_{fe} par 0.8 sans modifier les autres paramètres et on observe l'effet sur les spécifications. On répète le même exercice pour tous les paramètres afin d'obtenir une vision d'ensemble. On peut alors diminuer l'erreur observée en simulation en modifiant le paramètre qui possède la plus grande influence sur cette spécification,

en s'assurant toutefois de ne pas s'éloigner d'une autre spécification au passage. Afin de traiter toute cette information, on peut s'aider en construisant une table de valeurs à l'aide du fichier *compare_parameters_variation.m*. Cette table est ensuite analysée et comparée avec les erreurs sur les spécifications calculées avec les résultats de simulation. Les données sont disponibles dans un fichier excel qui porte le même nom et qui se trouve dans le même répertoire que le code MATLAB.

Un exemple avec la machine 10 de la base de données ABB est illustré à l'aide du Tableau 4.2 et du Tableau 4.3. Les erreurs relatives par rapport à un facteur de correction unitaire sont présentées pour chacune des spécifications. Un code de couleur a été ajouté afin d'identifier plus facilement l'influence de chaque paramètre. Les cases en rouge indiquent une augmentation de l'erreur alors que les cases en bleu indiquent une diminution. Plus la couleur est foncée et plus l'impact est grand. Afin de mieux comparer avec les paramètres analytiques, les erreurs obtenues en simulation sont présentées avec le même code de couleur au Tableau 4.4. Pour rappel, les équations analytiques retournent une erreur maximale de seulement 0.20% alors que l'erreur en simulation est de 16.25% pour les mêmes paramètres.

Tableau 4.2 : Influence sur les spécifications pour un facteur de correction de 0.8

	R_{fe}	R_s	R_{r1}	R_{r2}	X_{sd}	X_m	X_{1d}	X_{2d}	i_{sat}
T_n	-0.33	0.15	0.16	-0.02	0.01	-0.04	0.18	0.16	0.00
T_{st}	0.00	5.53	-14.07	-0.23	4.83	-0.35	18.40	16.55	9.44
T_{br}	0.00	5.39	0.00	0.00	2.78	-0.31	10.09	9.08	6.12
I_n	0.75	-0.12	0.12	-0.01	-0.08	6.92	-0.28	-0.26	0.00
I_{st}	0.07	2.66	3.43	0.05	2.35	0.88	8.22	7.42	4.34
pf_n	0.34	-0.32	0.03	0.00	0.08	-6.10	0.43	0.39	0.00

Tableau 4.3 : Influence sur les spécifications pour un facteur de correction de 1.2

	R_{fe}	R_s	R_{r1}	R_{r2}	X_{sd}	X_m	X_{1d}	X_{2d}	i_{sat}
T_n	0.00	-0.17	-0.02	0.12	-0.01	-0.09	-0.19	-0.17	0.00
T_{st}	0.00	-5.39	11.47	0.15	-4.59	0.23	-15.36	-14.05	-8.48
T_{br}	0.00	-5.02	-0.01	0.00	-2.67	0.21	-8.71	-7.95	-4.81
I_n	-0.66	0.11	-0.02	0.09	0.08	-4.19	0.28	0.25	0.00
I_{st}	-0.05	-2.67	-3.42	-0.04	-2.29	-0.59	-7.48	-6.81	-4.08
pf_n	-0.28	0.31	0.00	0.02	-0.08	4.04	-0.43	-0.39	0.00

Tableau 4.4 : Erreur sur les spécifications en simulation

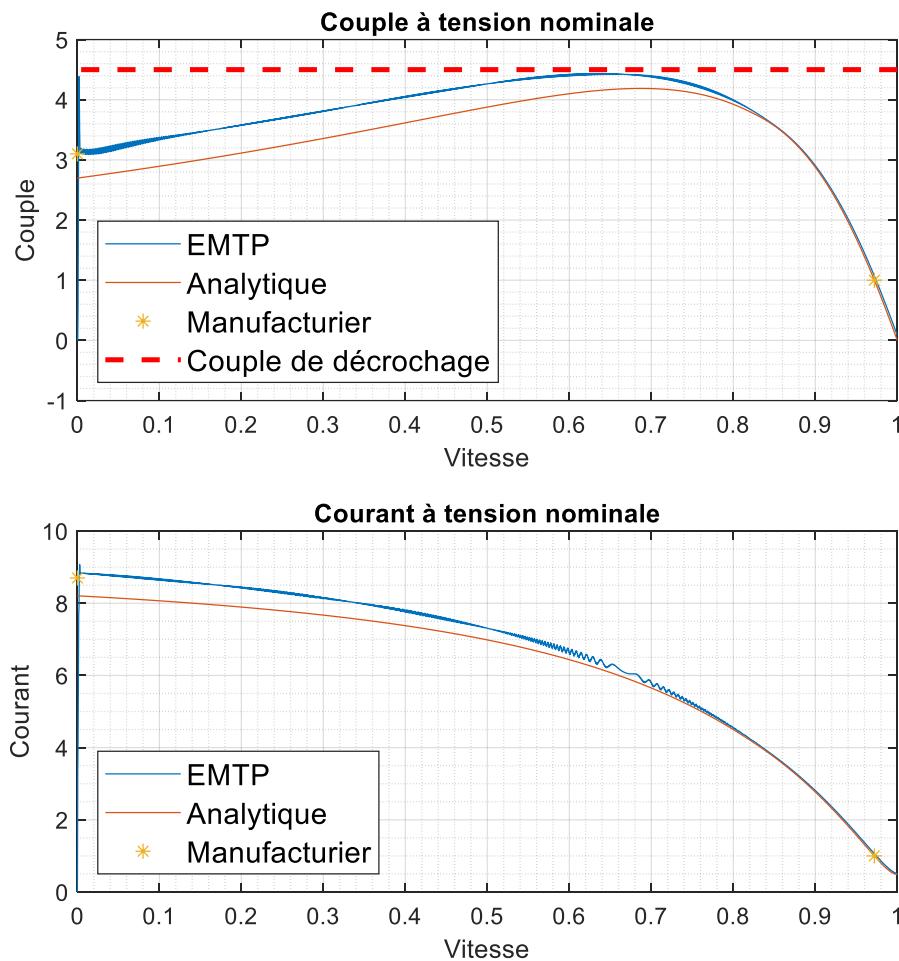
	Erreur (%)
T_n	0.00
T_{st}	16.25
T_{br}	6.08
I_n	0.30
I_{st}	7.08
pf_n	-0.27

Le but est donc d'identifier le paramètre qui permet de diminuer le plus les erreurs sur les spécifications. Dans ce cas précis, il faut diminuer le couple de démarrage, le couple de décrochage ainsi que le courant de démarrage sans toutefois trop modifier les conditions nominales qui sont déjà acceptables. Afin de ramener l'erreur plus près de zéro, on observe que le paramètre qui possède l'influence la plus grande est X_{1d} . Effectivement, un facteur de correction de 1.2 diminue de manière significative l'erreur sur le couple de démarrage, sur le couple de décrochage et sur le courant de démarrage de la machine. En appliquant un facteur de correction de 1.2 à ce paramètre, on obtient des résultats qui sont cette fois-ci inférieur aux spécifications du manufacturier. Après quelques essais, on trouve qu'un facteur de 1.08 permet d'obtenir des résultats plus satisfaisants, soit une erreur maximale de 2.32% sur le couple de démarrage.

Il est alors possible de réduire encore plus cette erreur en modifiant le paramètre R_{r1} . Un facteur de correction inférieur à 1 sur ce paramètre permet de diminuer le couple de démarrage et d'augmenter légèrement le courant de démarrage. En conservant le facteur de 1.08 sur X_{1d} et en appliquant un facteur de 0.98 sur R_{r1} , on obtient finalement des résultats acceptables. L'erreur sur les spécifications ainsi que les courbes simulées et analytiques sont présentées au Tableau 4.5 et à la Figure 4.5. Pour rappel, la courbe EMTP correspond aux résultats de simulation, alors que la courbe analytique provient des équations définies au Chapitre 2.

Tableau 4.5 : Erreur sur les spécifications en simulation après correction – Méthode 1

	Erreur (%)
T_n	-0.62
T_{st}	1.07
T_{br}	-1.75
I_n	0.01
I_{st}	1.33
pf_n	-1.07



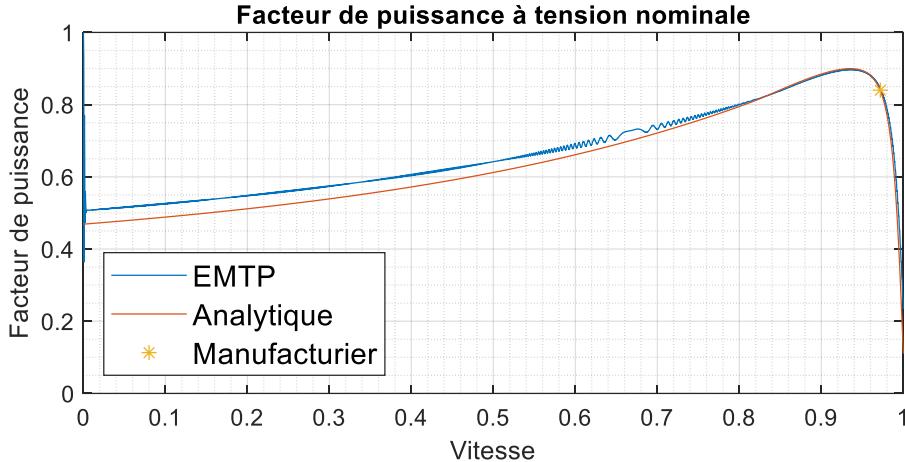


Figure 4.5 : Courbes simulées et analytiques après correction – Méthode 1

L’erreur maximale est inférieure à la limite de 2%, alors que les courbes simulées correspondent effectivement aux spécifications du manufacturier. Cette méthode est donc valide et permet de trouver une solution en se basant sur l’influence des paramètres sur les spécifications. Il est également possible d’observer que la courbe analytique ne respecte plus les spécifications avec ces nouveaux paramètres, ce qui confirme encore une fois qu’il y a réellement une différence entre le modèle analytique et la simulation lorsque PS est différent de 0.5.

Il est à noter que les tables de valeurs ont été obtenues à partir du modèle analytique étant donné que le calcul avec EMTP requiert un temps de calcul beaucoup plus élevé. Puisque seul l’ordre de grandeur de la variation et non la valeur exacte est désirée, cette approche permet d’obtenir des résultats rapidement sans avoir à faire de simulation. Il a aussi été validé que le comportement de la machine en simulation est le même que celui du modèle analytique en ce qui concerne la variation des paramètres.

À terme, cette technique évaluerait un plus grand nombre de cas et calculerait même les différentes combinaisons entre les paramètres afin de trouver une solution optimale. La technique de Monte-Carlo pourrait être utilisée afin de calculer les différents cas possibles. Sinon, il serait possible de calculer différents facteurs de correction et d’interpoler entre les points de données plutôt que d’utiliser une méthode par essais et erreurs.

4.3.2 Méthode 2 – Calcul basé sur l'erreur

La deuxième méthode se base directement sur l'erreur obtenue avec le modèle simulé. En ajoutant cette erreur à la fonction objectif, un nouvel ensemble de paramètres peut être calculé à l'aide de l'algorithme d'optimisation. Ainsi, en quelques itérations seulement, il est possible de trouver des paramètres qui respectent les spécifications en simulation.

Une première simulation avec les paramètres analytiques retourne l'erreur simulée sur chacune des spécifications. Cette erreur est comparée avec l'erreur analytique afin de trouver la différence entre la simulation et le modèle analytique. Le prochain calcul des paramètres tente alors de minimiser cette différence, ou en d'autres mots de trouver un ensemble de paramètres qui comble la différence entre les spécifications du modèle et celles de la simulation.

Dans le code MATLAB, les fonctions *dynamic_calculateParam.m*, *dynamic_objective.m*, *dynamic_getModelData.m* et *dynamic_graphics_calculateParam.m* sont modifiées par rapport aux fonctions correspondantes sans le préfixe *dynamic* afin de prendre en compte les changements énoncés plus haut.

La fonction objectif dynamique est présentée à l'équation suivante :

$$f_{dynamic}(x) = f(x) - \Delta e \quad (4.1)$$

$$\Delta e = e_{analytique} - e_{simulation} \quad (4.2)$$

où

$f_{dynamic}(x)$: fonction objectif dynamique

$f(x)$: fonction objectif

$e_{analytique}$: erreurs sur les spécifications obtenues à partir des équations analytiques

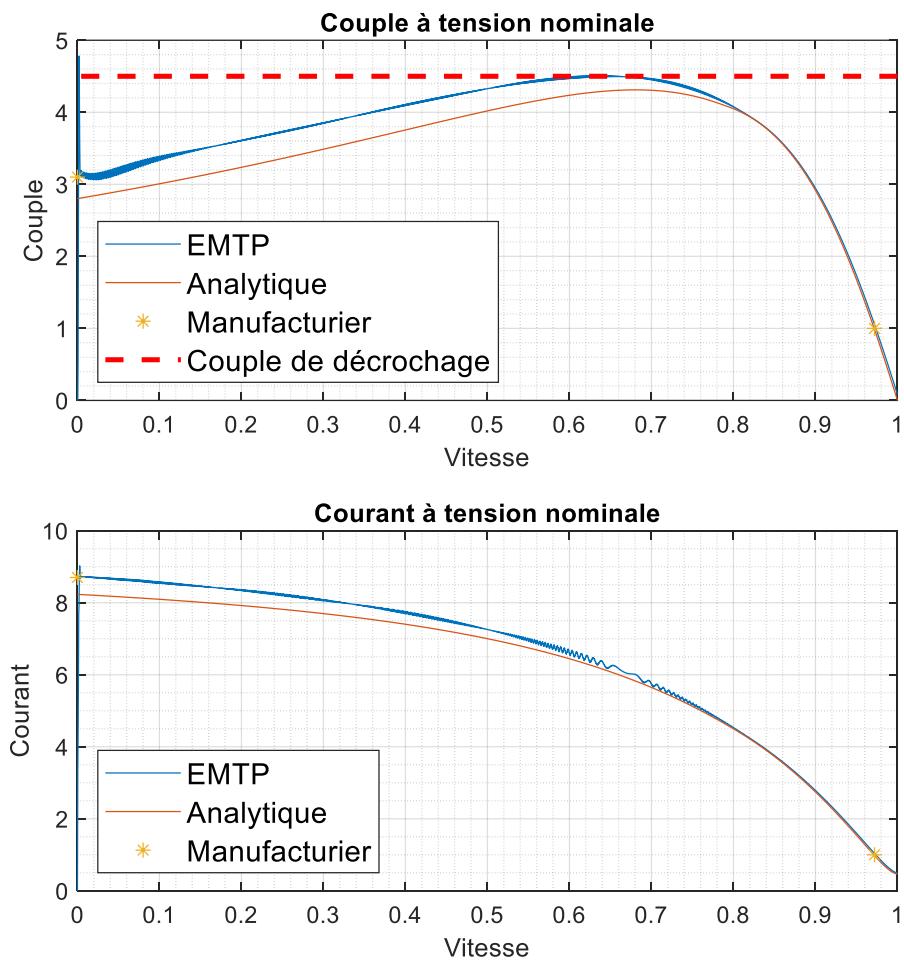
$e_{simulation}$: erreurs sur les spécifications obtenues à partir de la simulation

Le calcul des paramètres prend encore en compte la fonction objectif définie précédemment, mais la différence entre les résultats analytiques et simulés est aussi considérée. Puisque les paramètres analytiques sont bons en théorie, les bornes sont réduites à 0.9 et 1.1 fois la valeur de chaque paramètre, alors que les paramètres de l'optimisation sont laissés relativement faibles, soit 50 et 10 pour le nombre d'itérations et le nombre de particules respectivement. Quelques itérations sont

suffisantes pour trouver une solution valide. Le même exemple de la machine ABB 10 est utilisé afin de démontrer la validité de cette méthode. Les erreurs calculées sont regroupées au Tableau 4.6 alors que les courbes sont présentées à la Figure 4.6.

Tableau 4.6 : Erreur sur les spécifications en simulation après correction – Méthode 2

	Erreur (%)
T_n	0.02
T_{st}	0.38
T_{br}	-0.22
I_n	-0.35
I_{st}	0.07
pf_n	-0.06



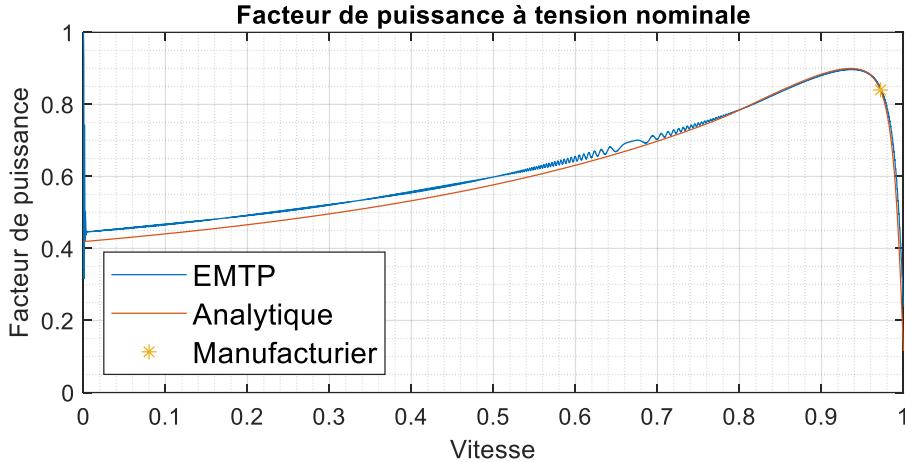


Figure 4.6 : Courbes simulées et analytiques après correction – Méthode 2

L’erreur maximale est de seulement 0.38%, alors que les courbes simulées respectent maintenant chacune des spécifications, c’est-à-dire que les courbes en bleu passent sur chacun des points de données provenant du manufacturier. Le calcul a nécessité 2 itérations avant de converger, ce qui a permis de trouver une solution en quelques dizaines de secondes seulement, y compris le temps des simulations dans EMTP.

4.3.3 Comparaison entre les deux méthodes

Les deux méthodes présentées plus haut permettent toutes les deux d’obtenir des courbes satisfaisantes en simulation en se basant sur un premier calcul analytique. Elles présentent toutefois certains avantages et désavantages qu’il convient de séparer.

La première méthode, soit celle qui évalue l’impact de chaque paramètre sur le respect des spécifications, est légèrement plus compliquée à implémenter. Elle permet toutefois de mieux cibler les différences entre le modèle analytique et la simulation. L’automatisation de cette méthode, que ce soit à l’aide d’une étude statistique comme Monte-Carlo ou encore à l’aide d’une interpolation, permettrait d’accélérer le calcul et de faciliter la tâche de l’utilisateur. Il est évident qu’une méthode par essais et erreurs n’est pas optimale.

La deuxième méthode qui incorpore l’erreur obtenue en simulation dans le calcul des paramètres est plus simple à comprendre puisqu’elle utilise essentiellement le même processus que le calcul analytique. Cette méthode considère toutefois que l’erreur est constante, ce qui n’est assurément

pas le cas si on se fie à l'allure des courbes. Le nouvel ensemble de paramètres n'engendrera pas forcément la même différence que l'ensemble de paramètres précédent.

Ainsi, la première méthode est plus prometteuse que la seconde méthode puisqu'elle considère réellement la différence entre les résultats du modèle analytique et du modèle simulé. L'automatisation de cette méthode devrait faire l'objet de travaux futurs.

Il est important de se rappeler que l'ajout du paramètre PS variable directement dans EMTP ferait en sorte qu'il ne serait pas nécessaire d'avoir recours à ces méthodes afin d'obtenir des résultats de simulation satisfaisants par rapport au calcul analytique.

CHAPITRE 5 RÉGIME TRANSITOIRE

Les résultats présentés au chapitre précédent ne tiennent compte que du régime statique d'opération de la machine sans considérer le régime transitoire. Une bonne représentation du régime transitoire, que ce soit lors du démarrage ou lors d'une variation soudaine des conditions d'opération à partir du régime permanent nécessite de bien définir les paramètres de la simulation et de connaître les paramètres électromécaniques de la machine. D'abord, les paramètres de simulation sont analysés en lien avec les oscillations créées dans le régime transitoire. Ensuite, une description des équations électromagnétiques qui sont utilisées dans EMTP et leur relation par rapport aux données du manufacturier est introduite, notamment à travers le calcul du moment d'inertie. Finalement, une comparaison avec de véritables résultats d'essais est présentée afin d'évaluer la réponse transitoire.

5.1 Paramètres de simulation

Les paramètres de simulation incluent la méthode de contrôle de la machine, mais aussi la durée de la simulation et le pas de temps. Ces paramètres possèdent une influence non négligeable sur le régime transitoire de la machine et sur le temps nécessaire afin d'exécuter la simulation.

Tout d'abord, il faut se pencher sur le contrôle en vitesse de la machine qui est nécessaire afin de recréer les courbes des différentes spécifications en fonction de la vitesse. La méthode la plus simple est d'appliquer une rampe de vitesse qui varie de manière linéaire entre 0% et 100% de la vitesse synchrone pour toute la durée de la simulation. Cette technique risque toutefois d'engendrer des oscillations si la variation est trop rapide étant donné que la vitesse de la machine n'a pas nécessairement le temps de se stabiliser.

Un exemple est présenté à l'aide de la Figure 5.1, de la Figure 5.2 et de la Figure 5.3 pour une durée de simulation correspondant à une rampe de vitesse de 10s, 100s et 1000s et un pas de temps de 50 μ s. La caractéristique du couple simulée provient de la machine 10 de la base de données ABB avec le paramètre PS fixe à 0.5.

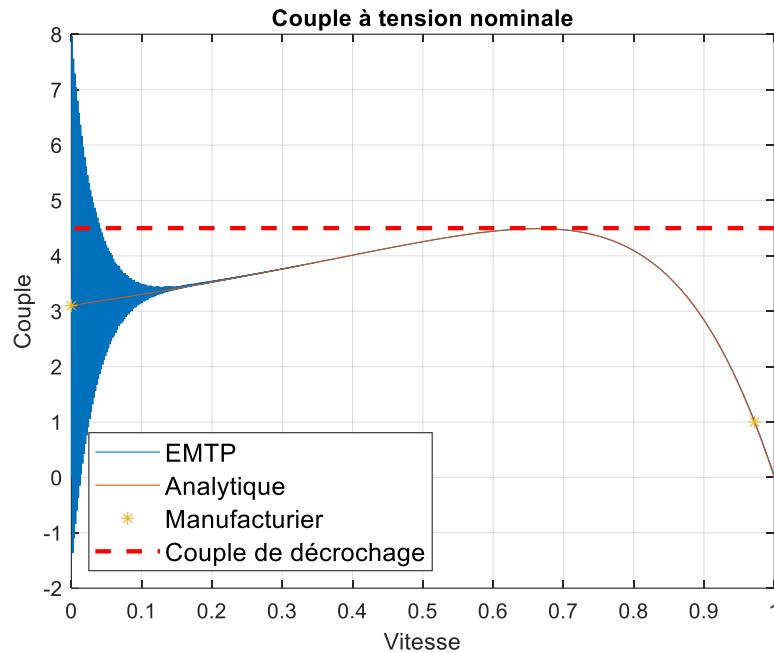


Figure 5.1 : Oscillations pour une simulation de 10s

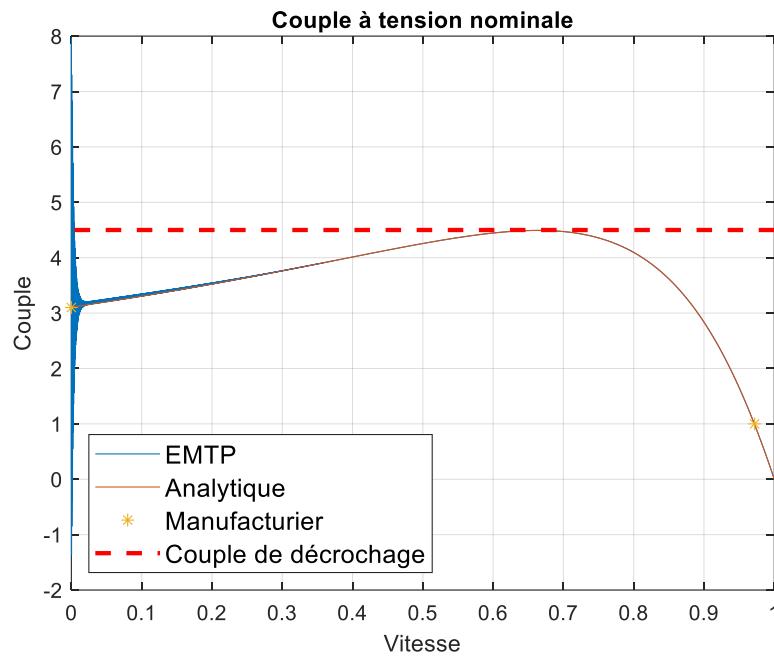


Figure 5.2 : Oscillations pour une simulation de 100s

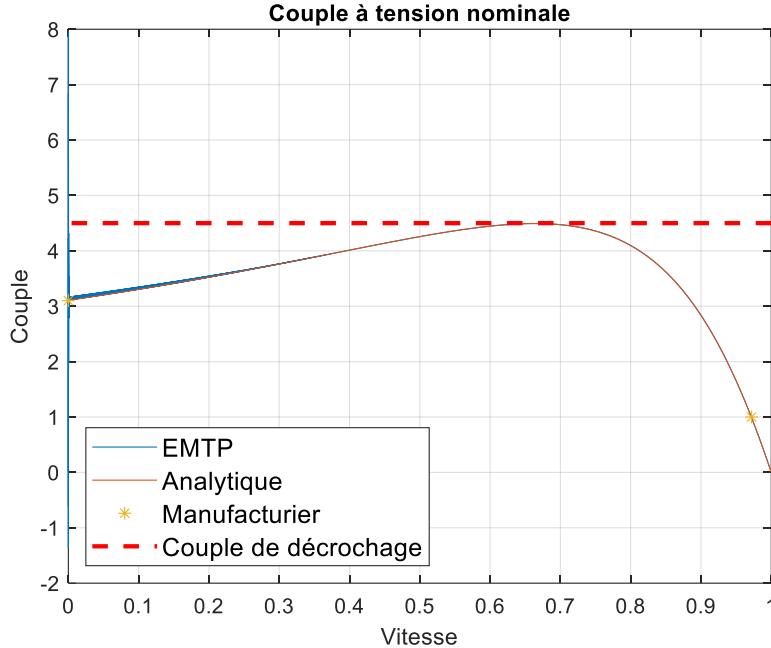


Figure 5.3 : Oscillations pour une simulation de 1000s

Il est possible d’observer que les oscillations sont bien présentes pour les vitesses faibles pour une simulation de seulement 10s, ce qui suggère effectivement que la variation de vitesse est trop rapide pour que la machine se stabilise avant la prochaine commande de vitesse. Au contraire, les oscillations sont grandement diminuées pour une simulation de 100s alors qu’elles ont presque disparues pour une simulation de 1000s. Le temps de simulation est toutefois très long, ce qui rend cette solution plus ou moins adéquate en pratique.

Une solution plus facile à intégrer est d’appliquer un filtre sur le couple afin de réduire les oscillations. Concrètement, le filtre est une simple moyenne à la fréquence d’opération de la source de tension, que ce soit 50 Hz ou 60 Hz. C’est d’ailleurs cette technique qui est utilisée dans les autres chapitres puisqu’elle permet de réduire le temps de simulation de manière significative en plus de fournir des résultats relativement satisfaisants, tel que présentés à la Figure 5.4.

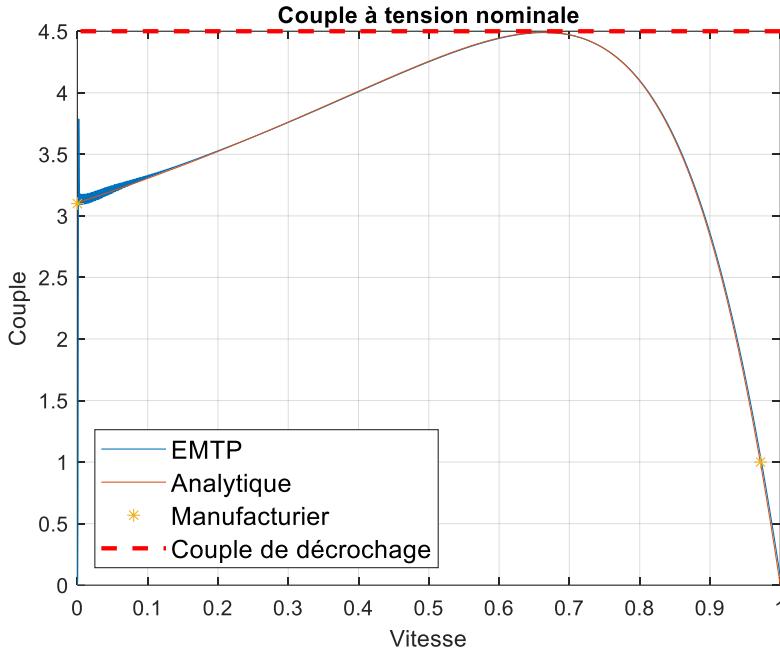


Figure 5.4 : Oscillations pour une simulation de 10s et un filtre à 50Hz

Les oscillations sont encore présentes, mais elles sont beaucoup moins importantes. Il s'agit donc d'un bon compromis entre la précision et le temps nécessaire afin d'exécuter la simulation.

Finalement, une méthode plus sophistiquée pour contrôler la machine est d'attendre que la machine se stabilise avant de faire varier la vitesse. En prenant la valeur en régime permanent pour différentes vitesses, il est possible de recréer les courbes en fonction de la vitesse de manière discrète. Ainsi, la machine est amenée à une vitesse fixe et les spécifications du couple, du courant et du facteur de puissance sont observées seulement lorsque le régime permanent est atteint.

Un exemple de cette méthode est présenté pour la machine 10. Les courbes du régime transitoire pour une vitesse nulle de même que pour la vitesse nominale sont d'abord présentées à la Figure 5.5 et à la Figure 5.6 respectivement, puis la courbe discrète obtenue à partir des résultats de simulation est affichée à la Figure 5.7.

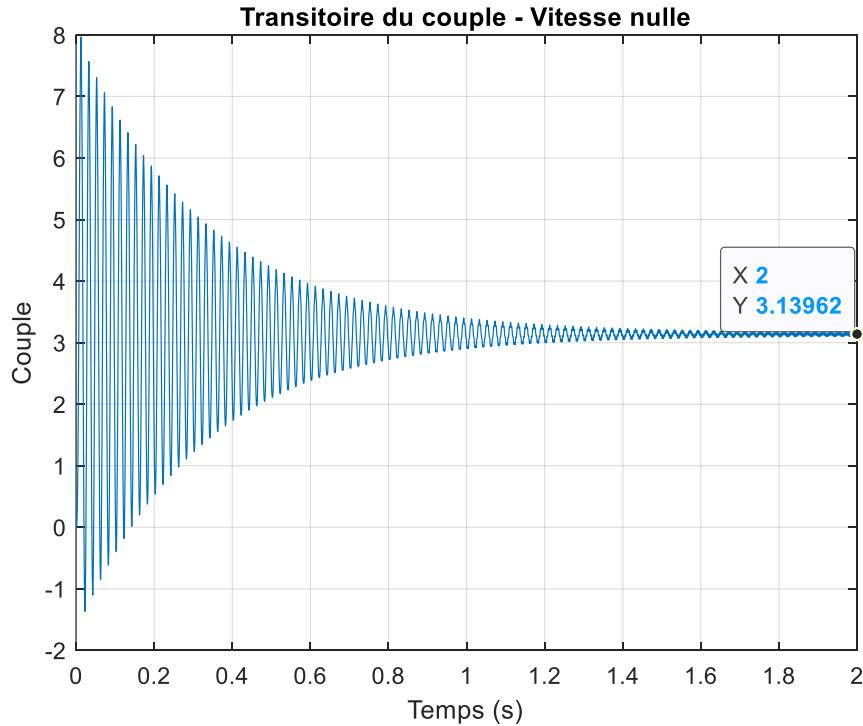


Figure 5.5 : Transitoire du couple à vitesse nulle

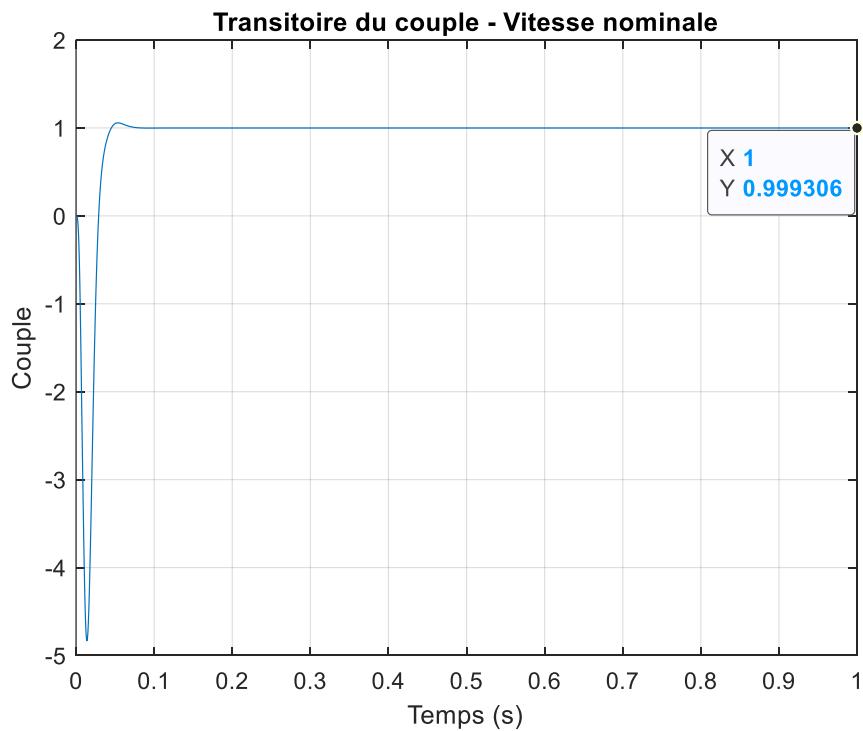


Figure 5.6 : Transitoire du couple à vitesse nominale

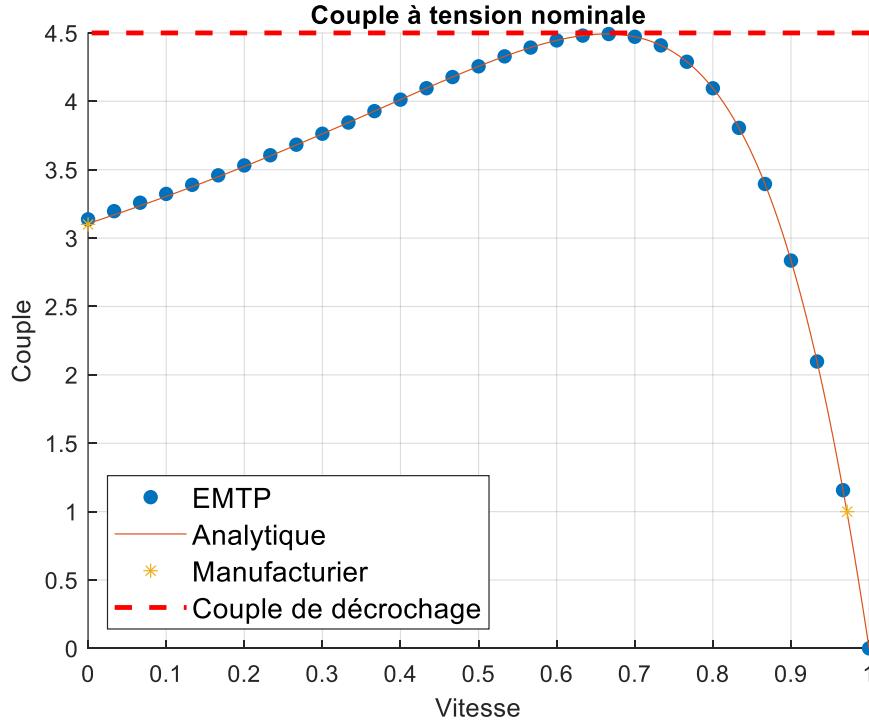


Figure 5.7 : Courbe discrète du couple

L'étude du régime transitoire permet de voir que les oscillations sont beaucoup plus importantes pour une vitesse faible que pour une vitesse élevée, soit plus de 2s pour se stabiliser dans le premier cas alors que le régime permanent est atteint en moins de 0.1s dans le deuxième cas. Au niveau de la courbe discrète, on observe que les points de la simulation correspondent exactement à la courbe analytique. En simulant plus de points, il serait possible de récréer de manière plus fidèle la courbe analytique. Dans le cas où un plus petit nombre de point est suffisant, le temps de calcul est grandement diminué étant donné que chaque point correspond à une simulation dans EMTP.

Ainsi, le contrôle en vitesse de la machine ajoute des oscillations non négligeables dans le régime transitoire, particulièrement à basse vitesse. Une explication n'a pas été cherchée, mais quelques pistes de solution ont été présentées afin de régler ce problème. La méthode utilisée dans ce mémoire est de filtrer le signal du couple, ce qui donne des résultats satisfaisants autant au niveau de la précision que du temps de calcul. Dans tous les cas, il faut s'assurer d'atteindre le régime permanent afin d'éviter des oscillations dans le signal.

5.2 Modèle électromécanique

Jusqu'à maintenant, les résultats de simulation proviennent d'un fichier avec un contrôle en vitesse de la machine. La vitesse est imposée de manière à recréer les courbes du couple, du courant et du facteur de puissance pour toute la plage de vitesse, ce qui permet de faire la comparaison avec les courbes analytiques. Le régime transitoire n'est pas réellement considéré.

Au lieu de contrôler la machine en vitesse, il est possible de la contrôler en lui imposant un couple. Le comportement de la machine dépend alors des variations de la tension et de la fréquence du réseau ou encore de la charge alimentée. Ces changements au niveau des conditions d'opération engendrent des oscillations dans le régime transitoire qu'il est possible d'observer dans EMTP. Il faut alors s'assurer que les conditions restent acceptables et que les autres équipements sont en mesure de fonctionner normalement.

Le modèle électromécanique de la machine asynchrone permet de relier la vitesse de la machine au couple électromagnétique. Le modèle avec masse unique est utilisé afin de simplifier l'explication. Il serait possible d'utiliser un modèle plus complexe à plusieurs masses (voir l'onglet *Help* dans EMTP), mais le principe de fonctionnement demeure le même.

$$J \frac{d\omega_m}{dt} + D\omega_m = T_e - T_{ext} \quad (5.1)$$

où

J : moment d'inertie [kg m²/rad]

ω_m : vitesse de rotation mécanique [rad/s]

D : coefficient d'amortissement [Nm s/rad]

T_e : couple électromagnétique [Nm]

T_{ext} : couple externe [Nm]

La vitesse et le couple de la machine dépendent essentiellement de deux constantes et du couple externe. Le moment d'inertie représente une opposition à la rotation et il est directement relié à la construction de la machine. Cette valeur n'est généralement pas fournie par le manufacturier, comme c'est le cas pour les machines des bases de données provenant de l'IREQ et de ABB. Le

coefficient d'amortissement correspond à un couple de friction qui est proportionnel à la vitesse de rotation et qui s'oppose au couple électromagnétique de la machine. La valeur correspond à environ 2% du couple nominal de la machine à la vitesse nominale. Le couple externe représente quant à lui la charge qui est alimentée par la machine et il s'oppose également au couple électromagnétique. Ainsi, la variation de la vitesse de la machine est déterminée par les différents couples qui s'opposent au couple électromagnétique, qu'ils soient constants ou dépendants de la vitesse de rotation elle-même.

Puisque le moment d'inertie est généralement inconnu, une technique pour déterminer cette valeur qui se base sur l'essai de démarrage à vide est proposée. Comme son nom l'indique, cet essai consiste à démarrer la machine sans charge et à observer le temps nécessaire afin d'atteindre la vitesse nominale. Des données d'essais fournies par EDF pour une machine de 37 kW sont utilisées afin d'illustrer le problème. Par souci de confidentialité, le nom du constructeur n'est pas mentionné. Les spécifications de la machine, les paramètres obtenus avec le CGV ainsi que la courbe de la vitesse en fonction du temps pour le démarrage à vide sont présentées au Tableau 5.1 et au Tableau 5.2 ainsi qu'à la Figure 5.8.

Tableau 5.1 : Spécifications de la machine de 37 kW

V_n [V]	P_n [W]	T_n [Nm]	T_{st} [pu]	T_{br} [pu]	I_n [A]	I_{st} [pu]	pf_n	N_n [RPM]	N_s [RPM]
400	37000	238.4	2.69	3.10	69.3	8.21	0.844	1482	1500

Tableau 5.2 : Paramètres de la machine de 37 kW

R_{fe} [Ω]	R_s [Ω]	R_{r1} [Ω]	R_{r2} [Ω]	X_{sd} [Ω]	X_m [Ω]	X_{1d} [Ω]	X_{2d} [Ω]	I_{sat} [pu]	PS
32.50	0.10	0.13	0.13	$3e^{-6}$	10.50	0.57	0.57	1.7180	0.5

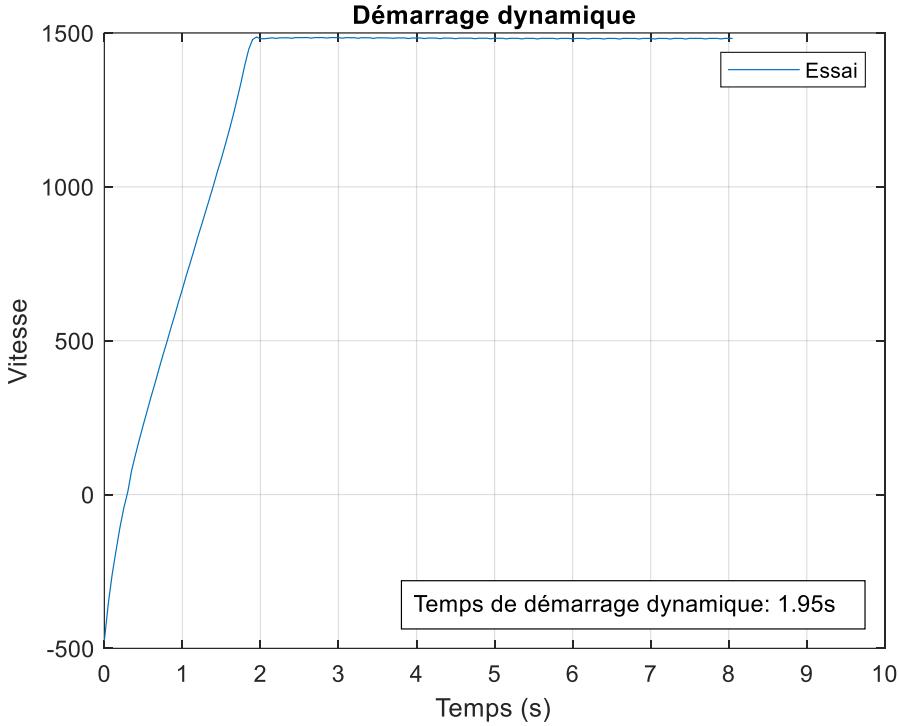


Figure 5.8 : Démarrage à vide

La machine atteint sa vitesse en régime permanent en environ 1.95s. En connaissant le temps de démarrage, il est possible de retrouver de manière analytique le moment d'inertie en posant certaines simplifications à l'équation (5.1).

$$T_{ext} = 0 \quad (5.2)$$

$$T_e = \text{constante} = T_{st} \quad (5.3)$$

D'abord, le couple externe est nul étant donné que la machine est à vide. Ensuite, le couple électromagnétique peut être considéré comme constant afin de simplifier le problème. Effectivement, le couple généré par la machine reste relativement stable pour des vitesses faibles et on peut approximer qu'il est environ égale au couple de démarrage sur cette plage de vitesse. Puisque la vitesse augmente graduellement, il s'agit d'une bonne approximation. L'équation (5.1) se simplifie donc à l'expression suivante :

$$J \frac{d\omega_m}{dt} + D\omega_m = T_e \quad (5.4)$$

Cette équation différentielle du premier ordre peut être résolue en considérant que la vitesse initiale est nulle. L'expression de la vitesse en fonction du temps devient alors :

$$\omega_m(t) = \frac{T_e}{D} \left(1 - e^{-\frac{D}{J}t} \right) \quad (5.5)$$

La seule valeur inconnue est le moment d'inertie. Les autres valeurs s'obtiennent à partir des spécifications du manufacturier.

$$T_e = T_{st} = 641.3 \text{ [Nm]} \quad (5.6)$$

$$D\omega_n = 0.02T_n$$

$$\Rightarrow D = \frac{0.02(238.4)}{1482(2\pi/60)} = 0.0307 \text{ [Nm s/rad]} \quad (5.7)$$

En isolant le moment d'inertie et en remplaçant les autres valeurs numériques, on trouve :

$$J = \frac{-Dt}{\ln \left(1 - \frac{D\omega_m}{T_e} \right)} = \frac{-0.0307(1.95)}{\ln \left(1 - \frac{0.0307(1482)(2\pi/60)}{641.3} \right)} = 8.0279 \text{ [kg m}^2/\text{rad]} \quad (5.8)$$

Ainsi, pour que la machine atteigne sa vitesse nominale en environ 1.95s, le moment d'inertie de la machine doit être de 8.0279 kg m²/rad pour un coefficient d'amortissement correspondant à 2% du couple nominal. Il est possible de comparer les résultats des essais avec ceux de la simulation à la Figure 5.9.

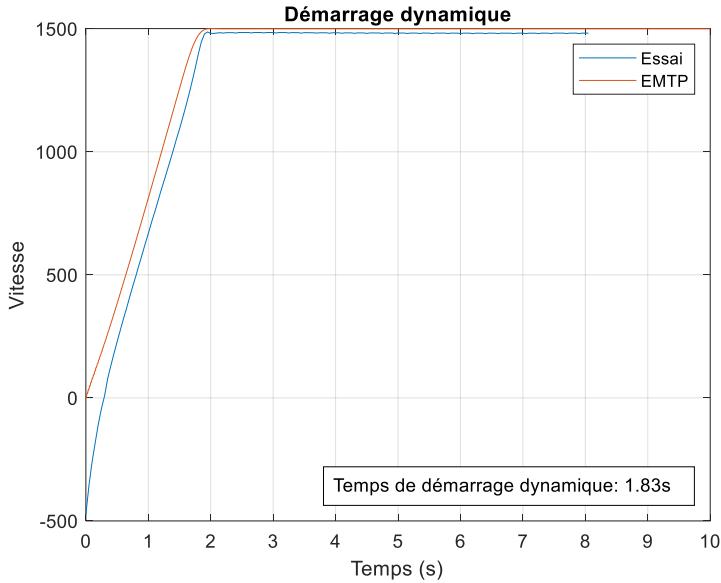


Figure 5.9 : Comparaison du temps de démarrage dynamique pour $D = 0.0307$

Le résultat est relativement proche de la valeur visée, soit un temps de 1.83s en simulation par rapport à un temps de 1.95s selon les données d'essais. Il est aussi possible de voir que la vitesse finale de la machine est légèrement plus élevée que celle de l'essai. Effectivement, puisqu'il n'y a aucun contrôle sur la machine, elle accélère sans cesse jusqu'à atteindre la vitesse synchrone, alors qu'elle se stabilise à la vitesse nominale dans le cas de l'essai. Il y a également un décalage au niveau des données d'essais, alors que la machine possède initialement une vitesse négative.

Dans le cas où l'amortissement est négligé, l'expression de la vitesse se simplifie encore plus. Il est alors possible de considérer que la variation de la vitesse est linéaire, ce qui est effectivement le cas pour la courbe présentée précédemment à la Figure 5.8 pour un démarrage à vide. En considérant encore une fois les simplifications énoncées précédemment et une vitesse initiale nulle, (5.1) devient :

$$J \frac{\Delta \omega_m}{\Delta t} = T_e \quad (5.9)$$

La valeur du moment d'inertie est alors :

$$J = \frac{T_e \Delta t}{\Delta \omega_m} = \frac{641.3(1.95)}{1482(2\pi/60)} = 8.0578 \quad (5.10)$$

Graphiquement, on retrouve un résultat pratiquement similaire, soit un temps de monté de 1.84s par rapport à 1.83s, tel qu'illustré à la Figure 5.10. Ceci indique que la simplification est valide et que le démarrage dynamique de la machine est quand même bien représenté.

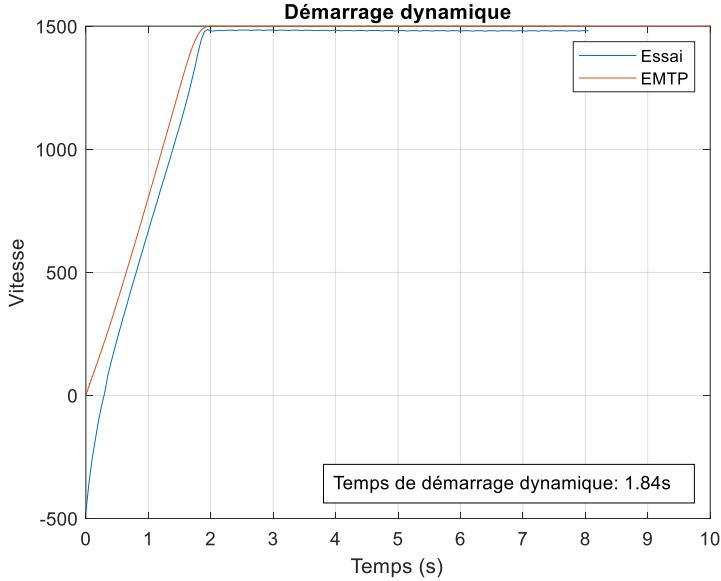


Figure 5.10 : Comparaison du temps de démarrage dynamique pour $D = 0$

Ainsi, il est possible d'approximer le moment d'inertie de la machine en se fiant aux données de l'essai de démarrage à vide. Il serait également possible de corrélérer ces résultats avec la puissance de la machine afin de fournir une valeur de base pour les machines qui ne possèdent pas de données d'essai. Le moment d'inertie est nécessaire afin d'observer le comportement transitoire de la machine lorsque la machine n'est pas contrôlée en vitesse.

En résumé, les transitoires sont causées autant par les paramètres de la simulation que par les paramètres électromécaniques de la machine. Il est donc primordial d'analyser les valeurs utilisées afin d'obtenir des résultats de simulation qui ont du sens.

CHAPITRE 6 CONCLUSION

En conclusion, un sommaire des contributions de ce mémoire sur la détermination des paramètres de la machine asynchrone, plus spécialement pour la simulation des transitoires électromagnétiques dans le logiciel EMTP, est présenté. Les limites et les contraintes du projet sont ensuite exposées, ce qui ouvre la porte à de nouveaux projets de recherche pour le futur.

Tout d'abord, le modèle analytique a été étudié et amélioré afin de représenter le plus fidèlement possible le comportement de la machine autant aux conditions de démarrage, de décrochage que nominales. Un modèle unique à cage double qui inclut les pertes fer sous la forme d'une résistance à l'entrée du modèle de même que la saturation variable des inductances de fuite permet de représenter la nature non-linéaire de la machine asynchrone. Ensuite, une fonction objectif qui considère un glissement nominal variable ainsi que l'algorithme d'optimisation PSO avec des bornes variables ont été utilisés afin de calculer les paramètres du modèle à partir des équations analytiques. Le calculateur à glissement variable (CGV) présente un taux de convergence d'environ 95% pour une banque de données d'une centaine de machines provenant de l'IREQ, ce qui est une nette amélioration par rapport au calculateur précédent qui avait un taux de convergence de 57%. La performance du CGV a d'ailleurs été confirmée avec une deuxième banque de données provenant du manufacturier ABB.

Le CGV a de la difficulté à bien identifier le couple de décrochage, ce qui résulte en un mauvais calcul des paramètres dans certains cas. L'inclusion de la classe NEMA de la machine pourrait potentiellement aider à régler ce problème. Dans tous les cas, le rapport entre le couple de démarrage et le couple de décrochage devrait être utilisé afin d'optimiser le calcul.

Une technique a été proposée afin de trouver une solution aux cas divergents, soit le retrait de la relation entre les paramètres. Toutefois, cette technique ne s'appuie sur aucune considération physique, si bien qu'elle est difficilement justifiable. L'étude des courbes du couple, du courant et du facteur de puissance démontre toutefois que cette solution est valide dans certains cas, alors que d'autres cas confirment plutôt la difficulté du CGV à calculer le couple de décrochage de manière adéquate.

Ensuite, les résultats analytiques ont été utilisés afin de vérifier le comportement de la machine en simulation, où il est possible d'observer qu'une certaine erreur se glisse entre les courbes simulées

et les courbes analytiques lorsque la partie saturable des inductances de fuite est différente de 0.5. Il aurait d'ailleurs été intéressant de pouvoir simuler le comportement de la machine en incorporant directement la partie saturable des inductances de fuite variable dans EMTP. Les erreurs dans la simulation semblent être dues à ce paramètre étant donné que la différence est surtout présente aux vitesses faibles où le courant et donc la saturation sont plus importants.

Pour régler ce problème, deux méthodes distinctes ont été présentées et testées. La première méthode évalue l'influence de chaque paramètre sur les spécifications, alors que la deuxième méthode utilise plutôt l'erreur obtenue lors de la simulation afin de recalculer les paramètres de la machine. La première méthode semble d'ailleurs plus désirable, à condition qu'elle soit automatisée. Il s'agit dans les deux cas d'une preuve de concept plutôt qu'une solution définitive et plus de travail doit certainement être fait à ce niveau afin de rendre le calcul plus performant.

Afin d'améliorer le modèle de la machine en simulation, il faut aussi se pencher sur la réponse transitoire. D'une part, les paramètres de la simulation ont été ajustés de manière à obtenir une bonne précision sans toutefois allonger inutilement la durée de la simulation. Dans tous les cas, il faut s'assurer que la machine a le temps de se stabiliser avant de faire varier la vitesse afin de limiter les oscillations. D'une autre part, la réponse dynamique de la machine, c'est-à-dire lorsqu'elle n'est pas contrôlée en vitesse, a également été analysée. Pour ce faire, un calcul qui se base sur le modèle à masse unique et sur le temps de démarrage à vide a été proposé afin de déterminer le moment d'inertie de la machine. Les données du manufacturier ne contiennent généralement aucune information à ce sujet, ce qui nécessite d'avoir recours à des essais avec la machine en tant que telle. Il faudrait voir s'il existe un test plus précis et surtout si les résultats présentés sont corrélés avec le comportement de la machine sous d'autres conditions d'opération.

Outre dans le dernier chapitre, les données d'essais ne sont pas prises en compte par le CGV. Il serait intéressant de valider le modèle en le comparant avec de réels résultats d'essais sur la machine plutôt qu'avec les spécifications du manufacturier seulement. Un outil qui compare les données d'essais avec la simulation et les équations analytiques a été développé, mais aucun travail n'a été fait afin de calculer les meilleurs paramètres pour ces machines.

RÉFÉRENCES

- [1] P. Kundur, *Power System Stability and Control*, 3rd^e éd. New York: McGraw-Hill, 1994.
- [2] F. Gauthier, "Calcul des paramètres de la machine asynchrone," Mémoire de maîtrise, École Polytechnique de Montréal, 2017. [En ligne]. Disponible: <https://publications.polymtl.ca/2940/>
- [3] Y. Seniani, "Modélisation avancée des alternateurs et des moteurs asynchrones," Rapport de stage de fin d'études, Université Paris-Saclay, 2019.
- [4] *IEEE Std 112-2017 - IEEE Standard Test Procedure for Polyphase Induction Motors and Generators*, 2018.
- [5] *IEC 60034-28:2012 - Rotating Electrical Machines - Part 28: Test Methods for Determining Quantities of Equivalent Circuit Diagrams for Three-Phase Low-Voltage Cage Induction Motors*, 2012.
- [6] J. Pedra, "On the Determination of Induction Motor Parameters From Manufacturer Data for Electromagnetic Transient Programs," vol. 23, n°. 4, p. 1709-1718, Nov 2008.
- [7] F. Corcoles *et al.*, "Analysis of the Induction Machine Parameter Identification," vol. 17, n°. 2, p. 183-190, 2002.
- [8] G. J. Rogers et D. Shirmohammadi, "Induction Machine Modelling for Electromagnetic Transient Program," vol. EC-2, n°. 4, p. 622-628, Dec 1987.
- [9] J. Pedra et F. Corcoles, "Estimation of Induction Motor Double-Cage Model Parameters from Manufacturer Data," vol. 19, n°. 2, p. 310-317, 2004.
- [10] S. A. Al-Jufout, W. H. Al-Rousan et C. S. Wang, "Optimization Induction Motor Equivalent Circuit Parameter Estimation Based on Manufacturer's Data," *Energies*, vol. 11, n°. 7, p. 13, Jul 2018.
- [11] G. F. V. Amaral *et al.*, "A High Precision Method for Induction Machine Parameters Estimation From Manufacturer Data," vol. 36, n°. 2, p. 1226-1233, 2021.
- [12] J. M. C. Guimaraes *et al.*, "Parameter Determination of Asynchronous Machines From Manufacturer Data Sheet," vol. 29, n°. 3, p. 689-697, Sep 2014.
- [13] A. H. Haque, "Determination of NEMA Design Induction Motor Parameters From Manufacturer Data," vol. 23, n°. 4, p. 997-1004, Dec 2008.
- [14] B. K. Johnson et J. R. Willis, "Tailoring Induction Motor Analytical Models to Fit Known Motor Performance Characteristics and Satisfy Particular Study Needs," vol. 6, n°. 3, p. 959-965, 1991.
- [15] M. A. Awadallah, "Parameter Estimation of Induction Machines from Nameplate Data Using Particle Swarm Optimization and Genetic Algorithm Techniques," vol. 36, n°. 8, p. 801-814, Aug 2008.
- [16] M. Calasan *et al.*, "Parameter Estimation of Induction Machine Single-Cage and Double-Cage Models Using a Hybrid Simulated Annealing-Evaporation Rate Water Cycle Algorithm," *Mathematics*, vol. 8, n°. 6, p. 29, Jun 2020.

- [17] R. Giri *et al.*, "Offline Parameter Estimation of Induction Motor Using a Meta Heuristic Algorithm," vol. 6466, p. 523, 2010.
- [18] S. C. Lima *et al.*, "Induction Motor Parameter Estimation from Manufacturer Data Using Genetic Algorithms and Heuristic Relationships," communication présentée à 2017 Brazilian Power Electronics Conference (COBEP), Nov 2017, p. 1-6.
- [19] H. R. Mohammadi et A. Akhavan, "Parameter Estimation of Three-Phase Induction Motor Using Hybrid of Genetic Algorithm and Particle Swarm Optimization," p. 148204 (6 pp.), 2014.
- [20] M. I. Mossad *et al.*, "A Novel Evolutionary Technique to Estimate Induction Machine Parameters from Name Plate Data," p. 66-71, 2016.
- [21] V. P. Sakthivel, R. Bhuvaneswari et S. Subramanian, "Artificial Immune System for Parameter Estimation of Induction Motor," vol. 37, n°. 8, p. 6109-6115, Aug 2010.
- [22] V. P. Sakthivel, R. Bhuvaneswari et S. Subramanian, "Bacterial Foraging Technique Based Parameter Estimation of Induction Motor from Manufacturer Data," vol. 38, n°. 6, p. 657-674, 2010.
- [23] O. Cetin, A. Dalcali et F. Temurtas, "A Comparative Study on Parameters Estimation of Squirrel Cage Induction Motors Using Neural Networks with Unmemorized Training," *Engineering Science and Technology-an International Journal-Jestech*, vol. 23, n°. 5, p. 1126-1133, Oct 2020.
- [24] M. G. Bijan *et al.*, "Induction Machine Parameter Range Constraints in Genetic Algorithm Based Efficiency Estimation Techniques," *Ieee Transactions on Industry Applications*, vol. 54, n°. 5, p. 4186-4197, Sep-Oct 2018.
- [25] I. Boldea, *Induction Machines Handbook: Transients, Control Principles, Design and Testing*, 3rd edition^e éd.: CRC Press, 2020.
- [26] L. Guasch-Pesquer *et al.*, "Parameters Calculation of Single- and Double-Cage Models for Induction Motors from Manufacturer Data," p. 237-242, 2015.
- [27] J. Kennedy et R. Eberhart, "Particle Swarm Optimization," communication présentée à Proceedings of the 1995 IEEE International Conference on Neural Networks. Part 1 (of 6), November 27, 1995 - December 1, 1995, Perth, Aust, 1995, p. 1942-1948.
- [28] *ANSI/NEMA MG 1-2016 - Motors and Generators*, 2016.

ANNEXE A – PARAMÈTRES DES FIGURES

Figure(s)	Description	Paramètres
2.5, 2.6, 3.3, 3.4	Courbes analytiques de la machine 1	calculateParam.m paramètres: param_ABB_2.mat data.database: ABB data.machine_number: 1 param.fixed_PS: variable param.relation: dependant
3.5	Courbes analytiques de la machine 10	calculateParam.m paramètres: param_ABB_2.mat data.database: ABB data.machine_number: 10 param.fixed_PS: variable param.relation: dependant
3.7, 3.8	Couple des cas divergents	calculateParam.m paramètres: param_ABB_ind.mat data.database: ABB data.machine_number: 17, 18, 49, 51 param.fixed_PS: variable param.relation: independant
4.3	Courbes simulées et analytiques de la machine 10	dynamic_calculateParam.m ASM_nominal.ecf paramètres: param_ABB_2.mat data.database: ABB data.machine_number: 10 param.fixed_PS: variable param.relation: dependant
4.4	Courbes simulées et analytiques de la machine 10 pour PS = 0.5	dynamic_calculateParam.m ASM_nominal.ecf paramètres: param_ABB10_PS05.mat data.database: ABB data.machine_number: 10 param.fixed_PS: 0.5 param.relation: dependant
4.5	Courbes simulées et analytiques de la machine 10 après correction – Méthode 1	dynamic_calculateParam.m ASM_nominal.ecf paramètres: param_ABB10_method1.mat data.database: ABB data.machine_number: 10 param.fixed_PS: variable param.relation: dependant
4.6	Courbes simulées et analytiques de la machine 10 après correction – Méthode 2	dynamic_calculateParam.m ASM_nominal.ecf paramètres: param_ABB10_method2.mat data.database: ABB

		data.machine_number: 10 param.fixed_PS: variable param.relation: dependant
5.1, 5.2, 5.3, 5.4	Oscillations du couple	dynamic_calculateParam.m ASM_nominal.ecf paramètres: param_ABB10_PS05.mat data.database: ABB data.machine_number: 10 param.fixed_PS: 0.5 param.relation: dependant
5.5, 5.6, 5.7	Transitoire et courbe discrète	nominal_discret.m ASM_nominal_discret.ecf paramètres: param_ABB10_PS05.mat data.database: ABB data.machine_number: 10 param.fixed_PS: 0.5 param.relation: dependant
5.8, 5.9, 5.10	Temps de démarrage dynamique	demarrage_dynamique.m ASM_demarrage_dynamique.ecf