

REMERCIEMENTS:

Je tiens à remercier le Professeur J.P. ROGNON, Directeur du Laboratoire d'Electrotechnique de Grenoble de non seulement m'avoir accueilli au sein de son laboratoire et de m'avoir donné l'opportunité d'élargir mes expériences dans la recherche, mais aussi pour m'avoir proposé ce sujet, d'avoir cru en moi et m'avoir encouragé. Sans son support je peux franchement dire que cette thèse n'aurait jamais été terminée.

J'adresse également mes sincères remerciements à Mr A. DOURADO PEREIRA CORREIA, Professeur à l'Université de Coimbra et à Mr J.P. HAUTIER, Professeur à l'Université de Lille et directeur du Laboratoire d'Electrotechnique et Electronique de Puissance, pour avoir accepté d'être rapporteurs de cette thèse et m'avoir fait l'honneur de leur présence dans ce jury et Mr. Daniel ROYE pour m'avoir fait l'honneur de présider le jury de cette thèse.

Je tiens aussi à remercier Mr. Ander ETXEBERRIA LARRAZABAL, Professeur à l'Université de Mondragon et co-encadrant de cette thèse, qui a aussi su m'encourager et m'aider quand j'en avais besoin.

J'ai voudrais remercier aussi tous les autres gens que j'ai côtoyés pendant cette thèse. Les remercier tous serait trop long mais je vais en citer quelques-uns, Oscar FRAILE et Nestor PIEROLA qui ont réalisé le montage des banc d'essais à Mondragon, Jacques DAVOINE qui à réalisé le dispositif expérimental à Grenoble, les diverses administrateurs du réseau du LEG (Etiennette, Bruno, Florence, Patrick et Patrick, Marie-Thérèse, ...) qui ont toujours été prêts à écouter et résoudre les problèmes du réseau informatique, tous les thésards du laboratoire pour leur bonne humeur et les bons temps que nous avons passé ensemble, les collègues au département d'électronique à Mondragon.

Je voudrais spécialement remercier Bertrand RAISON pour les transgressions linguistiques qu'il a du corriger et subir de ma part depuis les sept ans qu'on se connaît. C'est grâce à lui que je peux dire que je parle un peu mieux la langue française.

Finalement, je remercie mes parents et mes amis tant à Grenoble comme à Mondragon pour leur compréhension et leur support.

Table de matières.

INTRODUCTION.....	9
PROBLEMATIQUE DE L'USINAGE A TRES GRANDE VITESSE (UTGV).....	9
Les réseaux de neurones, la modélisation et la commande.....	12
ORGANISATION DU RAPPORT.....	17
BIBLIOGRAPHIE.....	18
CHAPITRE 1: MODELISATION, ANALYSE DES PERTURBATIONS ET QUALITE DU COUPLE D'UN AXE D'UNE MACHINE-OUTIL.....	21
1.1 INTRODUCTION.....	21
1.2 PARTIE MECANIQUE.....	21
1.2.1 <i>Transmission</i>	22
1.2.1.1 Flexibilité et résonances.....	23
1.2.1.2 Frottements et jeux.....	24
1.2.2 <i>Charge</i>	25
1.2.2.1 Variations de charge.....	25
1.2.2.2 Forces de coupe dans l'usinage.....	26
1.3 MOTEUR SYNCHRONE ET COMMANDE.....	26
1.3.1 <i>Modèle théorique de la machine</i>	26
1.3.2 <i>Perturbations de couple dues à la machine</i>	28
1.3.2.1 Anisotropie du rotor.....	28
1.3.2.2 Les harmoniques de FEM.....	29
1.3.2.3 Couple de crantage.....	30
1.3.3 <i>Perturbations de couple dues à l'onduleur et la commande</i>	31
1.3.3.1 Schémas.....	31
a) Alimentation continue.....	31
b) Onduleur.....	32
c) Commande.....	33
1.3.3.2 Influence de courants non sinusoïdaux.....	34
1.3.3.3 Influence de l'onduleur de tension et de la modulation de largeur d'impulsions (MLI).....	35
a) Influence du temps de garde.....	35
b) Influence de la MLI.....	37
1.3.3.4 Influence des variations de la tension de bus.....	37
1.3.3.5 Commande vectorielle.....	39
1.3.3.6 Influence des mesures.....	39
a) Non-synchronisme de la mesure des courants.....	40
b) Décalages des courants.....	41
c) Différences de gain des capteurs de courant.....	43
d) Discrétisation des signaux continus.....	45
1.3.3.7 Limitation de la tension à haute vitesse (surmodulation).....	46
1.4 BILAN.....	48
1.5 BIBLIOGRAPHIE.....	49
CHAPITRE 2: RESEAUX DE NEURONES POUR L'IDENTIFICATION DE L'AXE « EN TEMPS REEL ».....	51
2.1 INTRODUCTION.....	51
2.2 RESEAUX DE NEURONES.....	52

2.2.1	<i>Le neurone</i>	52
2.2.1.1	Bases biologiques.	52
2.2.1.2	Modélisation pratique d'un neurone.	53
2.2.2	<i>Types de réseaux de neurones</i>	55
2.2.3	<i>Les réseaux de neurones appliqués à notre cas</i>	57
2.2.4	<i>Modèle mathématique du réseau utilisé</i>	59
2.2.5	<i>Apprentissage des réseaux de neurones pour l'approximation</i>	61
2.2.5.1	Classification des méthodes d'apprentissage.	63
a)	Gestion de données : Apprentissage par lot ou incrémental.	63
b)	Actualisation des paramètres : Apprentissage en ligne, différé et hors ligne.	63
c)	Fondement théorique : Apprentissage déterministe, stochastique et adaptatif.	64
d)	Variation du nombre de paramètres : Apprentissage constructif et algorithmes de réduction ("pruning").	65
2.2.5.2	Méthodes d'apprentissage.....	65
2.2.5.3	Limitations et propriétés de l'apprentissage du réseau de neurones.	66
a)	Qualité des données d'apprentissage.....	66
b)	Nombre de paramètres du réseau.	67
c)	Types de fonctions et choix des fonctions de sortie.	67
d)	Minima locaux.	67
e)	Paralysie de l'apprentissage.	68
f)	Généralisation.	68
g)	Initialisation du réseau de neurones.	69
h)	Vitesse de convergence.	69
2.2.5.4	Méthode retenue.	70
a)	L'optimisation par moindres carrés avec la méthode de Levenberg-Marquardt.....	70
2.3	CONTRAINTES A PRIORI DES RESEAUX DE NEURONES POUR L'UTILISATION EN TEMPS REEL.	74
2.3.1	<i>Limitations de temps de calcul</i>	74
2.3.1.1	Méthodes d'apprentissage.....	74
2.3.1.2	Système à plusieurs périodes d'échantillonnage (apprentissage différé).	74
2.3.2	<i>Limitations de mémoire</i>	75
2.4	APPLICATION A L'IDENTIFICATION DE L'AXE.....	76
2.4.1	<i>Rôles du réseau de neurones</i>	76
a)	Commande de l'axe.....	76
b)	Réduction des oscillations de couple.....	76
c)	Identification de l'axe.....	76
2.4.2	<i>Identification de systèmes échantillonnés en utilisant des réseaux de neurones</i>	77
2.4.2.1	Principes de l'identification de modèles dynamiques.	77
2.4.2.2	Structure d'identification.	79
a)	Modèles paramétriques en forme d'espace d'état et de fonction de transfert.	79
b)	Modèle série-parallèle ou modèle parallèle.....	80
c)	Effet du bruit de mesure.	81
d)	Choix du type de fonction d'approximation.....	81
e)	Ordre du modèle (nombre de paramètres).	82
2.4.2.3	Importance de la persistance de l'excitation.	82
2.4.2.4	Valeurs initiales des paramètres du réseau de neurones.....	83
a)	Obtention des paramètres initiaux à partir d'un modèle.....	84
2.4.3	<i>Résultats de simulation</i>	86
2.4.3.1	Modèle du système linéaire	86
2.4.3.2	Modèles des systèmes non linéaires.....	87
2.4.3.3	Tableaux comparatifs des résultats.	87
a)	Identification du système linéaire avec un modèle linéaire.	88
b)	Identification du système linéaire avec un modèle non linéaire.	90

c) Identification du système non linéaire avec un réseau de neurones.....	92
d) Identification des systèmes avec des perturbations externes.....	94
2.5 BILAN.....	97
2.6 BIBLIOGRAPHIE.....	98
CHAPITRE 3: VALIDATION EXPERIMENTALE DES RESULTATS.....	101
3.1 INTRODUCTION.....	101
3.2 OUTILS DE SIMULATION.....	102
3.2.1 Modèles MATLAB/Simulink.....	102
3.2.1.1 Modèle simplifié.....	105
3.2.1.2 Modèle détaillé.....	106
3.3 MISE EN ŒUVRE DU BANC D'ESSAIS MOTEUR.....	107
3.3.1 Objectifs et description.....	107
3.3.2 Validation des modèles de simulation du moteur/variateur.....	109
3.3.2.1 Paramètres mécaniques.....	109
a) Inertie, frottement sec et frottement visqueux.....	109
3.3.2.2 Paramètres du moteur et du variateur.....	110
a) Paramètres du moteur FAGOR AXM 3.09.6.....	110
b) Paramètres du variateur FAGOR AXD 2.50.....	112
3.3.2.3 Comparaison des résultats expérimentaux et de simulation.....	113
3.4 MISE EN ŒUVRE DU BANC D'ESSAIS AVEC LA MACHINE-OUTIL.....	116
3.4.1 Objectifs et description.....	116
3.4.2 Validation des modèles de simulation mécanique.....	117
3.4.2.1 Paramètres mécaniques.....	117
3.4.2.2 Paramètres du système d'acquisition des données.....	121
3.4.2.3 Comparaison des résultats expérimentaux et de simulation.....	122
3.5 VALIDATION DES ALGORITHMES D'IDENTIFICATION.....	124
3.5.1 Identification et compensation du couple de crantage.....	124
3.5.2 Identification de la mécanique de l'axe.....	125
3.5.3 Utilisation « en temps réel » des réseaux de neurones.....	126
3.6 BILAN.....	129
3.7 BIBLIOGRAPHIE.....	130
CONCLUSIONS ET PERSPECTIVES.....	131
ANNEXE A: ETUDE DES HARMONIQUES DU COUPLE DE LA MACHINE SYNCHRONE.....	135
A.1 DEFINITIONS DE BASE.....	135
A.2 HARMONIQUES DE COUPLE AVEC UN COURANT NON SINUSOÏDAL.....	135
A.3 HARMONIQUES DE COUPLE AVEC UN COURANT AVEC DES FREQUENCES SUBHARMONIQUES.....	143
A.4 HARMONIQUES DE COUPLE EN PRESENCE D'UN DECALAGE DE MESURE DES COURANTS.....	145
A.5 HARMONIQUES DE COUPLE EN PRESENCE D'UNE ERREUR DU GAIN DE MESURE DES COURANTS.....	148
ANNEXE B: LA RETROPROPAGATION.....	153
ANNEXE C: TABLE DE FIGURES.....	155

Introduction.

Problématique de l'usinage à très grande vitesse (UTGV).

Lorsque nous parlons de l'usinage à très grande vitesse (UTGV), la définition utilisée habituellement est la suivante :

« Seront considérées comme grandes vitesses d'usinage celles qui entraînent, au niveau du contact entre l'outil et la pièce usinée, une élévation de la température suffisamment importante pour que les caractéristiques mécaniques du matériau usiné soient diminuées. »
[VIG-1991]

L'augmentation de la vitesse d'usinage entraîne l'augmentation proportionnelle des vitesses de rotation des axes en provoquant l'excitation des fréquences de résonance de la machine. Une solution consiste à changer les matériaux et/ou la structure de la machine. Par exemple, en augmentant la raideur de l'axe, les modes propres de la machine augmentent aussi et en choisissant des matériaux possédant un bon coefficient d'amortissement, nous pouvons réduire les oscillations.

Par ailleurs, l'usinage de formes complexes entraîne des problèmes de précision parce que les trajectoires de mouvement imposent aux différents axes de fonctionner en régime transitoire.

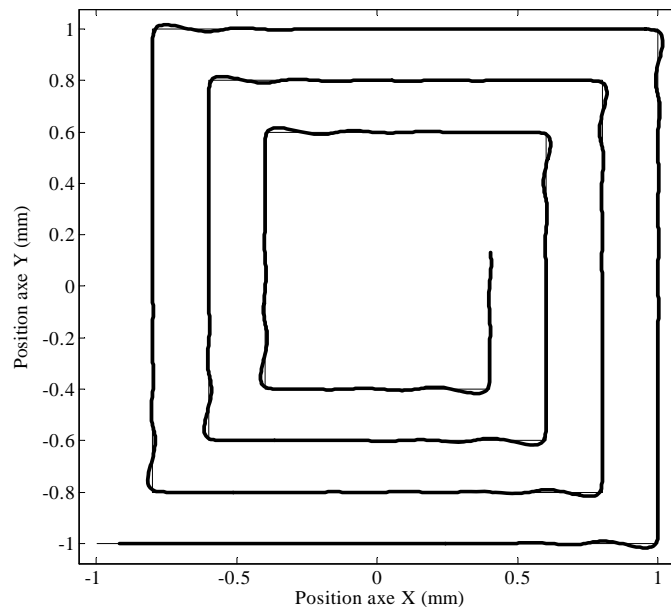


Figure I.1 : Erreur de poursuite dans le mouvement en spirale carrée sur les axes X et Y.

La figure I.1 montre l'évolution de la position des axes X et Y quand nous leur faisons suivre une trajectoire en spirale carrée à une vitesse constante. Nous pouvons constater que dans les coins de la spirale, il existe une erreur de poursuite due aux changements brutaux de direction.

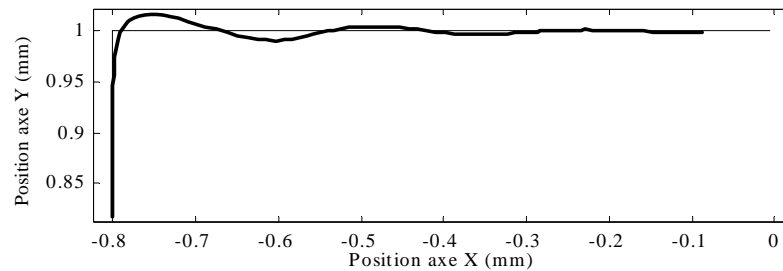


Figure I.2 : Détail de la trajectoire de la figure I.1.

La figure I.2 montre un agrandissement du coin supérieur gauche de la figure I.1. Nous pouvons observer une oscillation transitoire.

Il est évident que le gain réalisé sur la vitesse d'usinage ne doit pas se faire au détriment de la précision d'usinage.

La technique la plus utilisée pour diminuer l'erreur de poursuite consiste à calculer pour chaque axe les forces et les vitesses nécessaires pour parcourir les trajectoires souhaitées et à les injecter en références des correcteurs correspondants. De cette manière, les correcteurs de position et de vitesse ont simplement à corriger les écarts dus aux perturbations non prises en compte dans les calculs et aux imperfections des modèles de calcul.

Pour diminuer l'erreur de poursuite il faut également augmenter la bande passante des asservissements. Un certain nombre de phénomènes entrant alors dans la bande passante (dont les résonances mécaniques) ne peuvent plus être ignorés dans la synthèse des asservissements qui doit être réalisée à partir d'un modèle de l'axe prenant en compte ces phénomènes.

Enfin, la qualité de la surface usinée est surtout liée aux contrôles de vitesse et de couple. Sur ce dernier point, les imperfections du moteur d'axe, de son alimentation et des lois de commande doivent être minimisées voire connues et compensées.

Nous constatons donc, qu'en toute rigueur, il faut agir à tous les niveaux : structure et matériaux de la machine, conception du moteur électrique et de son alimentation, structure de la commande et synthèse des correcteurs de courant, de vitesse et de position pour résoudre le problème de manière satisfaisante.

Les solutions actuelles reposent sur un savoir faire lié à l'expérience des concepteurs de machines-outils et de systèmes contrôles numériques qui n'est pas très étendu hors du domaine.

Dans ce travail, nous avons fait le choix de nous intéresser plus particulièrement :

- A l'identification et la réduction des perturbations de couple générées par le moteur d'axe (synchrone dans notre cas), son alimentation et le dispositif de contrôle vectoriel.

Pour ce faire, nous avons fait une analyse exhaustive de ces perturbations (en particulier leur contenu spectral qui est à mettre en regard de la bande passante de l'asservissement de vitesse et des fréquences de résonance de la mécanique) : encoches stator, harmoniques du flux créés par le rotor dans les enroulements stator, harmoniques sur le bus continu d'alimentation de l'onduleur à MLI (modulation de largeur des impulsions), harmoniques de MLI, déséquilibres des mesures de courant, quantification des mesures, ...

Ce travail nous semblait justifié dans la mesure où, même si ces perturbations sont bien connues, leur importance relative par rapport aux problèmes rencontrés en machine-outil ne l'est pas, du moins à notre connaissance. Par ailleurs, nous souhaitons mettre en place une simulation suffisamment précise d'un axe de machine-outil pour pouvoir réellement valider sur cette simulation les dispositifs d'identification, de compensation et de contrôle qui font l'objet des points suivants.

- A l'identification en ligne de la mécanique de l'axe.

Nous nous sommes plus particulièrement intéressés aux modes résonants et à leurs variations avec la charge [BAR-1995]. Les forces de coupe sont également un paramètre de l'usinage peu connu. C'est cet aspect qui d'ailleurs actuellement le plus traité dans les études [OXL-1989][ALT-2000], mais nous ne l'avons pas abordé.

Une autre difficulté concerne le caractère non linéaire de la mécanique : nous connaissons bien évidemment les jeux (en général bien compensés dans les systèmes hautes vitesses) et les frottements, secs en particulier. Il faut y ajouter les variations d'inertie, de raideur, etc avec la position.

- A la synthèse de correcteurs de vitesse autoréglables rejetant les perturbations et à grande dynamique de poursuite pour garantir la qualité de suivi de trajectoire.

Dans la pratique, nous n'avons pas abordé ce troisième point.

Les réseaux de neurones, la modélisation et la commande.

Nous avons indiqué dans la partie précédente que les problèmes associés à l'usinage à vitesse élevée pouvaient être en partie corrigés en augmentant la dynamique des correcteurs. Pour cela, il nous faut :

- Des modèles généraux du comportement de la machine et/ou des perturbations.
- Les paramètres des modèles cités pour la machine/perturbation particulière.

Le grand inconvénient des modèles paramétriques est justement la difficulté d'obtention des valeurs des paramètres, éventuellement en ligne lorsqu'ils varient. Leur détermination directe est souvent impossible et il faut utiliser des méthodes indirectes (identification) pour les obtenir.

Dans ce contexte, il nous a semblé intéressant d'étudier l'apport que pouvaient avoir les réseaux de neurones artificiels.

Le neurone est le processeur élémentaire du cerveau. Chaque neurone traite l'information qui lui parvient localement, puis transmet aux autres neurones qui lui sont connectés l'information qu'il a traitée. Ces cellules peuvent apprendre en changeant l'intensité de leurs connexions avec d'autres cellules ou détruire ou même créer de nouvelles connexions. Elles peuvent aussi changer leur(s) règle(s) de traitement de l'information. Ce processus de changement est appelé apprentissage et joue un rôle fondamental dans le comportement du neurone.

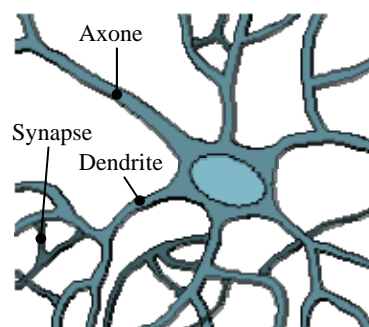


Figure I.3: Représentation d'un neurone réel.

La figure I.3 montre la représentation graphique d'un neurone. D'après la figure, nous pouvons observer que le neurone est composé des parties suivantes :

- **Corps cellulaire:** Noyau du neurone.
- **Dendrites:** Récepteurs principaux du neurone.
- **Axone:** Fibre nerveuse de transport pour les signaux émis par le neurone.
- **Synapse:** Connexion entre la dendrite et l'axone de deux neurones.

Les neurones artificiels reprennent cette structure comme le montre la figure I.4.

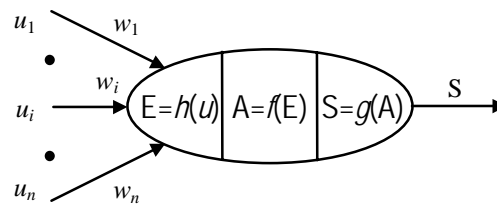


Figure I.4: Modèle général d'un neurone.

Où :

- u_i : Entrées du neurone ($i=1..n$).
- w_i : Coefficient de pondération de l'entrée i .
- $h(\bullet)$: Fonction d'entrée totale.
- $f(\bullet)$: Fonction d'activation.
- $g(\bullet)$: Fonction de sortie.
- $E=h(u_1, \dots, u_n)$: Entrée totale ou degré d'activation.
- $A=f(E)$: Etat du neurone.
- $S=g(A)$: Sortie.

Les réseaux de neurone artificiels (appelés réseaux de neurones par abus de langage) combinent les neurones selon différentes architectures en s'inspirant de manière assez libre de la structure du cerveau. Ils se divisent en deux classes majeures :

- Les réseaux à couches : les neurones sont groupés en couches, il n'y a pas d'interconnexion entre les neurones d'une même couche (voir figure I.5).

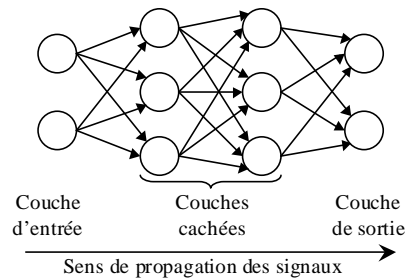


Figure I.5: Exemple d'un réseau de neurones avec une structure en couches.

- Les réseaux récurrents : des boucles fermées de connexion et de transmissions d'information entre tous les neurones sont permises (voir figure I.6).

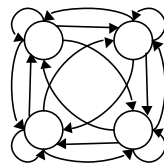


Figure I.6: Exemple d'un réseau de neurones récurrent.

Nous pouvons aussi trouver des réseaux récurrents à couches, avec des boucles entre les couches (combinaison des deux architectures précédentes, voir figure I.7).

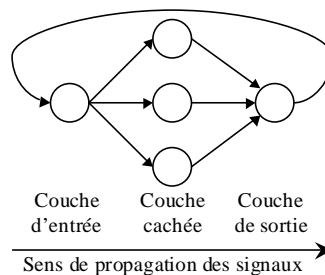


Figure I.7: Exemple d'un réseau de neurones en couches avec une récurrence entre couches.

D'après leur fondement théorique, ils sont capables d'approcher n'importe quelle fonction avec une précision arbitraire. De plus, les réseaux de neurones intègrent des méthodes d'apprentissage pour l'obtention des valeurs des paramètres.

Les domaines d'application des réseaux de neurones sont très étendus, voici quelques exemples [KEL-1997] :

Agroalimentaire :	Contrôle de cuisson dans un four micro-ondes. Analyse automatisé des odeurs dans le traitement de la nourriture (date de péremption). Contrôle de qualité des bières.
Assurance de qualité :	Classification des défauts de production. Vérification de pneumatiques.

	Contrôle de qualité de fabrication des microprocesseurs.
Détection des fraudes :	Dans l'usage des cartes de crédit. Dans l'usage des chèquiers. Vérification des signatures. Détection d'identité par l'analyse des habitudes des dépenses.
Finances :	Prévision de bourse. Gestions des fonds de pension.
Gestion des documents et imprimés :	Reconnaissance des caractères. Reconnaissance des visages. Reconnaissance des empreintes digitales. Reconnaissance des formes en général. Traitement automatisé des formulaires.
Marketing :	Serveur pour le placement des annonces « intelligents ». Optimisation des stratégies de marché. Prévision des demandes de transport par avion. Prévision de la consommation des boissons gazeuses. Sélection des usagers qui recevront des catalogues.
Médecine et santé :	Analyse des protéines pour le développement des drogues. Prédiction de la gravité de la maladie et l'usage des ressources de l'hôpital.
Production :	Contrôle de coulée continue. Contrôle des lamineurs. Contrôle et réduction de déchets dans la production des films photographiques.
Science et ingénierie :	Routage optimisé des circuits. Evaluation du timbre du voix et traitement d'audio. Spectroscopie et analyse des processus chimiques. Prévision de temps quantitative. Pilotage des véhicules autonomes.
Transport :	Détection des défauts dans le métro de London.

Tableau I.1: Domaines et applications du réseau de neurones.

Dans la reconnaissance et la prédiction le type de réseau utilisé est principalement le réseau à propagation par couches (voir figure I.5). Cependant quand il existe une structure temporelle

dans les données alors des variantes avec des récurrences sont souvent utilisées. Par exemple, les réseaux de neurones dits de Elman [ELM-1990] sont souvent utilisés pour les reconnaissances de voix (cas spécial du réseau présenté à la figure I.6). De même, les réseaux de type « Time Delay Neural Network » (voir figure I.7) sont aussi répandus grâce à la simplicité de mise en œuvre où l'on utilise une récurrence entre couches [WER-1990].

La plupart des applications sont des variantes de ces deux structures. Cependant, le pilotage des véhicules autonomes utilise des réseaux avec les structures présentées dans les figures I.5 à I.7 mais avec un algorithme d'apprentissage particulier. Dans ces applications, l'information fournie est la destination, mais le véhicule doit trouver le chemin pour arriver au but, éventuellement en évitant des obstacles et faisant une carte des chemins possibles pour y arriver par les méthodes d'essai et erreur.

Comme nous le montrerons au chapitre 2, compte tenu des objectifs de notre travail, nous avons opté pour un réseau non linéaire, avec ou sans récurrence, avec la fonction d'activation de type linéaire dans la couche de sortie et la fonction d'activation de type sigmoïde dans la couche cachée (voir §2.2.1.2 pour plus de détails).

Nous avons travaillé en plusieurs étapes :

- Pour ce qui concerne l'identification de l'axe :
 - Une analyse des structures et un parallèle avec les modèles linéaires ARX, OE, ARMAX ou leur extrapolation non linéaire.
 - Une étude théorique des performances des réseaux linéaires et non linéaires sur un modèle linéaire approché de la mécanique de l'axe et sur un modèle non linéaire emprunté à la littérature ; une comparaison avec les méthodes classiques d'identification.
 - Une validation en simulation sur le modèle détaillé de l'axe et expérimentale sur un axe de machine outil et l'analyse des résultats.
- Pour ce qui concerne l'identification des perturbations et leur compensation :
 - Une analyse de structure non récurrente.
 - La validation sur le modèle de simulation détaillé de l'axe.

- La validation sur le banc moteur et sur l'axe de machine outil et l'analyse des résultats à la lumière des difficultés particulières de mesure expérimentale.

Nous avons pu également vérifier si l'avance réalisée dans la puissance de calcul des processeurs de signal permet l'utilisation en temps réel des réseaux de neurones pour une application contraignante comme la commande des axes.

Organisation du rapport.

Le rapport est organisé en 3 chapitres. Le premier chapitre montre les perturbations présentes sur un axe de machine-outil et nous montrons qualitativement et/ou quantitativement leur importance dans la qualité de l'usinage en analysant la boucle de contrôle de couple.

Le deuxième chapitre présente les réseaux de neurones et explique les choix que nous avons réalisés pour la sélection du réseau utilisé en faisant ressortir les limitations de leur usage. Puis il montre comment utiliser les réseaux de neurones pour l'identification et les contraintes à respecter pour assurer une identification acceptable. Pour finir ce chapitre, nous appliquons ces techniques aux modèles de simulation pour valider les algorithmes utilisés.

Le troisième et dernier chapitre commence par décrire les plates-formes de simulation et expérimentales utilisées : le banc d'essais des moteurs et le banc d'essais de la machine, en donnant les paramètres que nous avons obtenus pour nos modèles de simulation et en validant ces modèles. Puis nous appliquons à nouveau les techniques d'identification présentées au chapitre précédent mais cette fois sur la machine réelle et nous présentons quelques résultats.

Bibliographie.

- [ALT-2000]: Y. ALTINTAS
Manufacturing Automation.
ISBN 0-521-65029-1. Cambridge University Press. 2000.
- [BAR-1995]: J.P. BARRE
Stratégies de commande pour un axe numérique de machine outil à usinage très grande vitesse.
Thèse de doctorat présentée à l'Université des Arts et Métiers. Centre de Lille. 1995.
- [BER-1993]: H. BERSINI, J.P. NORDVIK, A. BONARINI
A simple direct adaptive fuzzy controller derived from its neural equivalent.
Proceedings of the IEEE International Conference on Fuzzy Systems, pages 345-350. 1993.
- [ELM-1990]: J.L. ELMAN
Finding Structure in Time.
Cognitive Science, Vol.14, pages 179-211. 1990.
- [KEL-1997]: P.E. KELLER
Commercial applications of artificial neural networks.
URL: <http://www.emsl.pnl.gov:2080/proj/neuron/neural/products/>. 1997.
- [McC-1943]: W.S. McCULLOCH, W.H. PITTS
A logical calculus of ideas immanent in nervous activity.
Bulletin of Mathematical Biophysics, Vol.5, pages 115-133. 1943.
- [NAR-1990]: K.S. NARENDRA, K. PARTHASARATHY
Identification and control of dynamic systems using neural networks.
IEEE Transactions on Neural Networks, Vol.1, n°1, pages 4-27. 1990.
- [OXL-1989]: P.L.B. OXLEY
Mechanics of Machining.
ISBN 0-7458-0007-6. Jon Wiley & Sons. 1989.
- [REN-1994]: J.M. RENDERS
Algorithmes génétiques et réseaux de neurones.
Hermès. Paris. 1995.
- [ROG-1993]: J.S. ROGER JANG, C.T. SUN
Functional equivalence between radial basis function networks and fuzzy inference systems.
IEEE Transactions on Neural Networks, Vol.4, n°1, pages 156-159. 1993.
- [VIG-1991]: J. VIGNEAU
Usinage à très grande vitesse.
SNECMA. Oct. 1991, pp. 1-3.

[WER-1990]: P. WERBOS

Backpropagation through time: What it does and how to do it.
Proceedings of the IEEE, Vol.78, n°10, pages 1550-1560, 1990.

Chapitre 1: Modélisation, analyse des perturbations et qualité du couple d'un axe d'une machine-outil.

1.1 Introduction.

Dans ce chapitre, nous étudierons les différentes perturbations qui affectent le contrôle de couple et de vitesse d'un axe. Nous donnerons des modèles de chacune de ces perturbations. Ceci nous permettra l'évaluation de leur importance et nous donnera un renseignement au moment de décider si ces perturbations doivent être compensées ou non.

Le travail est présenté de la façon suivante : nous exposons tout d'abord les perturbations présentes dans la partie mécanique dues à la transmission et à la charge, puis nous présentons les perturbations dues à la machine synchrone. Nous étudions ensuite les perturbations dues à l'onduleur et la commande. Pour finir, nous faisons un bilan des conclusions obtenues.

1.2 Partie mécanique.

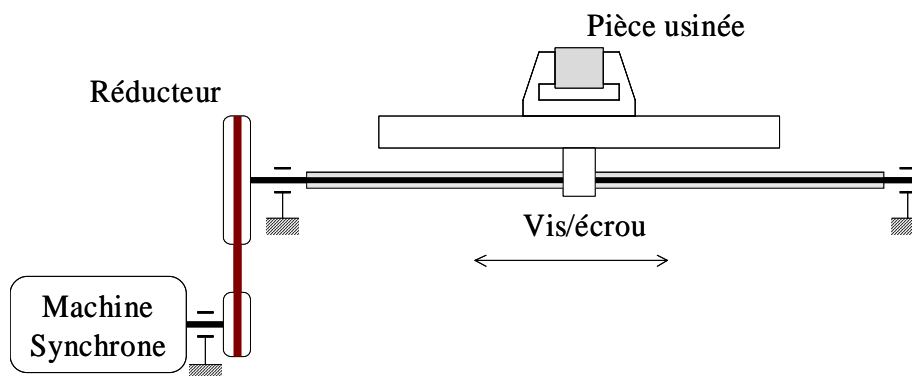


Figure 1.1: Modèle d'un axe de machine-outil.

A titre d'exemple, la figure 1.1 présente le modèle d'un axe de machine-outil comprenant un moteur, un réducteur à courroie et un ensemble vis/écrou entraînant une table supportant un outil de coupe. Sur ce schéma, les inerties du moteur, des poulies et de l'ensemble vis/écrou sont notées « J_{xx} », les masses de la courroie, des tables des axes X et Y et la charge sont notées « M_{xx} ». Les rigidités à la torsion, à la traction et à la compression sont représentées par des ressorts entre éléments et entre éléments et bâti. Les forces extérieures qui agissent sur le système, le couple généré par le moteur et les forces de coupe sur la pièce à usiner sont aussi représentés.

Dans une machine-outil réelle, il existe des couplages entre les axes en ce qui concerne les vibrations ; ces couplages sont suffisamment faibles pour, qu'en pratique, nous puissions les négliger.

1.2.1 Transmission.

Nous pouvons constater que le modèle analytique complet est très difficile à obtenir, car il nécessite une connaissance approfondie des éléments qui composent l'axe (inerties, masses, rigidité, etc.) et de leurs liaisons (jeux, frottements). Il est par ailleurs difficile à exploiter compte-tenu du nombre élevé de variables d'état.

Cependant, selon une étude réalisée par J.P.Barre [BAR-1995], pour la synthèse du correcteur de vitesse, un axe peut être dans la plupart des cas modélisé par un système simplifié comportant une ou deux résonances (figures 1.2 et 1.3). En effet, les autres résonances sont en dehors de la bande passante du correcteur et modifient peu la fonction de transfert de l'axe dans ce domaine de fréquences. Par conséquent, elles n'ont qu'un effet minimal sur la commande de l'axe.

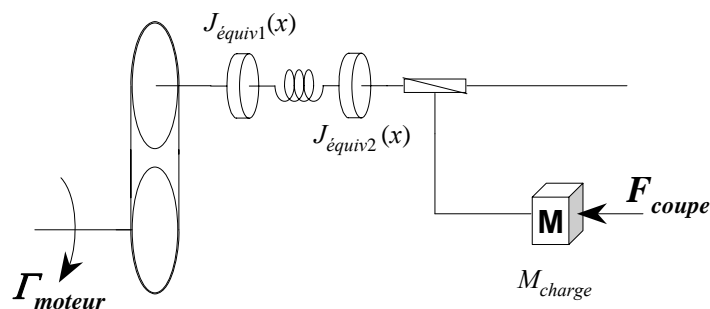


Figure 1.2: Modèle mécanique simplifié d'un axe de machine-outil avec une résonance.

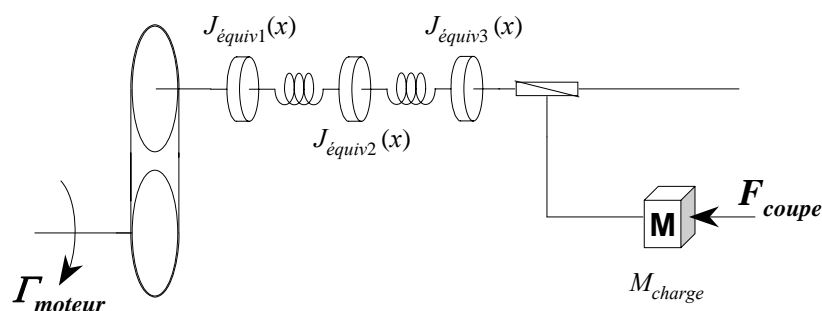


Figure 1.3: Modèle mécanique simplifié d'un axe de machine-outil avec deux résonances.

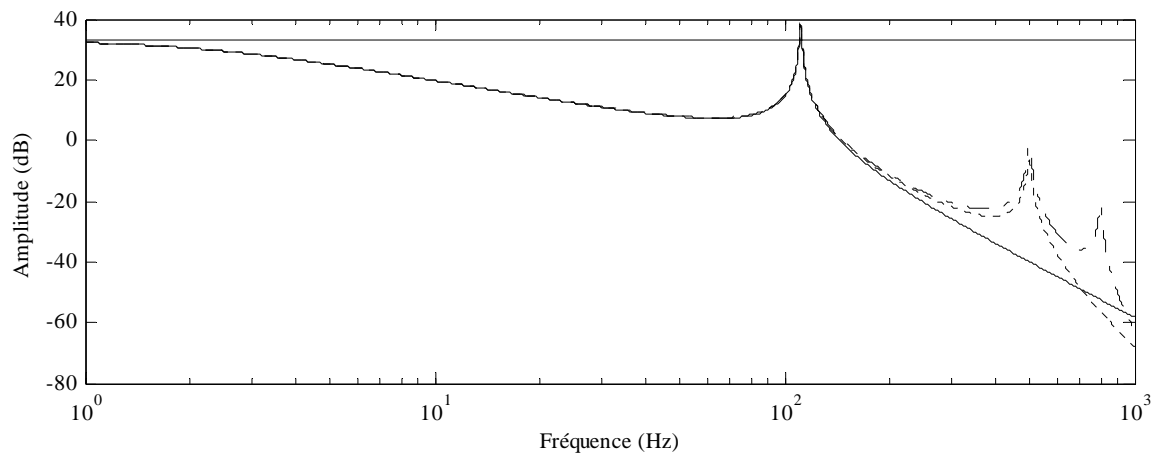


Figure 1.4: Diagramme de Bode du modèle complet et simplifié montrant l'approche de la réduction de l'ordre du modèle mécanique.

Sur la figure 1.4, nous pouvons observer le diagramme de Bode des modèles mécaniques initial et simplifiés linéarisés autour d'un point de fonctionnement (au milieu du parcours de l'ensemble vis/écrou). Le modèle simplifié avec deux fréquences de résonance approche correctement le système réel jusqu'à 6×10^2 rad/s (soit 200 Hz).

Maintenant que nous avons accepté qu'un modèle réduit puisse représenter assez fidèlement le modèle complet (au moins dans la zone où nous allons travailler), nous devons obtenir les paramètres qui le caractérisent. Il y a principalement deux groupes de méthodes pour les obtenir : la première repose sur une démarche analytique et requiert la connaissance approfondie de chaque élément, la deuxième est fondée sur la mesure directe des données expérimentales et l'identification des paramètres soit dans le domaine temporel soit dans le domaine fréquentiel. Les méthodes analytiques ne sont pas appropriées dans notre cas. En conséquence, nous utiliserons des méthodes d'identification à partir des données expérimentales.

1.2.1.1 Flexibilité et résonances.

En regardant attentivement la figure 1.1, nous constatons que l'inertie équivalente ramenée à l'axe du moteur n'est pas constante, mais qu'elle varie avec la position de l'ensemble vis/écrou. De la même manière, la rigidité de l'ensemble vis/écrou est variable en fonction de sa position. Nous pouvons en conclure que la fréquence de résonance et son amplitude varieront aussi en fonction de la position de l'ensemble vis/écrou. La figure 1.5 montre l'évolution de l'amplitude et la fréquence de résonance en fonction de la position de l'ensemble vis/écrou pour des paramètres approximatifs de la machine-outil. La variation de

la fréquence de résonance est de 92,6Hz à 235Hz quand la position de l'ensemble vis/écrou change de 0m jusqu'à 1,4m.

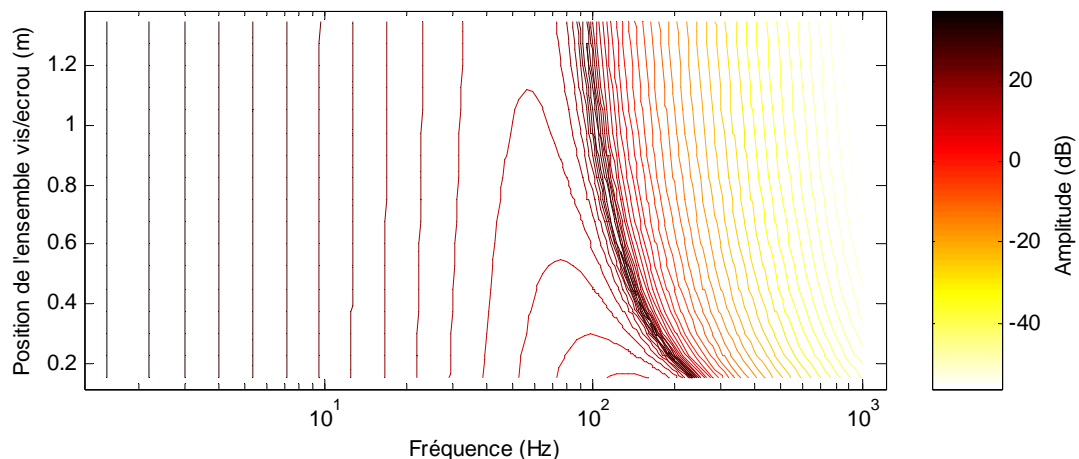


Figure 1.5: Amplitude et fréquence de résonance en fonction de la position de l'ensemble vis/écrou.

En pratique, dans le modèle mécanique utilisé pour la synthèse de la commande, nous considérons les fréquences de résonance constantes et nous utilisons un correcteur robuste par rapport à leurs variations.

1.2.1.2 Frottements et jeux.

Les frottements et les jeux sont une source considérable de problèmes compte-tenu de leur nature hautement non linéaire. Cependant, pour les machines-outils, ces problèmes sont moins importants car tout est fait dans leur conception pour les minimiser (choix de matériaux, lubrification, tolérances contraignantes,...). En tenant compte de ceci, il semble logique de négliger ces sources de perturbations dans le modèle de la partie mécanique.

Cependant, dans le cas où nous voudrions tenir en compte de ces non linéarités, nous considérons uniquement leur effet global.

Dans le cas des frottements, leur effet est représenté par un couple perturbateur ramené sur l'arbre moteur qui s'oppose au mouvement de l'axe (voir figure 1.6).

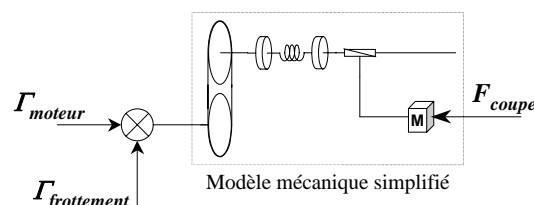


Figure 1.6: Représentation de l'effet global des frottements.

Les jeux sont représentés par une « zone morte » sur la position de la charge. La figure 1.7 montre la fonction entrée/sortie d'une zone morte. Cette approximation des jeux n'est pas exacte. Elle est néanmoins assez précise pour la simulation que nous voulons réaliser.

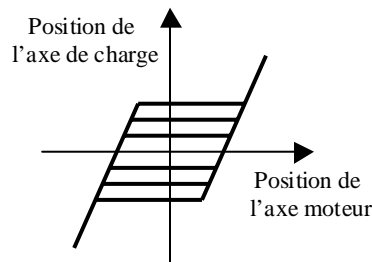


Figure 1.7: Représentation de l'effet des jeux.

1.2.2 Charge.

La charge est un élément inconnu dans l'usinage. Cependant, les contraintes physiques imposées par la machine ou par le constructeur limitent l'étude. Ainsi, les dimensions et la puissance de la machine imposent la charge maximale et minimale et les conditions de coupe et le matériau de la pièce nous donnent les paramètres des forces de coupe.

1.2.2.1 Variations de charge.

Un aspect important dans la modélisation de l'axe concerne les variations de charge. Ces variations peuvent influencer profondément la dynamique de l'axe et par conséquent le modèle équivalent. La figure 1.8 montre la variation de la résonance pour le modèle simplifié quand la charge varie de 1kg jusqu'à 100kg. La variation de la fréquence de résonance est de 106Hz à 122,3Hz quand la position de l'ensemble vis/écrou est fixée à 0,75m.

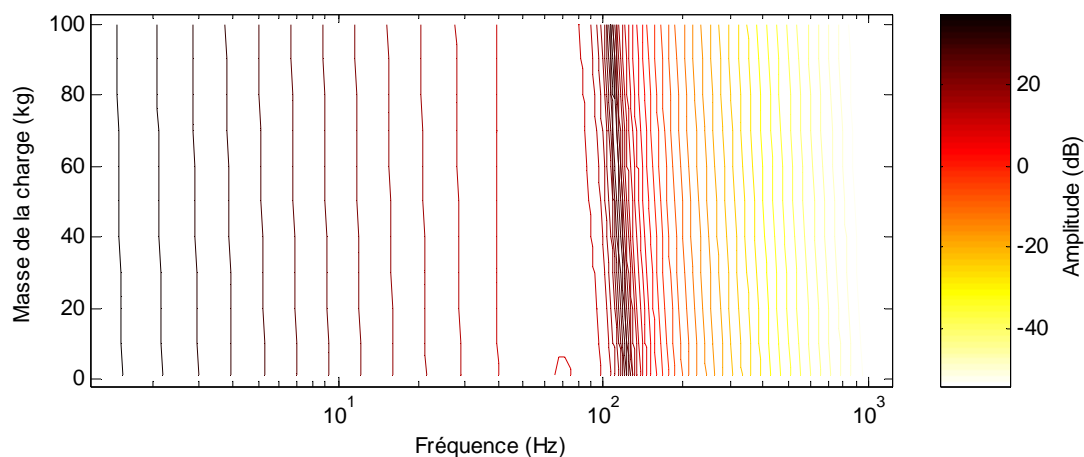


Figure 1.8: Amplitude et fréquence de résonance en fonction de la position de l'ensemble vis/écrou.

Nous pouvons constater que la variation d'inertie due à la charge avec les paramètres choisis reste négligeable par rapport à l'inertie de l'axe. Ainsi la variation de la fréquence de

résonance est également négligeable par rapport aux variations de charge. Cependant, dans les machines-outils pour l'usinage TGV, l'inertie de l'axe est réduite pour obtenir des vitesses plus grandes et, par conséquent, les variations des fréquences de résonance sont beaucoup plus importantes. Ceci illustre la nécessité d'un correcteur robuste vis-à-vis de ces variations.

1.2.2.2 Forces de coupe dans l'usinage.

La détermination des forces de coupe est sans doute la partie la plus difficile. Elles dépendent du type de matériel, de la vitesse d'avance, de la vitesse de coupe, de la lubrification, de l'état de l'outil de coupe et encore d'autres facteurs. De tous ces facteurs, seules les vitesses d'avance et de coupe sont facilement quantifiables. Nous constatons que l'obtention d'un modèle global des forces avec ces facteurs est pratiquement impossible. Par conséquent, il faut obtenir des modèles au cas par cas, soit par une approche « éléments finis », soit par un modèle empirique à partir d'essais.

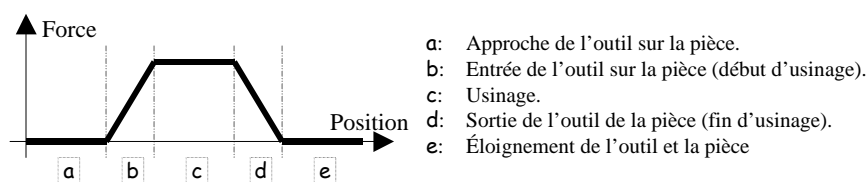


Figure 1.9: Evolution de la force de coupe sur un axe dans l'usinage.

Cette modélisation nécessite un travail très lourd [OXL-1989][ALT-2000]. Nous, nous contenterons d'obtenir des modèles très grossiers qui approchent le comportement moyen de ces forces. Par exemple, la figure 1.9 montre l'évolution de la force sur un axe au cours du processus d'usinage.

1.3 Moteur synchrone et commande.

Dans les asservissements des machines-outils, la machine synchrone à aimants permanents alimentée par onduleur MLI et associée à un dispositif de contrôle vectoriel du couple est couramment utilisée. Dans cette partie, nous analyserons les sources des oscillations provoquées par la machine, l'onduleur ou le dispositif de contrôle vectoriel.

1.3.1 Modèle théorique de la machine.

Hypothèse préalable: la saturation magnétique peut être négligée compte-tenu de la technologie des moteurs d'axe installés sur la machine outil.

Soient:

$$\mathbf{i}_s = \begin{bmatrix} i_{sa} \\ i_{sb} \\ i_{sc} \end{bmatrix} \quad \boldsymbol{\varphi}_s = \begin{bmatrix} \varphi_{sa} \\ \varphi_{sb} \\ \varphi_{sc} \end{bmatrix} \quad \mathbf{v}_s = \begin{bmatrix} v_{sa} \\ v_{sb} \\ v_{sc} \end{bmatrix} \quad \mathbf{k}(\theta) = \begin{bmatrix} k_a(\theta) \\ k_b(\theta) \\ k_c(\theta) \end{bmatrix} \quad (1.1)$$

Les vecteurs:

- courants dans les trois phases statoriques,
- flux créés par les courants dans les phases statoriques,
- tensions aux bornes des phases statoriques
- flux créés par les aimants permanents rotoriques dans les phases statoriques,

Où θ représente la position électrique du rotor par rapport à l'axe de la phase a.

Par ailleurs, les matrices des résistances et inductances statoriques s'écrivent:

$$\mathbf{R}_s = \begin{bmatrix} R_{sa} & 0 & 0 \\ 0 & R_{sb} & 0 \\ 0 & 0 & R_{sc} \end{bmatrix} \quad \mathbf{L}_s(\theta) = \begin{bmatrix} L_{sa}(\theta) & M_{sab}(\theta) & M_{sca}(\theta) \\ M_{sab}(\theta) & L_{sb}(\theta) & M_{sbc}(\theta) \\ M_{sca}(\theta) & M_{sbc}(\theta) & L_{sc}(\theta) \end{bmatrix} \quad (1.2)$$

Les équations différentielles de la machine synchrone sous forme matricielle s'écrivent alors comme suit:

$$\begin{aligned} \boldsymbol{\varphi}_s &= \mathbf{L}_s(\theta) \mathbf{i}_s + \mathbf{k}(\theta) \\ \mathbf{v}_s &= \mathbf{R}_s \mathbf{i}_s + \frac{d\boldsymbol{\varphi}_s}{dt} \end{aligned} \quad (1.3)$$

L'équation du couple est la suivante (obtenu à partir de l'énergie de la machine):

$$\Gamma = p \underbrace{\frac{1}{2} \mathbf{i}_s^T \frac{d\mathbf{L}_s(\theta)}{d\theta} \mathbf{i}_s}_{\Gamma_{réductant}} + p \underbrace{\mathbf{i}_s^T \frac{d\mathbf{k}(\theta)}{d\theta}}_{\Gamma_{flux}} - p \underbrace{\frac{dW_{rotor}(\theta)}{d\theta}}_{\Gamma_{crantage}} \quad (1.4)$$

Si nous observons l'équation de couple (1.4), nous distinguons trois termes.

Le premier terme $\Gamma_{réductant}$ correspond aux variations de la réluctance d'entrefer en fonction de la position du rotor par rapport au champ créé par les courants statoriques et consécutives à la présence d'une anisotropie au rotor.

Le deuxième Γ_{flux} correspond à l'interaction des champs créés par les courants statoriques et par les aimants permanents du rotor.

Le troisième $\Gamma_{crantage}$ correspond aux variations de la réluctance d'entrefer en fonction de la position du champ rotorique par rapport au stator et consécutives à la présence d'encoches au stator.

Pour éviter d'exciter les résonances mécaniques de la chaîne de transmission, il faut éviter dans la mesure du possible que le couple de la machine comporte des composantes harmoniques dans les domaines de fréquences correspondants. Les oscillations de couple peuvent être dues:

- Aux imperfections de la machine elle-même: décentrement du rotor, encoches stator, harmoniques du flux créé par le rotor dans le stator,
- Ou/et à celles de l'alimentation et du contrôle vectoriel: MLI, erreurs de décalage et de gain des capteurs de courant, quantification, performance des boucles de contrôle des courants.

1.3.2 Perturbations de couple dues à la machine.

Les courants statoriques sont supposés sinusoïdaux:

$$\mathbf{i}_s = \begin{bmatrix} i_{sa} \\ i_{sb} \\ i_{sc} \end{bmatrix} = \begin{bmatrix} i_{s0} \cos(\theta + \delta) \\ i_{s0} \cos(\theta + \delta - 2\pi/3) \\ i_{s0} \cos(\theta + \delta - 4\pi/3) \end{bmatrix} \quad (1.5)$$

Où δ représente le déphasage entre le champ créé par les aimants permanents du rotor et le champ créé par les courants statoriques.

1.3.2.1 Anisotropie du rotor.

L'anisotropie du rotor affecte l'allure des inductances du stator. Ainsi, pour une machine à pôles lisses, les inductances et les mutuelles statoriques sont constantes. Cependant, si la machine est à pôles saillants ou à pôles enterrés, les inductances varient en fonction de la position du rotor car l'entrefer n'est plus constant.

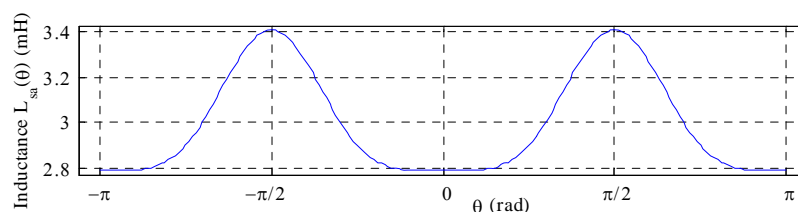


Figure 1.10: Allure de l'inductance statorique pour une machine synchrone à pôles enterrés.

La figure 1.10 montre une forme typique de l'inductance statorique $L_{sa}(\theta)$ pour une machine à pôles enterrés ou à saillance inverse.

L'équation de cette courbe a la forme suivante:

$$L_{sa}(\theta) = L_{s0} + L_{s2} \cos(2\theta) + L_{s4} \cos(4\theta) + \dots \quad (1.6)$$

Ainsi, la matrice d'inductances devient:

$$\mathbf{L}_s(\theta) = \begin{bmatrix} L_{s0} & M_{s0} & M_{s0} \\ M_{s0} & L_{s0} & M_{s0} \\ M_{s0} & M_{s0} & L_{s0} \end{bmatrix} + \sum_{k=1}^{\infty} L_{2k} \begin{bmatrix} \cos(2k\theta) & \cos[2k(\theta + 2\pi/3)] & \cos[2k(\theta - 2\pi/3)] \\ \cos[2k(\theta + 2\pi/3)] & \cos[2k(\theta - 2\pi/3)] & \cos(2k\theta) \\ \cos[2k(\theta - 2\pi/3)] & \cos(2k\theta) & \cos[2k(\theta + 2\pi/3)] \end{bmatrix} \quad (1.7)$$

$L_{s0} = l_0 + l_{\sigma}$ et $M_{s0} = -l_0/2$, où l_0 et l_{σ} sont respectivement l'auto-inductance et l'inductance de fuites de la bobine.

Si nous considérons la partie du couple $\Gamma_{réductant}$ due à la variation de réductance, nous obtenons la formule suivante:

$$\begin{aligned} \Gamma_{réductant} &= p \frac{1}{2} \mathbf{i}_s^T \frac{d\mathbf{L}_s(\theta)}{d\theta} \mathbf{i}_s = \\ &= p \frac{9}{8} \left[-2i_{s0}^2 L_2 \sin(2\delta) + i_{s0}^2 \sum_{k=1}^{\infty} [(6k-2)L_{6k-2} \sin(6k\theta + 2\delta) + (6k+2)L_{6k+2} \sin(6k\theta - 2\delta)] \right] \end{aligned} \quad (1.8)$$

Dans cette équation, nous constatons l'existence d'harmoniques de couple à des fréquences multiples de 6 fois la fréquence des courants statoriques et provenant des harmoniques 4, 8, 10, 14, ... de la matrice d'inductances. Les harmoniques 6, 12, 18, ... des inductances n'interviennent pas dans la génération de couple. Cela dit, avec les méthodes modernes de conception de moteurs, les constructeurs arrivent à minimiser les harmoniques de l'inductance. Par conséquent, l'effet de ces harmoniques est souvent négligeable pour les machines synchrones à aimants permanents.

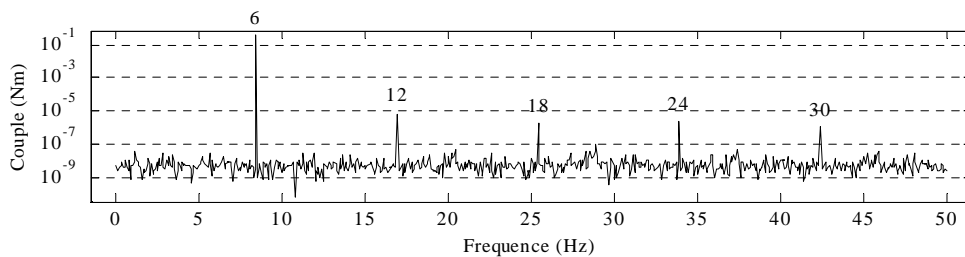


Figure 1.11: Harmoniques de couple en présence des harmoniques de l'inductance statorique ($F_s=1,41\text{Hz}$).

La figure 1.11 présente les harmoniques de couple en présence des harmoniques de l'inductance statorique sachant que le couple moyen est de 7,9Nm ($L_0=17,33\text{mH}$, $L_2=1,733\text{mH}$, $L_4=0,866\text{mH}$, $i_{s0}=5\text{A}$ et $\delta=0^\circ$).

1.3.2.2 Les harmoniques de FEM.

Quand la répartition du flux créé par les aimants dans le stator n'est pas sinusoïdale, la force électromotrice (FEM) créée par ce flux est également non sinusoïdale. Si nous décomposons le flux en série de Fourier,

$$\mathbf{k}_s(\theta) = \begin{bmatrix} k_a(\theta) \\ k_b(\theta) \\ k_c(\theta) \end{bmatrix} = k_1 \begin{bmatrix} \cos(\theta) \\ \cos(\theta - 2\pi/3) \\ \cos(\theta - 4\pi/3) \end{bmatrix} + k_3 \begin{bmatrix} \cos(3\theta) \\ \cos[3(\theta - 2\pi/3)] \\ \cos[3(\theta - 4\pi/3)] \end{bmatrix} + k_5 \begin{bmatrix} \cos(5\theta) \\ \cos[5(\theta - 2\pi/3)] \\ \cos[5(\theta - 4\pi/3)] \end{bmatrix} + \dots \quad (1.9)$$

nous pouvons calculer les oscillations qui apparaissent sur la composante du couple Γ_{flux} .

$$\begin{aligned} \Gamma_{flux} &= p \left(\mathbf{i}_s^T \frac{d\mathbf{k}(\theta)}{d\theta} \right) \\ &= \frac{3}{2} p \left[-i_{s0} k_1 \cos(\delta) + i_{s0} \sum_{z=1}^{\infty} [(6z-1)k_{6z-1} \cos(6z\theta + \delta) - (6z+1)k_{6z+1} \cos(6z\theta - \delta)] \right] \end{aligned} \quad (1.10)$$

D'après l'équation (1.10), nous observons l'apparition d'harmoniques de couple à des fréquences multiples de 6 fois la fréquence des courants statoriques et provenant des harmoniques 5, 7, 11, 13, ... du flux rotorique. Les harmoniques 3, 9, 15, ... du flux rotorique n'interviennent pas dans la génération de couple. Pour ce couple également, les méthodes modernes de conception de machines permettent de minimiser les harmoniques et nous permettent de négliger ses effets.

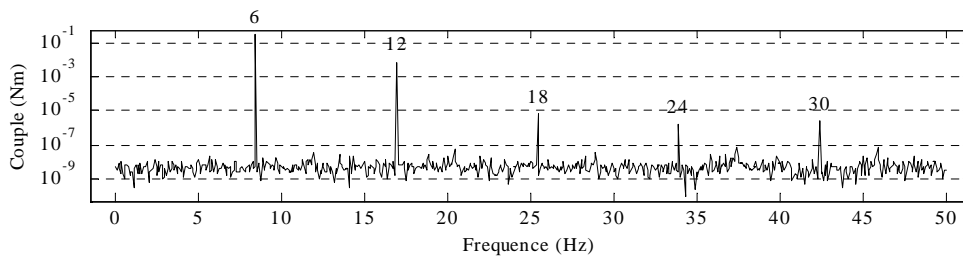


Figure 1.12: Harmoniques de couple en présence des harmoniques de FEM ($F_s=1,41\text{Hz}$).

La figure 1.12 montre les harmoniques de couple en présence des harmoniques de FEM.

1.3.2.3 Couple de crantage.

Le couple de crantage est un couple réductant créé par les encoches du bobinage statorique. En effet, l'énergie magnétique potentielle est maximale quand le pôle est aligné avec une encoche statorique. Par conséquent, la machine génère un couple pour s'aligner avec une dent d'encoche. La conception des machines (pas d'encoches, inclinaison des encoches) permet théoriquement de minimiser ce couple. En pratique, les tolérances de fabrication sont telles qu'il existe une grande dispersion d'une machine à l'autre. Ce couple est la principale source de perturbation en pratique.

L'obtention de la fonction reliant le couple à la position nécessite soit l'utilisation d'une méthode par éléments finis, soit un très long développement théorique. Nous avons décidé de

retrouver expérimentalement la fréquence fondamentale de ce couple et d'identifier les paramètres inconnus:

$$\Gamma_{crantage} = p \left(- \frac{dW_{rotor}(\theta)}{d\theta} \right) = \Gamma_{cr} \cos \left(\frac{N_e}{p} \theta + \delta_{cr} \right) \quad (1.11)$$

Dans l'équation (1.11) N_e , Γ_{cr} et δ_{cr} représentent le nombre d'encoches du stator, l'amplitude du couple de crantage et le déphasage du couple de crantage par rapport au flux rotorique.

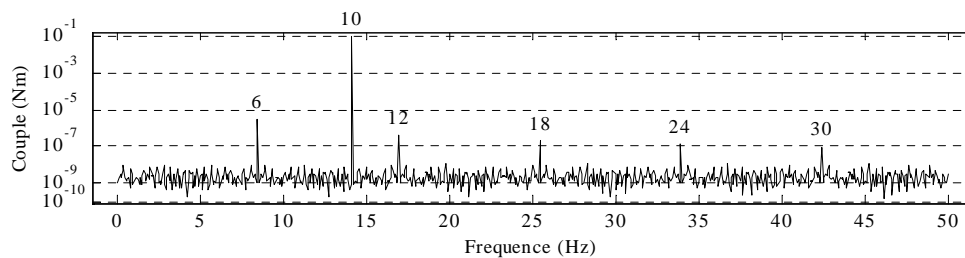


Figure 1.13: Harmoniques de couple causés par le couple de crantage ($F_s=1,41\text{Hz}$, $N_e/p=10$).

La figure 1.13 montre les harmoniques de couple en présence du couple de crantage.

Le couple de crantage dépend surtout de la position du rotor de la machine. Cependant, il dépend probablement aussi de la vitesse de rotation du moteur à cause du délai de propagation magnétique (changement du déphasage). Ce dont nous pouvons être sûrs est qu'il dépend de l'état de saturation magnétique de la machine, parce que quand les dents d'encoche sont saturées, la réluctance augmente et l'amplitude de couple diminue. Toutefois, dans cette étude, nous avons supposé que la machine n'était pas saturée.

1.3.3 Perturbations de couple dues à l'onduleur et la commande

1.3.3.1 Schémas

a) Alimentation continue.

La figure 1.14 montre le dispositif de redressement, de filtrage de la tension du bus continu et de limitation de cette tension lors des phases de freinage par insertion d'une résistance de dissipation. La commande de limitation de la tension de bus est réalisée par un simple correcteur à hystérésis. Ce correcteur déclenche le transistor de freinage T_f quand la tension de bus dépasse un seuil haut et bloque le transistor quand la tension redescend au-dessous d'un seuil bas.

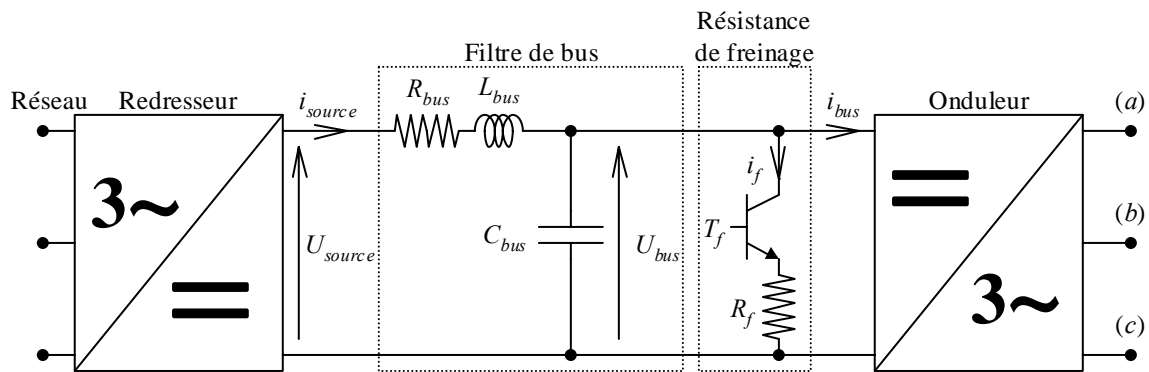


Figure 1.14: Filtrage de bus et résistance de freinage.

b) Onduleur.

La figure 1.15 montre la structure d'un onduleur de tension. Nous pouvons voir les trois bras identiques de l'onduleur qui servent à commander la tension sur chaque phase. La tension de bus est notée U_{bus} et les tensions de sortie de l'onduleur sont repérées par rapport au point neutre fictif de l'onduleur n_0 . Dans le modèle, nous avons considéré des interrupteurs idéaux et nous avons tenu compte des temps de garde ménagés entre les commandes des deux transistors d'un même bras (voir §1.3.3.3a).

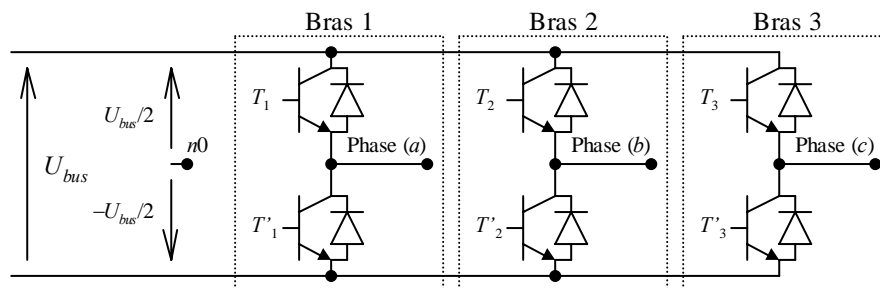


Figure 1.15: Structure de l'onduleur de tension.

La modulation de largeur d'impulsion (MLI) est obtenue par intersection des trois références de tension β_i avec une porteuse en dent de scie (MLI intersective) (voir figure 1.16). Le rafraîchissement des références est synchronisé avec le passage à chaque extremum de la porteuse. Il s'effectue donc à une fréquence double de la fréquence de MLI.

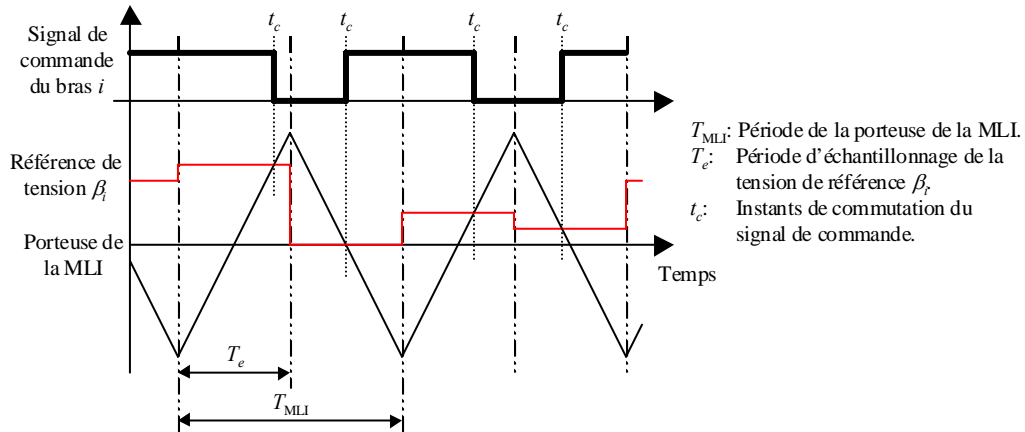


Figure 1.16: Modulation des références de tension avec la porteuse de MLI.

c) Commande

La figure 1.17 montre la structure de la commande. Dans cette structure, nous pouvons distinguer les blocs suivants:

1. Mesure de la tension de bus, des courants, de la position et calcul de la vitesse.
2. Transformation de Park et Transformation inverse de Park
3. Correcteur de courant.
4. Compensation des perturbations de la machine.
5. Limitation de la tension demandée à l'onduleur de tension.

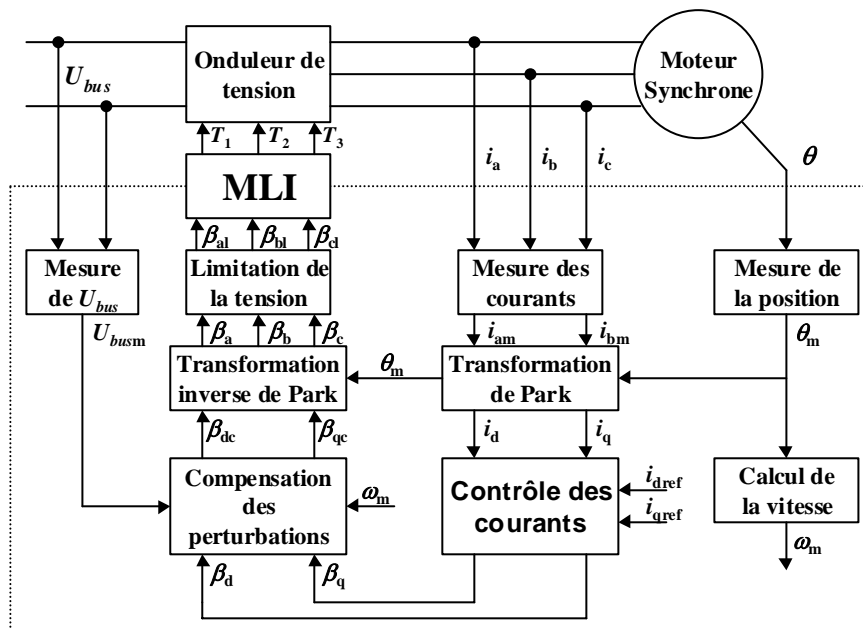


Figure 1.17: Structure de la commande vectorielle de la machine synchrone.

Le contrôle des courants est réalisé par deux correcteurs digitaux indépendants avec un dispositif de compensation des variations de la tension de bus U_{bus} obtenue en divisant la valeur de la référence de la tension par la tension de bus mesurée (voir figure 1.18).

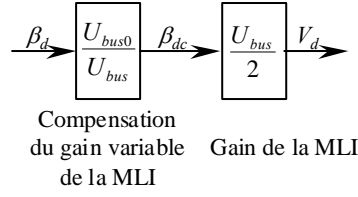


Figure 1.18: Compensation des variations de la tension de Bus.

1.3.3.2 Influence de courants non sinusoïdaux

Nous allons supposer que les imperfections du moteur et du variateur font apparaître des courants non sinusoïdaux, tels que:

$$\mathbf{i}_s = \begin{bmatrix} i_{sa} \\ i_{sb} \\ i_{sc} \end{bmatrix} = \sum_{n=1}^{\infty} i_n \begin{bmatrix} \sin[n(\theta) + \delta_n] \\ \sin[n(\theta - 2\pi/3) + \delta_n] \\ \sin[n(\theta - 4\pi/3) + \delta_n] \end{bmatrix} \quad (1.12)$$

$$i_{3k} = 0, \quad \forall k \in \mathbb{N}^+$$

δ_n représente le déphasage de l'harmonique n par rapport au champ créé par les aimants permanents du rotor.

Nous allons séparer les différents termes de l'équation du couple. La partie du couple correspondant à l'interaction des champs créés par les courants statoriques et par les aimants permanents du rotor Γ_{flux} est la suivante (voir dans l'annexe A pour avoir plus de détails sur le développement):

$$\Gamma_{flux} = \frac{3}{2} p \sum_{k=1}^{\infty} \sum_{m=1}^{\infty} \begin{bmatrix} - (6k-5) \cdot k_{(6k-5)} i_{(3m-2)} \cos[(3m-6k+3)\theta + \delta_{(3m-2)}] + \\ (6k-1) \cdot k_{(6k-1)} i_{(3m-2)} \cos[(3m+6k-3)\theta + \delta_{(3m-2)}] + \\ (6k-5) \cdot k_{(6k-5)} i_{(3m-1)} \cos[(3m+6k-6)\theta + \delta_{(3m-1)}] - \\ (6k-1) \cdot k_{(6k-1)} i_{(3m-1)} \cos[(3m-6k)\theta + \delta_{(3m-1)}] \end{bmatrix} \quad (1.13)$$

La partie du couple correspondant à la variation de l'inductance $\Gamma_{réductant}$ est:

$$\Gamma_{réductant} = p \frac{9}{8} \sum_{k=1}^{\infty} \sum_{o=1}^{\infty} \sum_{z=1}^{\infty} \begin{bmatrix} (6k-4)L_{(6k-4)} i_{(3o-2)} i_{(3z-2)} \sin[(6k-3o-3z)\theta - \delta_{(3o-2)} - \delta_{(3z-2)}] + \\ (6k-2)L_{(6k-2)} i_{(3o-2)} i_{(3z-2)} \sin[(6k+3o+3z-6)\theta + \delta_{(3o-2)} + \delta_{(3z-2)}] - \\ (6k-4)L_{(6k-4)} i_{(3o-2)} i_{(3z-1)} \sin[(6k-3o+3z-3)\theta - \delta_{(3o-2)} + \delta_{(3z-1)}] - \\ (6k-4)L_{(6k-4)} i_{(3o-1)} i_{(3z-2)} \sin[(6k+3o-3z-3)\theta + \delta_{(3o-1)} - \delta_{(3z-2)}] - \\ (6k-2)L_{(6k-2)} i_{(3o-1)} i_{(3z-2)} \sin[(6k-3o+3z-3)\theta - \delta_{(3o-1)} + \delta_{(3z-2)}] - \\ (6k-2)L_{(6k-2)} i_{(3o-2)} i_{(3z-1)} \sin[(6k+3o-3z-3)\theta + \delta_{(3o-2)} - \delta_{(3z-1)}] + \\ (6k-4)L_{(6k-4)} i_{(3o-1)} i_{(3z-1)} \sin[(6k+3o+3z-6)\theta + \delta_{(3o-1)} + \delta_{(3z-1)}] + \\ (6k-2)L_{(6k-2)} i_{(3o-1)} i_{(3z-1)} \sin[(6k-3o-3z)\theta - \delta_{(3o-1)} - \delta_{(3z-1)}] \end{bmatrix} \quad (1.14)$$

La première conclusion que nous pouvons tirer de ces équations est que le couple n'aura que des harmoniques multiples de 3. Une autre propriété de ces équations est que si le courant n'a que des harmoniques impairs, alors le couple généré a uniquement des harmoniques multiples de 6.

1.3.3.3 Influence de l'onduleur de tension et de la modulation de largeur d'impulsions (MLI).

a) Influence du temps de garde

Nous pouvons facilement calculer les instants de commutation. Si nous supposons que la référence de tension β_i est normalisée dans l'intervalle $[-1,1]$ pour la tension minimale et maximale respectivement, les formules suivantes donnent les instants de commutation dans les phases ascendante ou descendante de la porteuse, en tenant compte du temps de garde, et avec l'hypothèse que le courant de phase ne change pas de polarité pendant le temps d'application du temps de garde:

$$t_{ci} = \begin{cases} \frac{T_{MLI}}{4}(1 + \beta_i) & ; I_i \geq 0 \\ \frac{T_{MLI}}{4}(1 + \beta_i) + t_g & ; I_i < 0 \end{cases} \quad \text{dans la phase ascendante} \quad (1.15)$$

$$t_{ci} = \begin{cases} \frac{T_{MLI}}{4}(1 - \beta_i) & ; I_i \leq 0 \\ \frac{T_{MLI}}{4}(1 - \beta_i) + t_g & ; I_i > 0 \end{cases} \quad \text{dans la phase descendante}$$

La tension moyenne en sortie i de l'onduleur par rapport au neutre de l'onduleur sur une demi-période de MLI s'écrit alors:

$$\langle V_{in0} \rangle = \frac{U_{bus}}{2} \beta_i - U_{tg} \text{sign}(I_i) \quad \text{avec} \quad U_{tg} = \frac{2t_g}{T_{MLI}} U_{bus} \quad (1.16)$$

La validité de l'hypothèse peut être mise en cause. Cependant la durée d'application du temps de garde est très faible par rapport à la période de MLI. Par conséquent, la probabilité d'avoir un changement de signe du courant pendant l'application du temps de garde est très faible et l'erreur moyenne commise est vraiment négligeable.

Bien évidemment, pour de faibles amplitudes du courant, la probabilité augmente considérablement. Les changements de signe deviennent très fréquents et l'erreur commise dans ce cas est plus importante.

La figure 1.19 nous montre l'allure et la transformée de Fourier de la perturbation moyenne introduite par les temps de garde sur la tension de la phase i par rapport au neutre de la machine et dont l'amplitude est égale à:

$$\langle \Delta V_{in} \rangle_{max} = \frac{4}{3} \frac{2t_g}{T_{MLI}} U_{bus} \quad (1.17)$$

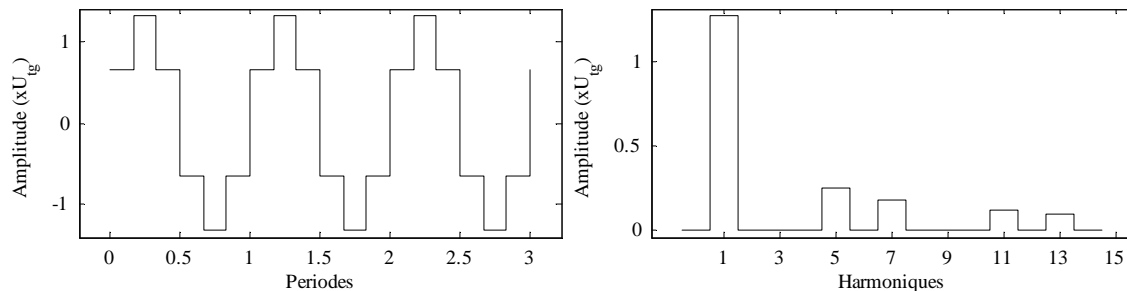


Figure 1.19: Allure de la tension perturbatrice moyenne introduite par les temps de garde de la MLI.

Cette perturbation affecte les boucles de contrôle des courants et n'est pas totalement annulée par les correcteurs de courant : elle se retrouve donc dans les courants de phase, d'autant moins atténuée que la vitesse de la machine, donc sa fréquence, est élevée. Les courants comporteront donc des harmoniques impairs et non multiples de 3 (5, 7, 11, 13,...) (voir les figures figure 1.20 et figure 1.21). Ainsi, si nous tenons compte les conclusions obtenues dans la partie 1.3.3.2, nous pouvons dire que le temps de garde crée des harmoniques multiples de 6 dans le couple de la machine.

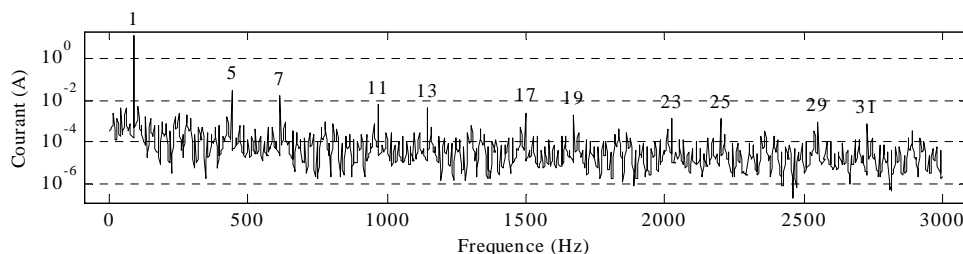


Figure 1.20: Harmoniques de courant causés par les temps de garde en boucle ouverte ($F_s=88,07\text{Hz}$, $F_{MLI}=8\text{kHz}$).

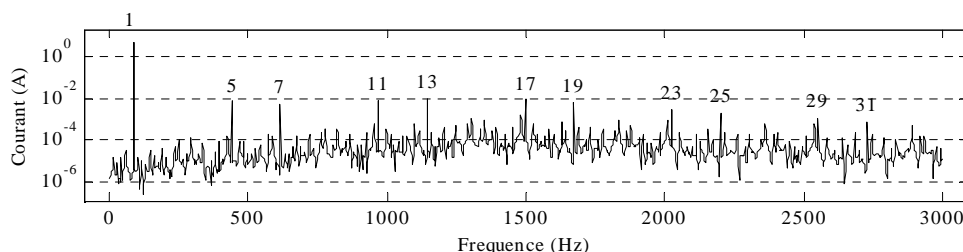


Figure 1.21: Harmoniques de courant causés par les temps de garde en boucle fermée ($F_s=88,07\text{Hz}$, $F_{MLI}=8\text{kHz}$).

D'après ces figures, nous pouvons constater qu'à basse vitesse et pour les harmoniques de rang faible, l'effet du temps de garde est réduit en boucle fermée. Par contre, au-delà de la bande passante du correcteur de courant ($\approx 1200\text{Hz}$), nous pouvons constater que les harmoniques ne sont plus atténués mais amplifiés.

b) Influence de la MLI

En boucle ouverte, et en utilisant des références de tension sinusoïdales, les harmoniques de courant générés par la MLI sont négligeables. D'une manière plus générale, la fréquence de la porteuse de la MLI n'est pas un multiple de la fréquence des signaux de référence. Nous avons donc une modulation asynchrone et, par conséquent, le courant possède des subharmoniques de la fréquence de la MLI. Dans notre cas, le ratio F_{MLI}/F_s est très grand et ces subharmoniques sont négligeables (voir figure 1.22).

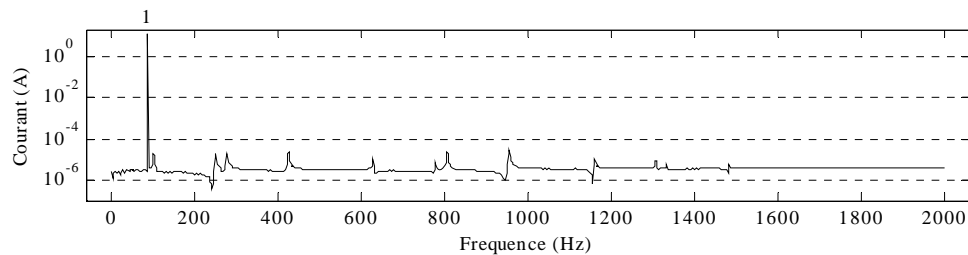


Figure 1.22: Harmoniques de courant causés par la MLI en boucle ouverte ($F_s=88,07\text{Hz}$, $F_{MLI}=8000\text{Hz}$).

En boucle fermée avec le contrôle vectoriel et en ayant des courants de référence sinusoïdaux, nous voyons apparaître des harmoniques du courant à cause de la mesure et parce que la bande passante des correcteurs n'est pas suffisamment grande pour les compenser complètement (voir figure 1.23). Néanmoins, nous observons que l'amplitude des harmoniques reste négligeable devant la composante fondamentale du courant.

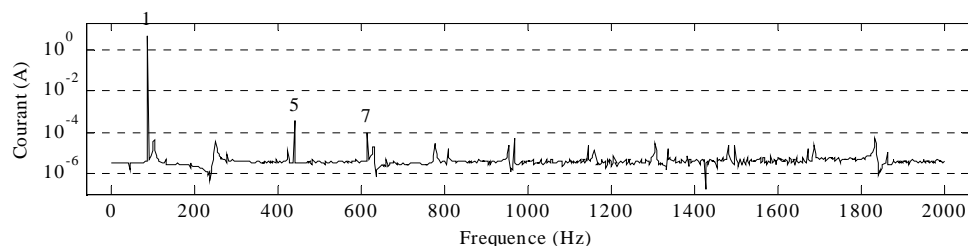


Figure 1.23: Harmoniques de courant causés par la MLI en boucle fermée ($F_s=88,07\text{Hz}$, $F_{MLI}=8000\text{Hz}$).

Nous pouvons noter que les harmoniques de courant dans le domaine de fréquence des résonances mécaniques sont très faibles.

1.3.3.4 Influence des variations de la tension de bus.

En fonctionnement moteur, la tension de bus contient des harmoniques à des fréquences multiples de 300Hz (fréquence fondamentale de la tension de sortie d'un redresseur triphasé avec une alimentation à 50Hz). En phase de freinage, la fréquence des harmoniques de la tension de bus est fixée par la charge et la décharge du condensateur de filtre. Nous considérons que les harmoniques présents sont principalement ceux du redresseur ou de la

charge/décharge du condensateur du filtre. Cette hypothèse n'est pas vérifiée en réalité parce que les harmoniques de courant ont une influence sur la tension de bus mais c'est la seule manière de pouvoir développer les équations.

Nous devons développer les équations du couple (1.13) et (1.14) pour tenir compte des courants non-sinusoïdaux avec des fréquences qui ne sont pas des multiples de la fréquence fondamentale. A partir de l'équation (1.16), nous pouvons écrire que la tension est une multiplication des harmoniques des références de tension et les harmoniques de la tension de bus. Ainsi, le courant possède les mêmes harmoniques:

$$\mathbf{i}_s = \begin{bmatrix} i_{sa} \\ i_{sb} \\ i_{sc} \end{bmatrix} = \left[1 + \sum_k A_k \cos(2\pi k f_1 t + \gamma_k) \right] \sum_{z=1} i_z \begin{bmatrix} \sin[z(\theta) + \delta_z] \\ \sin[z(\theta - 2\pi/3) + \delta_z] \\ \sin[z(\theta - 4\pi/3) + \delta_z] \end{bmatrix} \quad (1.18)$$

$$i_{3o} = 0, \quad \forall o \in N^+$$

Où A_k et γ_k sont l'amplitude et le déphasage des harmoniques générés par la tension du bus et f_1 est la fréquence fondamentale des harmoniques de la tension de bus. i_z et δ_z sont l'amplitude et le déphasage de courant sans la perturbation introduite par la tension de bus. Dans cette représentation, les harmoniques de courant sont une combinaison des harmoniques originaires du bus et les harmoniques originaires du variateur et du moteur.

Avec le courant de l'équation (1.18), les équations du couple (1.13) et (1.14) sont modifiées comme suit:

$$\Gamma'_{flux} = \left[1 + \sum_k A_k \cos(2\pi k f_1 t + \gamma_k) \right] \Gamma_{flux} \quad (1.19)$$

$$\Gamma'_{réductant} = \left[1 + \sum_k A_k \cos(2\pi k f_1 t + \gamma_k) \right]^2 \Gamma_{réductant} \quad (1.20)$$

Nous pouvons constater que le couple aura les mêmes harmoniques que précédemment, mais avec des harmoniques supplémentaires provenant de la fréquence de bus.

L'obtention des paramètres A_k et γ_k de manière théorique est quasiment impossible parce qu'en pratique, ils dépendent de la puissance fournie par l'onduleur et du filtrage réalisé par le contrôle des courants et la machine synchrone. Pour compliquer encore ce développement, en freinage, la fréquence de variation de la tension de bus dépend aussi de la puissance. Néanmoins, avec la mesure de courant, nous pouvons calculer ses harmoniques et, par conséquent, les harmoniques de couple.

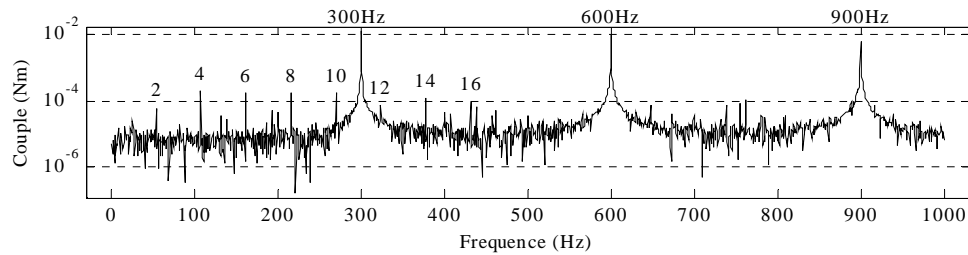


Figure 1.24: Harmoniques de couple causés par la tension de bus ($F_s=26,96\text{Hz}$, $F_{UBUS}=300\text{Hz}$).

La figure 1.24 montre les harmoniques de couple quand la tension de bus a une variation de 32,1Volts pic à pic. Nous observons l'apparition de composantes harmoniques à des fréquences multiples de la tension de bus.

A partir du schéma de la figure 1.14, nous pouvons constater que l'amplitude des harmoniques de la tension de bus dépend de la puissance demandée par la machine. Par conséquent, l'effet des variations de la tension de bus sera beaucoup plus important à forte charge et à haute vitesse.

1.3.3.5 Commande vectorielle.

La commande vectorielle n'introduit théoriquement pas d'éléments perturbateurs dans le couple de la machine. Qui plus est, elle est même capable de compenser les perturbations créées par les autres imperfections [LEO-1985].

L'implantation numérique, des algorithmes de la commande vectorielle introduit des erreurs, mais elles sont au moins d'un ordre de grandeur plus faible que les erreurs introduites par les mesures des signaux analogiques et nous pouvons les négliger sans aucun problème.

Cependant, comme nous verrons dans la partie dédiée à l'influence de la mesure, la commande sera limitée par la qualité des signaux qu'elle reçoit.

1.3.3.6 Influence des mesures.

La figure 1.25 présente les modèles schématiques des capteurs :

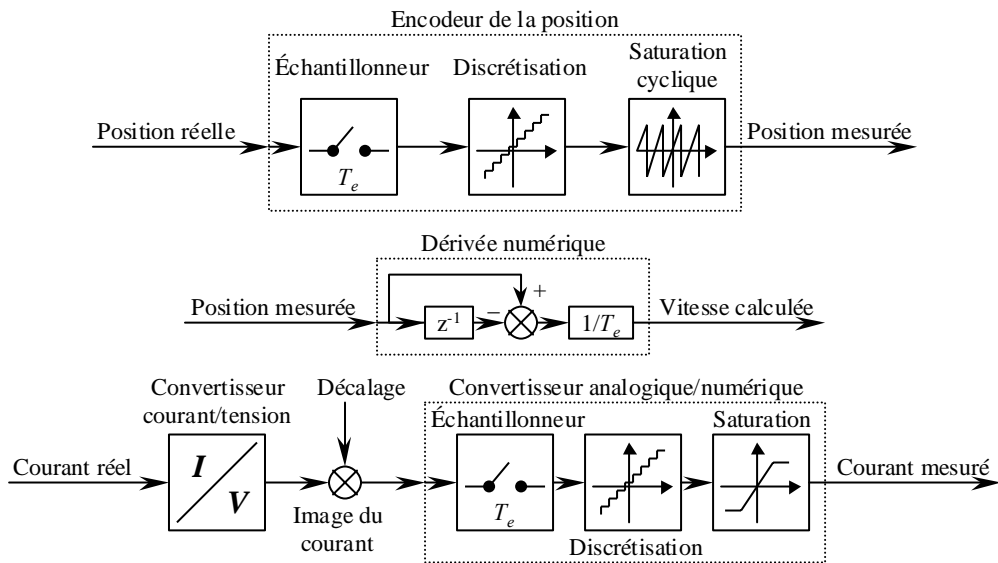


Figure 1.25: Modèles des capteurs de mesure du courant, de la position et la vitesse.

Il y a plusieurs sources d'erreurs dans la mesure. Nous allons considérer les plus importantes: le non-synchronisme de la mesure des courants avec la porteuse de la MLI, les erreurs de décalage et de gain des capteurs de courant et la discrétisation des signaux continus.

a) Non-synchronisme de la mesure des courants.

Le synchronisme de la mesure des courants joue un rôle très important dans la qualité de la mesure. En regardant la figure 1.26, nous constatons que le courant a 6 changements de pente qui correspondent aux 6 changements de configuration de l'onduleur dans une période de MLI. Cependant, si nous réalisons la mesure des courants aux instants où la porteuse est à sa valeur maximale ou minimale, nous nous affranchissons de leurs oscillations et nous obtenons approximativement leur valeur moyenne.

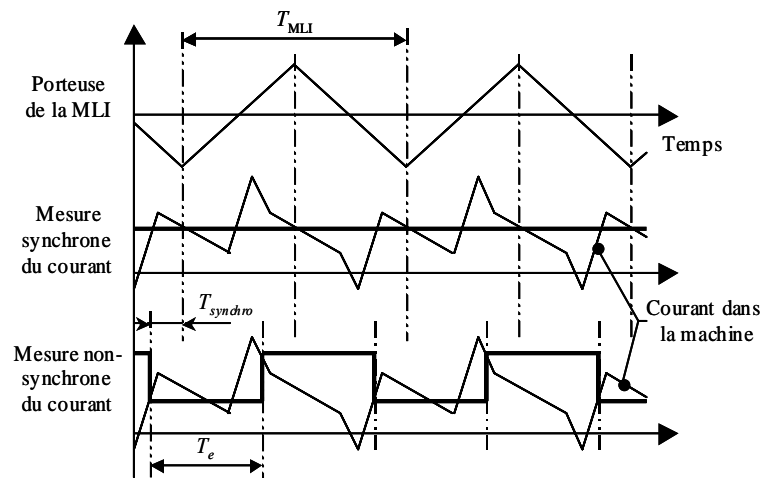


Figure 1.26: Effet du synchronisme dans la mesure des courants.

A contrario, si nous avançons la mesure des courants dans le but de donner aux convertisseurs analogiques/numériques suffisamment de temps pour réaliser la conversion, nous pouvons observer une augmentation de la puissance des harmoniques du courant. Suite à l'apparition d'une oscillation de période $2 \times T_e$ dans le courant mesuré, les harmoniques de courant augmentent considérablement à basse fréquence à cause du repliement des fréquences (voir figure 1.27).

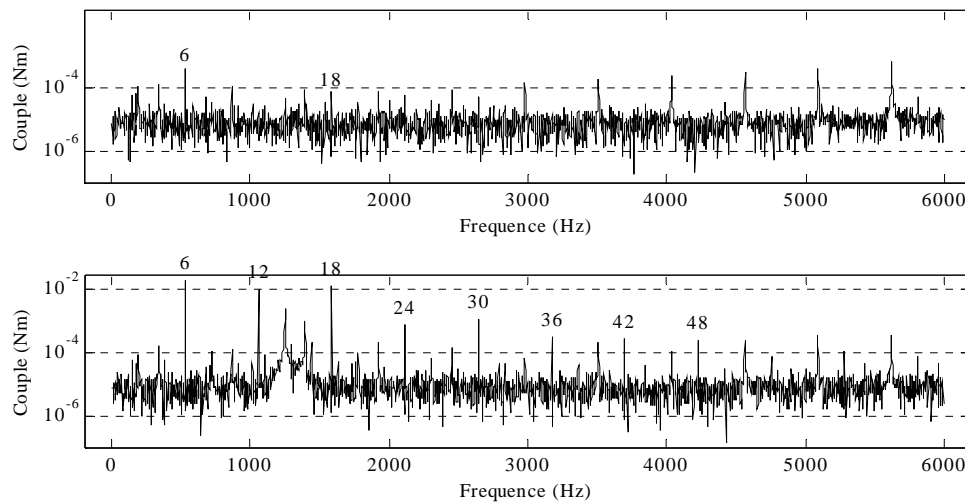


Figure 1.27: Harmoniques du couple en mesure synchrone et non synchrone à vitesse constant ($I_1=5A$, $F_s=88,07Hz$).

En pratique, il y a aussi une augmentation des pics et erreurs de mesure du courant dues aux commutations des interrupteurs. Les transitoires de très haute fréquence provoqués par les commutations des interrupteurs génèrent des bruits dans le système de mesure.

Du fait de l'augmentation des harmoniques de courant, les harmoniques de couple augmentent également et nous pouvons constater que le non synchronisme des mesures peut, selon les cas, augmenter les harmoniques de couple d'un ou deux ordres de grandeur.

b) Décalages des courants.

Les décalages des capteurs de courants sont une autre source d'harmoniques. Ils ne peuvent pas être facilement compensés car ils varient avec la température de fonctionnement.

Supposons que nous ayons une mesure des courants avec des décalages comme suit:

$$\begin{aligned} i_{am} &= i_a + \Delta i_a \\ i_{bm} &= i_b + \Delta i_b \end{aligned} \quad (1.21)$$

A basse vitesse, le correcteur de courant arrive à contrôler les courants i_{am} et i_{bm} pour qu'ils suivent les références sinusoïdales et par conséquent les courants i_a , i_b et i_c sont sinusoïdaux mais ils ont une composante continue. Si nous observons la représentation vectorielle des

courants mesurés et des courants de la machine (figure 1.28), nous pouvons observer que les deux courants tournent mais les courants de la machine sont décalés par rapport au centre du référentiel.

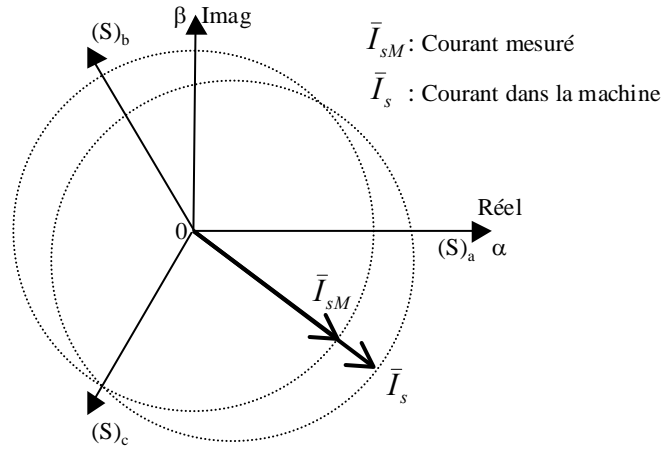


Figure 1.28: Représentation vectorielle des courants de la machine et les courants de la commande en présence d'un décalage sur le courant i_a .

Si nous reprenons les équations (1.13) et (1.14) du couple de la machine, nous obtenons les expressions suivantes avec les courants décalés:

$$\Gamma''_{flux} = \Gamma_{flux} - p \sum_{k=1} \left[\begin{array}{l} (6k-5)k_{(6k-5)}\sqrt{3}[\Delta i_a \sin[(6k-5)\theta - \pi/6] - \Delta i_b \cos[(6k-5)\theta]] + \\ (6k-1)k_{(6k-1)}\sqrt{3}[\Delta i_a \sin[(6k-1)\theta + \pi/6] + \Delta i_b \cos[(6k-1)\theta]] \end{array} \right] \quad (1.22)$$

$$\Gamma''_{reluctant} = \Gamma_{reluctant} + p \frac{3\sqrt{3}}{2} \sum_{z=1} \sum_{k=1} \left[\begin{array}{l} \Delta i_a \left[\begin{array}{l} + (6k-2)L_{(6k-2)}i_{(3z-2)} \cos[(6k+3z-4)\theta + \delta_{(3z-2)} + \pi/6] \\ + (6k-4)L_{(6k-4)}i_{(3z-1)} \cos[(6k+3z-5)\theta + \delta_{(3z-1)} - \pi/6] \\ - (6k-4)L_{(6k-4)}i_{(3z-2)} \cos[(6k-3z-2)\theta + \delta_{(3z-2)} - \pi/6] \\ - (6k-2)L_{(6k-2)}i_{(3z-1)} \cos[(6k-3z-1)\theta + \delta_{(3z-1)} + \pi/6] \end{array} \right] + \\ \Delta i_b \left[\begin{array}{l} - (6k-2)L_{(6k-2)}i_{(3z-2)} \sin[(6k+3z-4)\theta + \delta_{(3z-2)}] \\ + (6k-4)L_{(6k-4)}i_{(3z-1)} \sin[(6k+3z-5)\theta + \delta_{(3z-1)}] \\ - (6k-4)L_{(6k-4)}i_{(3z-2)} \sin[(6k-3z-2)\theta + \delta_{(3z-2)}] \\ + (6k-2)L_{(6k-2)}i_{(3z-1)} \sin[(6k-3z-1)\theta + \delta_{(3z-1)}] \end{array} \right] \end{array} \right] \quad (1.23)$$

$$+ p \sum_{k=1} \left[\begin{array}{l} \Delta i_a^2 [(6k-4)L_{(6k-4)} \sin[(6k-4)\theta - 2\pi/3] + (6k-2)L_{(6k-2)} \sin[(6k-2)\theta + 2\pi/3]] - \\ 2\Delta i_a \Delta i_b \left[\begin{array}{l} (6k-4)L_{(6k-4)} \sin[(6k-4)\theta + 2\pi/3] + \\ (6k-2)L_{(6k-2)} \sin[(6k-2)\theta - 2\pi/3] \end{array} \right] + \\ \Delta i_b^2 [(6k-4)L_{(6k-4)} \sin[(6k-4)\theta] + (6k-2)L_{(6k-2)} \sin[(6k-2)\theta]] \end{array} \right]$$

A partir de ces équations, nous constatons qu'un décalage rajoute aux harmoniques originaux des harmoniques à tous les multiples de la fréquence fondamentale sauf les multiples de 3. La figure 1.29 montre les harmoniques de couple avec un décalage de $\Delta i_a=0,1A$ et sans harmoniques de FEM ni d'inductance dans la machine.

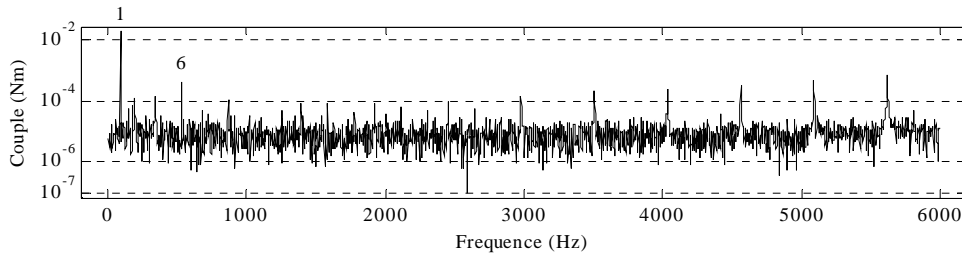


Figure 1.29: Harmoniques du couple avec un décalage de $\Delta i_a = 0,1A$ ($I_1 = 5A$, $F_s = 88,07Hz$).

c) Différences de gain des capteurs de courant.

Les différences de gain des capteurs de courants influent aussi sur les oscillations de couple. Cependant, contrairement aux décalages, ces différences de gain peuvent être assez bien compensées après un calibrage des capteurs.

Supposons que nous ayons une mesure des courants avec des gains comme suit:

$$\begin{aligned} i_{am} &= G_a i_a & \text{ou} & & i_a &= G_a^{-1} i_{am} = G_1 i_{am} \\ i_{bm} &= G_b i_b & & & i_b &= G_b^{-1} i_{am} = G_2 i_{bm} \end{aligned} \quad (1.24)$$

Nous pouvons conclure qu'à basse vitesse, le correcteur de courant arrive à contrôler les courants i_{am} et i_{bm} pour qu'ils suivent les références. Par conséquent, les courants i_a et i_b sont sinusoïdaux mais d'amplitudes différentes. i_c est aussi sinusoïdal mais son amplitude et sa phase ont changé pour que la somme des trois courants soit nulle. Si nous observons la représentation vectorielle des courants mesurés et des courants de la machine, nous pouvons observer que les deux courants sont centrés mais qu'ils ont une trajectoire elliptique (voir figure 1.30 avec $G_1 = 0,8$ et $G_2 = 1,1$).

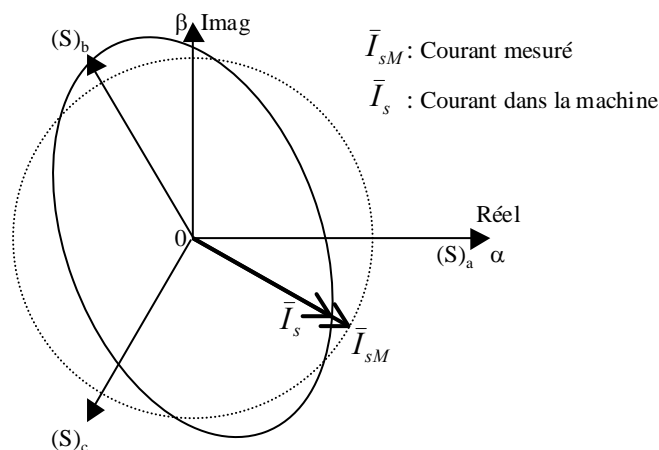


Figure 1.30: Représentation vectorielle des courants de la machine et de la commande en présence des différences de gain.

Si nous reprenons l'équation (1.13) et (1.14) du couple de la machine, nous obtenons les expressions suivantes avec les différences de gain:

$$\Gamma_{flux} = p \frac{\sqrt{3}}{2} \sum_{p=1} \sum_{z=1} i_z \begin{bmatrix} + (6p-5)k_{(6p-5)}A(-1,-z)\cos[(6p-5+z)\theta + \delta_z + \phi(-1,-z)] \\ - (6p-5)k_{(6p-5)}A(-1,+z)\cos[(6p-5-z)\theta - \delta_z + \phi(-1,+z)] \\ + (6p-1)k_{(6p-1)}A(+1,-z)\cos[(6p-1+z)\theta + \delta_z + \phi(+1,-z)] \\ - (6p-1)k_{(6p-1)}A(+1,+z)\cos[(6p-1-z)\theta - \delta_z + \phi(+1,+z)] \end{bmatrix} \quad (1.25)$$

$$A(s, z) = \left\| G_1 e^{j\frac{\pi}{6}s} + G_2 e^{j\left(\frac{2\pi}{3}z + \frac{\pi}{2}s\right)} \right\| \quad \phi(s, z) = \text{angle} \left(G_1 e^{j\frac{\pi}{6}s} + G_2 e^{j\left(\frac{2\pi}{3}z + \frac{\pi}{2}s\right)} \right)$$

$$\Gamma_{réductant} = p \frac{3}{4} \sum_{m=1} \sum_{p=1} \sum_{z=1} i_m i_z \begin{bmatrix} + (6p-4)L_{(6p-4)}A(1,-m,-z)\sin[(6p-4+m+z)\theta + \delta_m + \delta_z + \phi(1,-m,-z)] \\ - (6p-4)L_{(6p-4)}A(1,-m,z)\sin[(6p-4+m-z)\theta + \delta_m - \delta_z + \phi(1,-m,z)] \\ - (6p-4)L_{(6p-4)}A(1,m,-z)\sin[(6p-4-m+z)\theta - \delta_m + \delta_z + \phi(1,m,-z)] \\ + (6p-4)L_{(6p-4)}A(1,m,z)\sin[(6p-4-m-z)\theta - \delta_m - \delta_z + \phi(1,m,z)] \\ + (6p-2)L_{(6p-2)}A(-1,-m,-z)\sin[(6p-2+m+z)\theta + \delta_m + \delta_z + \phi(1,-m,-z)] \\ - (6p-2)L_{(6p-2)}A(-1,-m,z)\sin[(6p-2+m-z)\theta + \delta_m - \delta_z + \phi(1,-m,z)] \\ - (6p-2)L_{(6p-2)}A(-1,m,-z)\sin[(6p-2-m+z)\theta - \delta_m + \delta_z + \phi(1,m,-z)] \\ + (6p-2)L_{(6p-2)}A(-1,m,z)\sin[(6p-2-m-z)\theta - \delta_m - \delta_z + \phi(1,m,z)] \end{bmatrix} \quad (1.26)$$

$$A(s, m, z) = \left\| G_1 e^{-j\frac{2\pi}{3}s} - G_2 e^{j\frac{2\pi}{3}z} \right\| \cdot \left\| G_1 e^{-j\frac{2\pi}{3}s} - G_2 e^{j\frac{2\pi}{3}m} \right\|$$

$$\phi(s, m, z) = \text{angle} \left(G_1 e^{-j\frac{2\pi}{3}s} - G_2 e^{j\frac{2\pi}{3}z} \right) + \text{angle} \left(G_1 e^{-j\frac{2\pi}{3}s} - G_2 e^{j\frac{2\pi}{3}m} \right) + \pi$$

A partir de ces équations, nous constatons que les différences de gain rajoutent aux harmoniques originels des harmoniques à des multiples pairs de la fréquence fondamentale. La figure 1.31 montre les harmoniques de couple avec un gain de $G_a=1,1$ et sans harmoniques de FEM ni d'inductance dans la machine. Nous pouvons constater l'apparition d'un harmonique au double de la fréquence de la fondamentale.

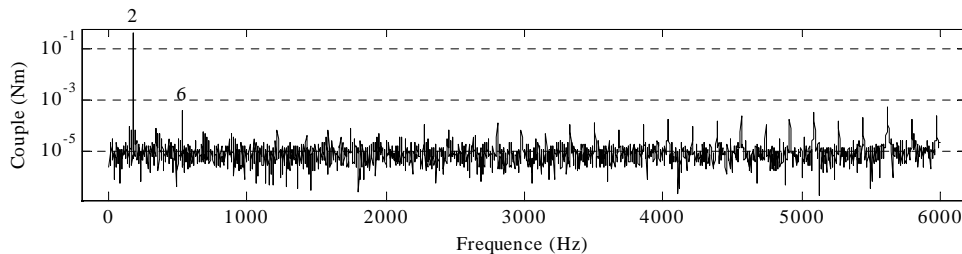


Figure 1.31: Harmoniques du couple avec un gain de $G_a=1,1$ ($I_1=5A$, $F_s=88,07Hz$).

d) Discrétisation des signaux continus.

Dans la commande vectorielle numérique, nous utilisons des convertisseurs analogiques/numériques pour obtenir les courants et la tension de bus et d'autre part, nous utilisons un encodeur pour récupérer la position. Tous ces systèmes d'acquisition de données ont une précision limitée et, par conséquent, ils introduisent des erreurs dans le système de contrôle.

L'acquisition des courants pose des problèmes si les amplitudes sont faibles. Dans ce cas, l'erreur relative devient très importante et nous ne pouvons plus contrôler efficacement les courants. Par conséquent, les harmoniques courant deviennent très importants quand la machine n'est pas chargée.

L'influence du capteur de position est négligeable car les encodeurs utilisés ont une résolution très grande. Néanmoins la dérivation de la position pour calculer la vitesse provoque une amplification considérable de l'erreur de position. Dans le contrôle des courants, la vitesse n'est utilisée que pour la compensation des perturbations. Par conséquent, nous pouvons dire que l'erreur de position ne joue pas un rôle déterminant dans les harmoniques créés dans le couple.

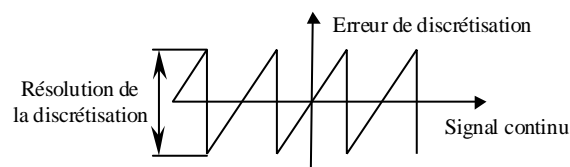


Figure 1.32: Erreur de discretisation en fonction du signal original.

La figure 1.32 présente l'allure de l'erreur de discrétisation. Il existe une corrélation entre l'erreur discrétisation et le signal mesuré. Cependant, cette corrélation varie en fonction du régime de fonctionnement de la machine. Du fait de la nature hautement non linéaire de l'erreur de discrétisation, nous ne pouvons pas retrouver une corrélation simple entre la résolution de la discrétisation et les harmoniques introduits dans le couple.

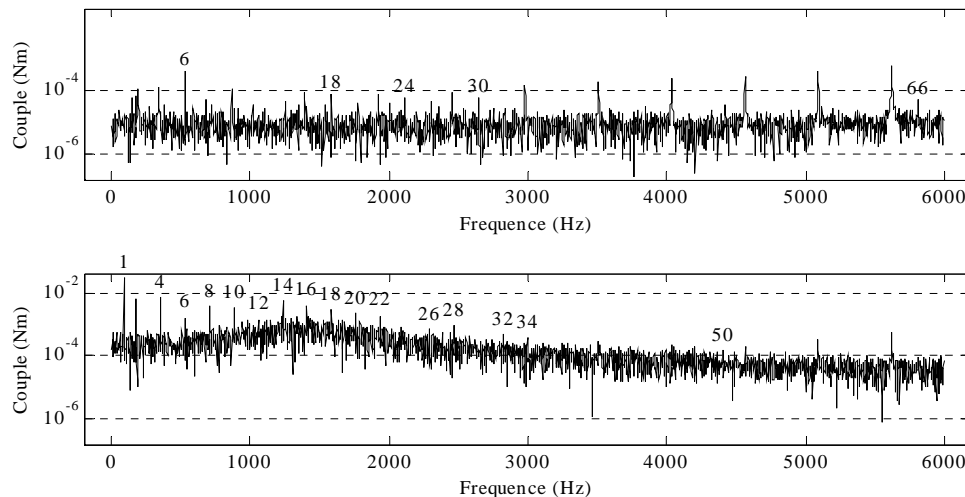


Figure 1.33: Harmoniques du couple sans discrétisation et avec discrétisation à vitesse constant ($I_1=5A$, $F_s=88,07Hz$).

La figure 1.33 montre les harmoniques de couple sans et avec discrétisation et sans harmoniques de FEM et d'inductance dans la machine. Nous pouvons observer l'apparition des harmoniques à cause des décalages introduits dans la mesure des courants et l'augmentation de la base des harmoniques de couple.

Cette augmentation peut être expliquée d'une manière qualitative si nous considérons l'erreur de discrétisation comme un bruit blanc. Dans ce cas, cette erreur est filtrée par la boucle de courant et ce que nous voyons apparaître dans la transformée de Fourier du couple est l'allure de la fonction de transfert en boucle fermée. Si nous nous reportons à l'équation (1.4), nous pouvons observer que ceci est vrai seulement quand la machine n'a pas d'harmoniques d'inductance.

1.3.3.7 Limitation de la tension à haute vitesse (surmodulation).

Ce bloc a pour but la modification des références de tension pour exploiter la tension maximale utilisable dans l'onduleur. Le problème vient du fait que les références de tension fournies à l'onduleur créent des tensions simples par rapport au neutre fictif de l'onduleur. Ainsi, la référence de tension β_{al} de la figure 1.17 crée une tension V_{an0} dans l'onduleur.

La valeur maximale de la tension V_{an0} est $U_{bus}/2$ sans tenir compte des limitations imposées par les temps de garde. Cependant la valeur maximale de la tension entre phases est limitée à la tension de bus U_{bus} et par conséquent la tension V_{an} est théoriquement limitée à $U_{bus}/\sqrt{3}$.

La solution à ce problème réside dans l'addition d'harmoniques multiples de 3 de la fréquence fondamentale aux références de tensions fournies par le correcteur. Ainsi, nous créons une tension V_{mn0} entre le neutre de la machine et le neutre fictif de l'onduleur. Même si les tensions V_{in0} restent limités à $U_{bus}/2$, les tensions de phase V_{in} peuvent atteindre la limite de $U_{bus}/\sqrt{3}$

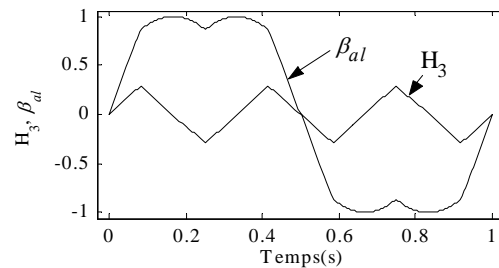
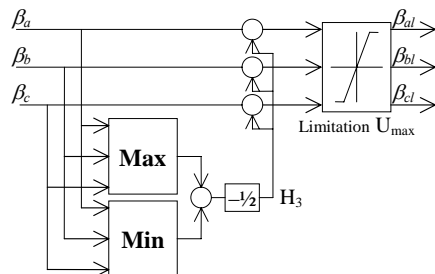


Figure 1.34: Schéma de correction des références de tension. **Figure 1.35:** Harmonique H_3 et référence de tension modifiée β_{al} pour la tension maximale de sortie.

La figure 1.34 montre le schéma de calcul des harmoniques [VEC-1995]. La figure 1.35 montre l'harmonique utilisé pour modifier les références de tension et l'allure de la référence de tension modifiée.

Ces harmoniques multiples de 3 ne créent pas d'harmoniques sur les courants de la machine car le neutre de la machine n'est pas connecté à l'extérieur.

Quand la commande arrive en limitation de tension à haute vitesse, elle n'a plus de marge pour pouvoir contrôler les courants et par conséquent, elle devient quasiment une commande à pleine onde.

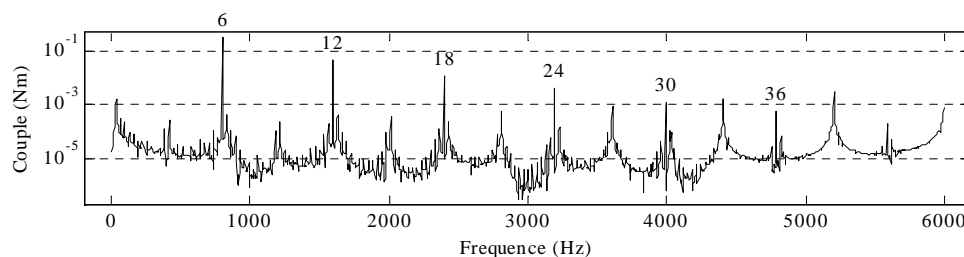


Figure 1.36: Harmoniques de couple en la limitation de tension ($F_s=133,07\text{Hz}$).

Dans ce cas et même si la machine et les capteurs sont parfaits, les courants ne sont plus du tout sinusoïdaux. Par conséquent, ces courants provoquent de très fortes harmoniques dans le couple (voir figure 1.36).

1.4 Bilan.

Les axes mécaniques, tels qu'ils sont utilisés généralement dans les modèles de simulation pour la synthèse des lois de commande, sont très simples et ne prennent pas en compte des non linéarités comme des changements de charge ou des variations de rigidité mécanique selon le point de fonctionnement. Cependant ces modèles peuvent être valables, car il existe des modèles mécaniques où les variations sont suffisamment faibles pour pouvoir les négliger. Par contre, au niveau électrique il existe des phénomènes qui introduisent des harmoniques non désirés. Le couple de crantage est un exemple de perturbation qui pourrait être compensée sans trop d'efforts. Il y a aussi d'autres contraintes matérielles au niveau pratique qui limitent les performances de la commande et créent des harmoniques supplémentaires (tels que la discrétisation, erreurs de gain, erreurs de décalage).

Des perturbations analysées, exceptés les variations de charge et l'effet de la position des vis/écrou dans la transmission, le reste des perturbations ont un effet direct sur le contrôle de couple. Nous pourrions compenser ces perturbations en calculant en parallèle avec le correcteur de vitesse les valeurs des perturbations de couple. De la sorte, nous simplifierions la tâche du correcteur du couple et nous pourrions augmenter ainsi la bande passante du correcteur du couple.

1.5 Bibliographie.

- [ALT-2000]: Y. ALTINTAS
Manufacturing Automation.
ISBN 0-521-65029-1. Cambridge University Press. 2000.
- [BAR-1995]: J.P. BARRE
Stratégies de commande pour un axe numérique de machine outil à usinage très grande vitesse.
Thèse de doctorat présentée à l'Université des Arts et Métiers. Centre de Lille. 1995.
- [LEO-1985]: W. LEONHARD
Control of Electrical Drives.
Springer Verlag. 1985..
- [OXL-1989]: P.L.B. OXLEY
Mechanics of Machining.
ISBN 0-7458-0007-6. Jon Wiley & Sons. 1989.
- [VEC-1995]: Manual del chip VECON.
Document interne FAGOR Automation. 1995.

Chapitre 2: Réseaux de neurones pour l'identification de l'axe « en temps réel ».

2.1 Introduction.

Les réseaux de neurones n'ont pas eu la diffusion attendue dans l'industrie. Ils ont été souvent confinés à des domaines informatiques et économiques. Cependant les réseaux de neurones appartiennent à l'intelligence artificielle, et comme tels, ils sont théoriquement capables de s'adapter et d'apprendre à réaliser presque n'importe quelle tâche. Notre but est d'exploiter cette possibilité pour étudier les possibilités qu'offrent les réseaux de neurones par rapport aux systèmes d'identification traditionnels.

Dans ce chapitre, nous présenterons les réseaux de neurones et ce dont ils sont capables. Nous étudierons surtout l'application des réseaux de neurones à la modélisation et à l'identification. Nous essaierons aussi d'énumérer les contraintes et les limitations des réseaux de neurones dans un contexte d'utilisation en temps réel.

Le travail est présenté de la façon suivante : nous exposons tout d'abord les généralités sur les réseaux de neurones (RN), puis nous introduisons l'apprentissage des réseaux de neurones et nous explicitons ensuite les contraintes à respecter pour leur utilisation en temps réel. Puis nous montrons comment nous pouvons utiliser les réseaux de neurones pour l'identification d'un système et les résultats de simulation obtenus. Nous finissons en faisant le bilan des conclusions obtenues dans le chapitre et en précisant le travail qui reste à réaliser.

2.2 Réseaux de neurones.

2.2.1 Le neurone.

Avant de commencer à énumérer les types de neurones, nous allons expliquer très brièvement les bases biologiques dont ils sont originaires.

2.2.1.1 Bases biologiques.

Le neurone est le processeur élémentaire du cerveau. Chaque neurone traite l'information qui lui parvient localement, puis transmet aux autres neurones qui lui sont connectés l'information qu'il a traitée. Ces cellules peuvent apprendre en changeant l'intensité de leurs connexions avec d'autres cellules ou détruire ou même créer de nouvelles connexions. Elles peuvent aussi changer leur(s) règle(s) de traitement de l'information. Ce processus de changement est appelé apprentissage et joue un rôle fondamental dans le comportement du neurone.

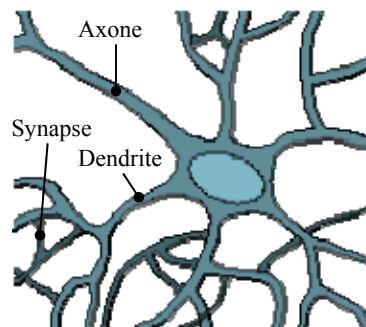


Figure 2.1: Représentation d'un neurone réel.

La figure 2.1 montre la représentation graphique d'un neurone. D'après la figure, vous pouvez constater que le neurone est composé des parties suivantes :

- **Corps cellulaire:** Noyau du neurone.
- **Dendrites:** Récepteurs principaux du neurone.
- **Axone:** Fibre nerveuse de transport pour les signaux émis par le neurone.
- **Synapse:** Connexion entre la dendrite et l'axone de deux neurones.

A partir de cette structure, nous montrerons le modèle mathématique que nous avons utilisé pour notre étude. Les références [SHE-1994] et [SEA-1994] apportent des renseignements sur les neurones, le cerveau et les aspects psychologiques.

2.2.1.2 Modélisation pratique d'un neurone.

Nous pouvons définir un neurone par les trois éléments suivants:

- La fonction d'entrée totale qui définit le pré-traitement effectué sur les entrées.
- La fonction d'activation (ou d'état) du neurone qui définit son état interne en fonction de son entrée totale.
- La fonction de sortie qui calcule la sortie du neurone en fonction de son état d'activation.

Nous adopterons les notations suivantes:

- u_i : Entrées du neurone ($i=1..n$).
- w_i : Coefficient de pondération de l'entrée i .
- $h(\bullet)$: Fonction d'entrée totale.
- $f(\bullet)$: Fonction d'activation.
- $g(\bullet)$: Fonction de sortie.
- $E=h(u_1, \dots, u_n)$: Entrée totale ou degré d'activation.
- $A=f(E)$: Etat du neurone.
- $S=g(A)$: Sortie.

La figure 2.2 montre la représentation graphique du modèle général d'un neurone.

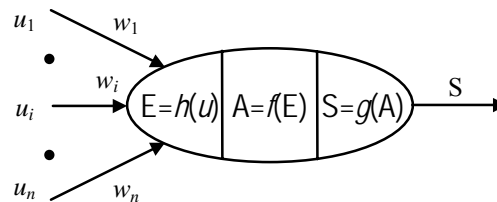


Figure 2.2: Modèle général d'un neurone.

Les entrées et les sorties peuvent être des grandeurs réelles ou binaires. La nature des différentes entrées, sorties et fonctions est explicitée ci-dessous.

La fonction d'entrée totale $h(\bullet)$ peut être :

- Une combinaison booléenne des entrées.
- Une combinaison linéaire des entrées.

$$h(u_1, \dots, u_n) = b + \sum_{i=1}^n w_i u_i \quad \text{ou} \quad h(u_1, \dots, u_n) = b + \mathbf{w}^T \mathbf{u} \quad (2.1)$$

- La distance entre les vecteurs formés par les valeurs d'entrée et les coefficients de pondération. Cette fonction est utilisée avec la fonction d'activation à base radiale. Dans

ce cas, les w_i représentent plutôt des points de l'espace d'entrée que des coefficients de pondération.

$$h(u_1, \dots, u_n) = b \sqrt{\sum_{i=1}^n (w_i - u_i)^2} \quad \text{ou} \quad h(u_1, \dots, u_n) = b \|\mathbf{w} - \mathbf{u}\| \quad (2.2)$$

Ces deux derniers cas sont les plus fréquents.

La fonction d'activation $f(\bullet)$ prend généralement les formes suivantes:

- Une fonction binaire à seuil (figure 2.3a).
- Une fonction linéaire à seuil ou seuils multiples (figure 2.3b, figure 2.3c et figure 2.3d).
- Une fonction sigmoïde (figure 2.3e).
- Une fonction à base radiale (figure 2.3f).

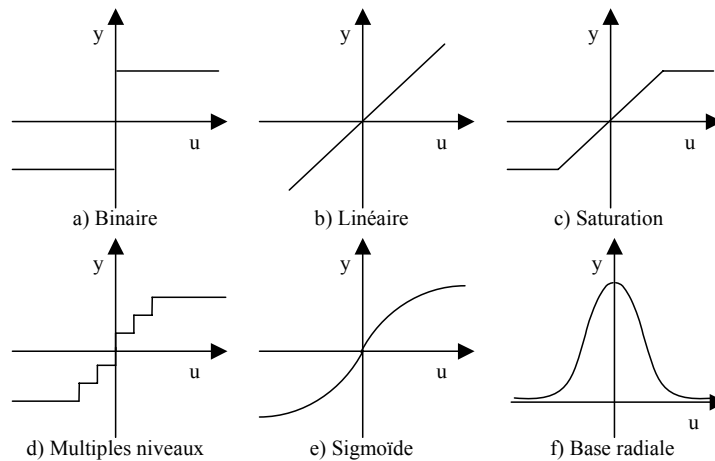


Figure 2.3: Fonctions de sortie des neurones.

- Une fonction stochastique.

$$f(x) = \begin{cases} 1 & \text{avec probabilité } (1 + e^{-kx})^{-1} \\ 0 & \text{autrement} \end{cases} \quad (2.3)$$

- Une fonction « compétitive ».

Avec cette fonction, seule la sortie du neurone dont l'entrée totale est la plus grande est activée (mise à 1) et les autres sont forcées à zéro, d'où le terme de compétition ou concurrence entre plusieurs neurones [KOH-1987].

Chacune des fonctions d'activation est adaptée à certaines applications. Par exemple, la fonction binaire ou les fonctions de type « compétitive » sont bien adaptées pour l'organisation et la distribution classification des entrées [KOH-1987] (par exemple, le maillage automatique). Les fonctions linéaires, sigmoïdes, à base radiale et plus généralement

des fonctions continues et dérivables sont appropriées pour l'approximation des fonctions continues (voir la partie sur l'apprentissage ou les références [HOR-1989][HOR-1990][ROS-1996] pour plus de détails sur cette propriété).

La fonction de sortie $g(\bullet)$ est souvent considérée comme la fonction identité. Par la suite, nous confondrons toujours activation et sortie du neurone: $S=A=f(E)$.

Ce modèle de neurone utilisé dans la littérature provient du modèle proposé par McCulloch et Pitts en 1943 [MCC-1943] et affiné postérieurement par Rosenblatt avec l'inclusion d'une méthode d'apprentissage de ce modèle [ROS-1961]. Il existe d'autres modèles de neurones plus généraux qui prennent en compte le modèle connexionniste des neurones biologiques mais ne se limitent aucunement à un type de fonction donnée (par exemple, en prenant une fonction d'entrée de type polynomiale ou autre [ROS-1996]).

2.2.2 Types de réseaux de neurones.

Les neurones trouvent leur utilité quand ils sont associés entre eux pour obtenir un réseau. Les réseaux de neurones se divisent du point de vue architecture de fonctionnement en deux classes majeures :

- Les réseaux à couches : les neurones sont groupés en couches, il n'y a pas d'interconnexion entre les neurones d'une même couche (voir figure 2.4).

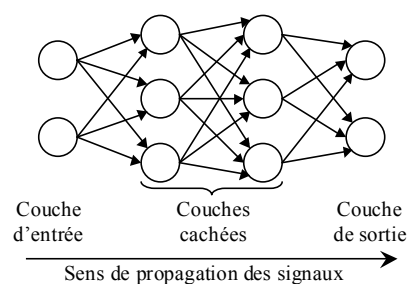


Figure 2.4: Exemple d'un réseau de neurones avec une structure en couches.

- Les réseaux récurrents : des boucles fermées de connexion et de transmission d'information entre tous les neurones sont permises (voir figure 2.5).

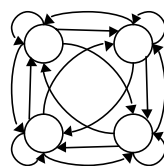


Figure 2.5: Exemple d'un réseau de neurones récurrent.

Nous pouvons aussi trouver des réseaux récurrents à couches, avec des boucles entre les couches (combinaison des deux architectures précédentes, voir figure 2.6).

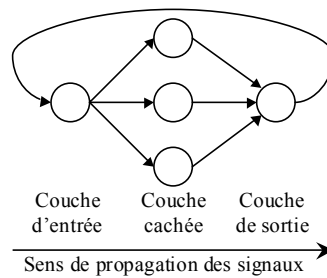


Figure 2.6: Exemple d'un réseau de neurones en couches avec une récence entre couches.

Des procédures d'apprentissage sont utilisées pour déterminer les paramètres du réseau pour qu'il se comporte correctement face à des entrées données. Selon le degré de supervision utilisé à l'apprentissage, les RN se classent en trois catégories :

- Apprentissage non supervisé : l'ajustement des paramètres repose sur des critères internes au réseau ; aucune information de référence n'est disponible.
- Apprentissage semi-supervisé ou apprentissage par renforcement : l'utilisateur ne possède que des indications imprécises sur le comportement final désiré (correct/incorrect).
- Apprentissage supervisé : les paramètres du réseau sont ajustés via un comportement de référence précis. Le réseau de neurones est entraîné de façon à reproduire le plus fidèlement possible les réponses désirées à un certain nombre de vecteurs d'entrée d'apprentissage.

L'apprentissage non supervisé est généralement utilisé pour la classification automatique des entrées, où le réseau apprend les caractéristiques des données d'entrée [KOH-1987].

L'apprentissage semi-supervisé est généralement utilisé dans les systèmes de navigation autonome, où un robot doit choisir un chemin selon un critère de type « bon/mauvais chemin ». Le travail de Kaelbling, Littman et Moore [KAE-1996] fournit une introduction plus détaillée sur l'apprentissage par renforcement.

Cependant ce sont les réseaux à apprentissage supervisé qui sont les plus utilisés. Les domaines d'application de ces réseaux peuvent être classés essentiellement en deux catégories : la reconnaissance et la généralisation.

- Dans les problèmes de reconnaissance, le réseau va apprendre à reproduire les sorties d'un ensemble de vecteurs d'apprentissage dont les entrées sont bruitées. Le réseau est censé générer les sorties adéquates bien que les entrées soient entachées d'erreur. Les applications de ces réseaux sont la reconnaissance de formes, de l'écriture, l'analyse des signaux, etc. Le réseau va reproduire une des sorties qu'il a déjà vues lors de

l'apprentissage. Pour mémoriser toutes les entrées et sorties, le réseau va construire une hyper surface dans l'espace des vecteurs d'apprentissage et passant par tous les points d'apprentissage. En général, il réclame un nombre important de neurones et, par conséquent, de paramètres.

- Pour les tâches d'approximation et d'interpolation, le réseau va apprendre à reproduire le plus précisément possible les sorties d'un ensemble de vecteurs d'apprentissage. Il doit généraliser l'information acquise à des vecteurs qui n'ont pas été présentés lors de l'apprentissage. Le fait de faire apprendre au réseau à reproduire ou mémoriser exactement les informations données par les vecteurs d'apprentissage n'est pas désiré, parce que cela fait perdre sa performance de généralisation. Cette perte de généralisation est due au phénomène de sur-apprentissage. Les applications typiques de ces réseaux apparaissent surtout dans la robotique.

Il s'ensuit qu'un réseau de neurones se distingue par trois caractéristiques principales :

- L'architecture du réseau qui caractérise un chemin particulier par lequel les éléments du réseau sont connectés et qui définit la direction de propagation des informations.
- La fonction d'activation des neurones qui dicte leur comportement.
- L'algorithme d'apprentissage du réseau.

Nous pouvons déjà constater que le nombre de types de réseaux possibles est énorme. La dénomination des réseaux est souvent spécifique selon l'architecture et les fonctions d'activations utilisées. Les réseaux des figures 3.4, 3.5 et 3.6 ne sont que des cas concrets parmi les types de réseaux existants. Une compilation des tous les types de réseau est en dehors du cadre du travail de cette thèse mais pour des références supplémentaires, vous pouvez vous adresser à la compilation de questions/réponses accessible sur le réseau Internet à la référence [SAR-1997].

2.2.3 Les réseaux de neurones appliqués à notre cas.

La nature du problème étudié impose certaines conditions sur le type de modèle que nous utiliserons : les entrées et les sorties sont réelles, continues et dérivables. Par conséquent, nous écarterons les fonctions d'activation binaires et à dérivée discontinue et utiliserons uniquement celles qui peuvent être utilisées dans l'approximation des fonctions.

Parmi les fonctions d'activation utilisées dans l'approximation, celles à base radiale constituent un cas spécial. Elles sont utilisées avec la fonction d'entrée totale de distance (équation (2.2)) et forment ce qui dans la littérature est appelé réseau de neurones à base radiale. La figure 2.7 montre un exemple de la sortie d'un tel type de réseau.

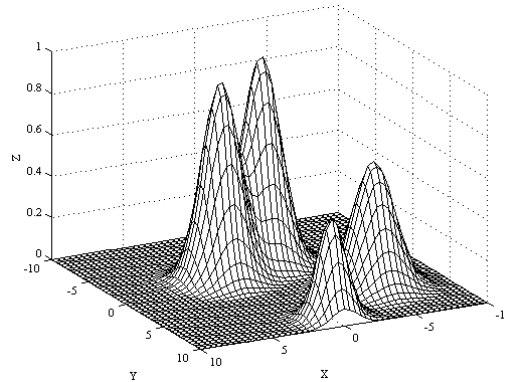


Figure 2.7: Sortie d'un réseau de neurones à base radiale avec deux entrées (X,Y), 4 neurones à base radiale et une sortie (Z).

A partir de la figure 2.7, nous pouvons observer que les réseaux de neurones à base radiale ne sont pas adaptés pour extrapoler leur sortie hors du domaine d'apprentissage à cause du support limité de la fonction d'activation à base radiale (en dehors du rayon d'action des neurones, la sortie est nulle). De même, pour approcher une surface lisse, il faudrait soit élargir le rayon d'action des neurones soit ajouter beaucoup de neurones.

La vision de réseaux de neurones à base radiale que nous donnons ici est très simpliste ; les possibilités offertes par ce type de réseau sont bien plus nombreuses. Il existe une similitude entre les réseaux de neurones à base radiale et des techniques particulières de modélisation floue [BER-1993][ROG-1993] et ces similitude permettent de donner une interprétation aux paramètres du réseau de neurones à base radiale. Il faut remarquer l'existence des applications où le choix que nous avons fait ne serait pas approprié. Vous pouvez vous reporter au livre de J.M. Renders « Algorithmes génétiques et réseaux de neurones » pour une description beaucoup plus détaillée sur les réseaux de neurones à base radiale [REN-1994].

Un des objectifs de notre travail est la mise en œuvre du réseau de neurones en temps réel pour une application de dynamique très rapide (le contrôle de couple). Ainsi, nous avons des contraintes assez importantes sur le temps de calcul. A priori, les réseaux de neurones utilisant des neurones sigmoïdes et linéaires ont moins de neurones et donc un temps de calcul plus faible. Par conséquent, nous avons choisi d'utiliser uniquement les fonctions d'activation linéaire et sigmoïde et la fonction d'entrée totale linéaire.

Avec les fonctions de sortie choisies, nous avons deux possibilités :

- Un réseau linéaire : Sans couche cachée et en utilisant la fonction de sortie linéaire dans la couche de sortie.
- Un réseau non linéaire : Une couche cachée avec la fonction de sortie sigmoïde et une couche de sortie avec la fonction de sortie linéaire.

Finalement, nous avons retenu la structure de la figure 2.6 pour le réseau car notre but est de pouvoir commander un système dynamique. La nécessité de considérer la récurrence prend son origine dans le fait qu'en automatique, la conduite de processus a besoin d'une certaine mémoire du passé et que cela ne peut se faire qu'en introduisant une certaine forme de récurrence [NAR-1990] ou des échantillons de différents instants de temps. Ce choix est justifié par le fait que, pour une récurrence externe, nous pouvons facilement l'enlever pour obtenir une structure qui prend des échantillons à différents instants de temps provenant de la mesure au la place de la sortie de notre réseau (voir la partie 2.4.2.2 pour plus de détails).

2.2.4 Modèle mathématique du réseau utilisé.

Maintenant que nous avons décidé la structure du réseau que nous allons utiliser, nous allons développer le formalisme mathématique qui sera utilisé pour représenter ce réseau.

Dans une première approche, nous allons présenter le modèle mathématique sans tenir compte de la récurrence. Ce choix peut être justifié par le fait que la récurrence peut être considérée externe au réseau de neurones et, par conséquent le modèle de base des réseaux des figures 3.4 et 3.6 est le même. Nous introduirons l'effet de la récurrence quand nous présenterons le modèle complet utilisé pour l'identification.

A partir du réseau de la figure 2.4, nous pouvons observer que les neurones d'une même couche sont complètement indépendants entre eux, c'est-à-dire que le calcul de la sortie des neurones peut se faire en parallèle en connaissant la sortie de la couche précédente. Ce parallélisme de calcul peut être exploité en utilisant des processeurs parallèles mais, dans notre cas, les calculs seront programmés sur un processeur de signal (DSP).

Une autre caractéristique, que nous pouvons observer sur cette figure est que chaque neurone d'une couche donnée (c) est relié uniquement aux neurones de la couche précédente ($c-1$, voir figure 2.8). Finalement, la dernière caractéristique du réseau utilisé est que tous les neurones d'une même couche ont la même fonction d'activation. Ce type de réseau de neurones est

appelé “réseau perceptron multicouches” dans la littérature (Provenant du terme anglais “Multilayer Perceptron”).

En reprenant l'équation du neurone avec une fonction d'entrée totale en forme de combinaison linéaire, nous pouvons obtenir les équations équivalentes pour une couche sous forme scalaire :

$$a_s^c = f^c \left(b_s^c + \sum_{i=1}^{n_{c-1}} w_{si}^c a_i^{c-1} \right) \text{ avec } s=1 \dots n_c \quad (2.4)$$

a_s^c est la sortie du neurone s de la couche c , f^c est la fonction d'activation de la couche c , b_s^c est le coefficient de décalage de la sortie du neurone s de la couche c , w_{si}^c est le coefficient de pondération du neurone s de la couche c par rapport à l'entrée i et a_i^{c-1} est l'entrée i de la couche c (ou la sortie du neurone i de la couche $c-1$). n_c et n_{c-1} sont respectivement le nombre de neurones des couches c et $c-1$.

La même équation peut s'écrire sous forme matricielle comme suit :

$$\mathbf{a}^c = f^c(\mathbf{b}^c + \mathbf{W}^c \mathbf{a}^{c-1}) \quad (2.5)$$

\mathbf{W}^c est la matrice des coefficients de pondération des entrées, \mathbf{b}^c le vecteur des coefficients de décalage de la sortie. La figure 2.8 montre graphiquement l'équation (2.4) en prenant $n_c=2$ et $n_{c-1}=3$.

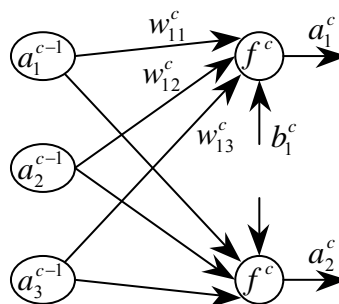


Figure 2.8: Représentation des paramètres d'une couche du réseau des neurones.

La couche dite “d'entrée” est une couche spéciale car elle ne fait que distribuer les entrées du réseau de neurones à la couche suivante (voir figure 2.4). Ainsi, pour le décompte du nombre de couches, nous ne prendrons en compte que celles qui transforment les entrées. En suivant cette convention le modèle global pour un réseau de N_c couches est :

$$\begin{aligned}
\mathbf{a}^{0)} &= \boldsymbol{\phi} \\
\mathbf{a}^{1)} &= f^{1)}(\mathbf{b}^{1)} + \mathbf{W}^{1)}\mathbf{a}^{0)} \\
&\vdots \\
\mathbf{a}^c &= f^c(\mathbf{b}^c + \mathbf{W}^c\mathbf{a}^{c-1}) \\
&\vdots \\
\hat{\mathbf{y}} = \mathbf{a}^{N_c} &= f^{N_c}(\mathbf{b}^{N_c} + \mathbf{W}^{N_c}\mathbf{a}^{N_c-1})
\end{aligned} \tag{2.6}$$

où chaque équation représente la sortie de la couche correspondante, $\boldsymbol{\phi}$ est le vecteur d'entrée du réseau de neurones et $\hat{\mathbf{y}}$ est le vecteur de sortie (nous avons représenté les valeurs numériques des couches avec une fermeture de parenthèse pour pouvoir les distinguer d'une puissance).

2.2.5 Apprentissage des réseaux de neurones pour l'approximation.

Dans la section 2.2.2, nous avons évoqué sommairement les types généraux d'apprentissage existants et nous avons décrit le choix de la structure du réseau utilisée dans ce travail dans la section suivante. Dans cette partie, nous allons approfondir plus concrètement les méthodes utilisées dans l'approximation des fonctions et la généralisation.

La propriété d'approximation universelle des réseaux de neurones a été démontrée par Hornik, Stinchcombe et White [HOR-1989][HOR-1990] sous certaines hypothèses en pratique vérifiables sur l'échantillonnage, le bruit, le nombre de neurones dans la couche cachée, la magnitude des coefficients de pondération et la forme de la fonction d'activation de la couche cachée avec un réseau de neurones entraîné avec une méthode de moindres carrées et ayant une seule couche cachée.

Un algorithme d'apprentissage pour l'approximation requiert les éléments suivants :

- Des données d'apprentissage : Utilisées pour calculer les paramètres du réseau de neurones.
- Des données de validation : Utilisées pour valider les paramètres trouvés (pas toujours utilisées).
- Un critère pour quantifier la qualité de l'approximation ou de la généralisation.
- Une méthode pour calculer les paramètres (faisant partie de l'algorithme en soi).

Nous allons représenter l'ensemble des données d'apprentissage de la manière suivante :

$$\mathbf{D}_a = \begin{bmatrix} \Phi_a \\ \mathbf{Y}_a \end{bmatrix} \text{ avec } \Phi_a = [\phi_1 \cdots \phi_j \cdots \phi_{Nja}] \text{ et } \mathbf{Y}_a = [\mathbf{y}_1 \cdots \mathbf{y}_j \cdots \mathbf{y}_{Nja}] \quad (2.7)$$

\mathbf{y}_j est le vecteur de sortie de la fonction à approcher (ou le vecteur de sortie de référence) pour un vecteur d'entrée ϕ_j . \mathbf{y}_j et ϕ_j représentent l'unité d'information nécessaire pour calculer la sortie du réseau de neurones et l'erreur d'approximation. Nous allons appeler cette unité d'information jeu de données. Nja est le nombre total de jeux de données utilisés pour le calcul des paramètres.

L'ensemble des données de validation a la même représentation que les données d'apprentissage :

$$\mathbf{D}_v = \begin{bmatrix} \Phi_v \\ \mathbf{Y}_v \end{bmatrix} \text{ avec } \Phi_v = [\phi_1 \cdots \phi_j \cdots \phi_{Njv}] \text{ et } \mathbf{Y}_v = [\mathbf{y}_1 \cdots \mathbf{y}_j \cdots \mathbf{y}_{Njv}] \quad (2.8)$$

Njv est le nombre total de jeux de données utilisés pour la validation des paramètres. Selon l'algorithme, les données de validation ne sont pas utilisées ou les données de validation peuvent être les données d'apprentissage ($\mathbf{D}_v = \mathbf{D}_a$).

La caractéristique principale des méthodes d'apprentissage pour l'approximation et la généralisation est l'utilisation d'un critère pour quantifier la qualité de l'approximation. Ce critère est appelé généralement "fonction objectif" ou "fonction de coût". Cette fonction dépend de l'algorithme d'apprentissage mais elle a souvent une forme quadratique par rapport aux erreurs d'approximation. Les équations (2.9) et (2.10) présentent deux formes typiques de la fonction objectif.

Nous allons maintenant préciser les notations que nous allons utiliser dans le reste de ce chapitre :

$\hat{\mathbf{y}}_j = f_{NN}(\mathbf{p}, \phi_j)$	Sortie prédite par le réseau de neurones pour les données d'entrée ϕ_j et un jeu de paramètres \mathbf{p} .
$\hat{\mathbf{Y}} = f_{NN}(\mathbf{p}, \Phi)$	Ensemble des sorties prédites pour un ensemble de données d'entrée Φ .
$\hat{\mathbf{Y}} = [\hat{\mathbf{y}}_1 \cdots \hat{\mathbf{y}}_j \cdots]$	
$\mathbf{e}_j = \hat{\mathbf{y}}_j - \mathbf{y}_j$	Résidu ou erreur de la sortie prédite par rapport à la valeur réelle.
$\mathbf{E} = \hat{\mathbf{Y}} - \mathbf{Y}$	Ensemble des résidus pour un ensemble de jeux de données.
$F(\mathbf{p})$	Fonction objectif donnant une quantification de la qualité de l'approximation.
$F(\mathbf{p}, \Phi, \mathbf{Y})$	Fonction objectif exprimée de manière à faire ressortir la dépendance vis-à-vis d'autres variables explicitement.
$F(\mathbf{E}, \mathbf{p})$	

Tableau 2.1: Notation et représentation des grandeurs du réseau de neurones et de l'apprentissage.

2.2.5.1 Classification des méthodes d'apprentissage.

Le tableau 2.2 résume les facteurs qui déterminent la classe d'un algorithme d'apprentissage.

Fondement théorique de l'optimisation	Fréquence de mise à jour des paramètres	Gestion des données d'apprentissage	Changement du nombre de paramètres
Déterministe	En ligne	Incrémental	Constructif
Stochastique	Différé		
Adaptatif	Hors ligne	Par lot	De réduction

Tableau 2.2: Différentes méthodes et dénominateurs des algorithmes d'apprentissage.

a) Gestion de données : Apprentissage par lot ou incrémental.

La différence entre les deux méthodes provient de l'usage qui est fait des données d'apprentissage. La méthode par lots traite toutes les données d'apprentissage et puis elle actualise les paramètres. La méthode incrémentale actualise les paramètres au fur et à mesure du traitement des jeux de données d'apprentissage.

b) Actualisation des paramètres : Apprentissage en ligne, différé et hors ligne.

Dans l'apprentissage différé, toutes les données sont enregistrées et peuvent être accédées à plusieurs reprises. L'apprentissage par lots est toujours différé.

Dans l'apprentissage en ligne, chaque jeu de données est jeté après son traitement et la mise à jour des paramètres. L'apprentissage en ligne est toujours incrémental.

L'apprentissage incrémental peut être réalisé en ligne ou en temps différé.

L'apprentissage différé permet :

- Le calcul de la fonction objectif pour les paramètres de chaque itération, avec par conséquent, la possibilité de vérifier le progrès de l'apprentissage.
- Le calcul du minimum de la fonction objectif avec une précision arbitraire en itérant plusieurs fois sur les données stockées.
- D'éviter les mauvais minima locaux en utilisant divers algorithmes, tels que des initialisations aléatoires multiples ou des algorithmes globaux d'optimisation.
- Le calcul de la fonction objectif avec les données de validation pour arrêter l'apprentissage ou choisir entre différents réseaux pour améliorer la généralisation.
- L'utilisation de méthodes spécifiques pour estimer l'erreur de généralisation.
- Le calcul des marges de confiance.

Dans l'apprentissage en ligne, nous ne pouvons plus réaliser les calculs précédents, parce que les données ne sont pas enregistrées. Par conséquent, l'apprentissage en ligne est généralement beaucoup plus difficile et incertain que l'apprentissage différé.

Certaines des difficultés théoriques de l'apprentissage en ligne sont allégées en prenant des hypothèses d'apprentissage stochastique et en supposant que l'apprentissage continue indéfiniment [SAA-1998].

L'apprentissage hors ligne suit la même démarche que l'apprentissage différé mais l'apprentissage est réalisé uniquement une fois jusqu'à l'obtention d'un réseau de neurones qui représente fidèlement les données. Par conséquent, dans l'apprentissage hors ligne, le réseau ne peut pas tenir compte des variations dans le temps.

c) Fondement théorique : Apprentissage déterministe, stochastique et adaptatif.

Selon le type de modèle que nous voulons approcher, les algorithmes d'apprentissage sont classés en deux groupes :

- L'apprentissage déterministe repose sur l'optimisation d'une fonction objectif qui peut être recalculée plusieurs fois et produit toujours la même valeur pour des valeurs de paramètres donnés. La fonction objectif a souvent la forme quadratique suivante :

$$F(\mathbf{E}, \mathbf{p}) = \sum_j \mathbf{e}_j^T \mathbf{e}_j \quad (2.9)$$

où la somme est réalisée pour tous les éléments de l'ensemble des résidus \mathbf{E} .

- L'apprentissage stochastique est généralement utilisé quand les données sont affectées par un bruit. Dans ce type d'apprentissage, l'obtention des paramètres est un processus itératif où le modèle approché est raffiné pour chaque échantillon traité et la fonction objectif prend la forme suivante :

$$F(\mathbf{e}_j, \mathbf{p}) = E\{\mathbf{e}_j^T \mathbf{e}_j\} \quad (2.10)$$

où les grandeurs présentées ont la même signification qu'auparavant et $E\{x\}$ représente l'espérance de la variable aléatoire x . La fonction objectif représente ici la covariance de l'erreur du modèle. Supposant que le modèle ne change pas et que le bruit est stationnaire, d'une itération à l'autre, le taux de variation de paramètres (ou taux d'apprentissage) doit être réduit progressivement pour que les paramètres convergent vers un minimum local.

Plus nous avons traité de données, moins nous avons d'information sur les nouvelles données d'apprentissage.

Si le modèle change dans le temps, alors le taux d'apprentissage ne peut plus être réduit car le réseau doit constamment s'adapter aux changements dans le modèle. Ce cas correspond à l'apprentissage adaptatif. Dans ce cas, il faut réaliser un compromis entre la vitesse et la précision de l'adaptation.

d) Variation du nombre de paramètres : Apprentissage constructif et algorithmes de réduction (“pruning”).

L'apprentissage constructif ajoute des neurones ou des connexions dans le réseau pendant l'apprentissage. Typiquement, il commence par un réseau sans neurones cachés et qui est entraîné pendant un moment. Puis sans changer les paramètres existants, un ou plusieurs neurones sont ajoutés au réseau, l'apprentissage reprend, et ainsi de suite. Beaucoup de variations sont possibles, impliquant différents modèles des connexions et stratégies pour fixer ou modifier les paramètres [FAH-1990][KWO-1997].

L'algorithme de réduction commence avec un réseau avec beaucoup de neurones et de connexions et supprime des neurones et/ou des connexions pendant l'apprentissage. Le processus de suppression des neurones ou des connexions s'appelle “taille” symbolisant la coupure des branches d'un réseau (du terme anglais “prune”) [REE-1993].

2.2.5.2 Méthodes d'apprentissage.

Il existe trois grands groupes de méthodes d'apprentissage provenant des domaines différents.

- Le premier groupe comprend les méthodes provenant de l'optimisation classique d'un problème de moindres carrés. Il comprend des méthodes telles que la descente du gradient, les méthodes quadratiques (Gauss-Newton et Levenberg-Marquardt [MAR-1963]), les méthodes de Quasi-Newton (BFGS [BRO-1970][FLE-1970] [GOL-1970][SHA-1970]), les méthodes des gradients conjugués [CHA-1992] et d'autres variantes. Ces méthodes sont généralement utilisées hors-ligne ou en temps différé à cause du temps de calcul nécessaire. De plus, certains de ces algorithmes ont besoin d'une quantité de mémoire considérable. La rétro propagation entre aussi dans cette catégorie. Cependant plus qu'une méthode d'apprentissage, la rétro propagation est un algorithme performant du calcul du gradient des réseaux de neurones (voir Annexe B).

- Le deuxième groupe comprend les méthodes provenant des méthodes d'identification récursive ou des méthodes d'observation d'état. Ces méthodes sont des extensions au cas non linéaire des méthodes linéaires (par exemple, le filtre de Kalman Etendu [SIN-1989] et ses variantes [WAN-2000]). Ces méthodes sont généralement utilisées en ligne.
- Le troisième groupe comprend les méthodes dites “évolutionnistes” reposant sur l'évolution des espèces vivantes et qui comprend entre autres le recuit simulé [KIR-1983] et les algorithmes génétiques [SHE-1992]. Ces algorithmes sont ceux qui ont été conçus pour rechercher l'optimum global.

2.2.5.3 Limitations et propriétés de l'apprentissage du réseau de neurones.

a) Qualité des données d'apprentissage.

Les données d'apprentissage doivent représenter le plus fidèlement possible la situation que nous voulons apprendre. Les bruits ou des données partielles nuisent considérablement à l'apprentissage. Par exemple, la figure 2.9 montre les résultats obtenus dans l'apprentissage d'une fonction sinusoïdale avec des données partielles ou avec des données bruitées.

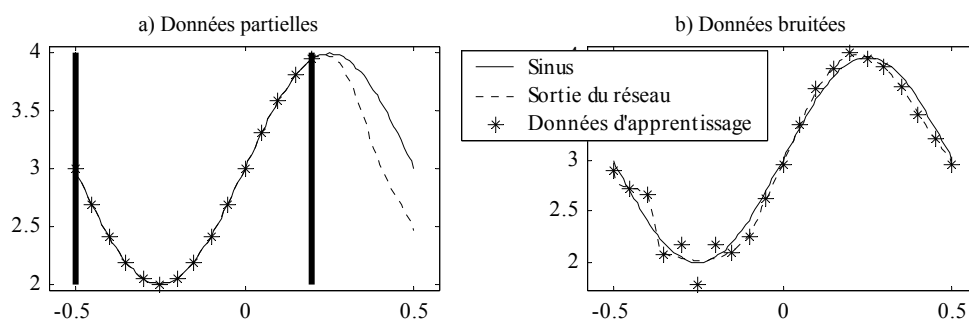


Figure 2.9: Résultats d'un apprentissage d'un sinus.

La conséquence d'avoir des données d'apprentissage de mauvaise qualité est évidente d'après les figures. Dans le cas des données partielles, le réseau interpole correctement le comportement de la fonction dans le domaine des données d'apprentissage. Cependant, il n'est pas capable d'extrapoler son comportement en dehors de ce domaine. Ceci montre l'importance d'avoir un jeu de données d'apprentissage qui comprend le comportement de la fonction à approcher dans *tout* le domaine d'utilisation du réseau de neurones.

La présence de bruit dans les données d'apprentissage est aussi mauvaise. Cependant, en modifiant l'algorithme d'apprentissage et en prenant quelques hypothèses statistiques, nous pouvons filtrer ce bruit pour extraire des données plus propres (voir la partie sur la généralisation).

b) Nombre de paramètres du réseau.

La structure et le nombre de paramètres d'un réseau de neurones est un autre facteur intervenant dans la qualité. Il n'existe pas de méthode définie pour choisir le nombre de paramètres du réseau de neurones. Plusieurs auteurs recommandent un nombre différent de neurones dans la couche cachée [BER-1990][CAR-1993][SWI-1991]. Cependant, en vérité, le choix du nombre de paramètres reste une méthode d'essais et erreurs plus qu'autre chose et il dépend du type d'application. Le choix d'un nombre trop élevé de paramètres risque d'entraîner le sur-apprentissage (voir figure 2.10). Inversement, le choix d'un nombre trop faible de paramètres ne permet pas de capter le comportement de la fonction que nous voulons faire apprendre au réseau. Il existe une discussion très intéressante sur ce sujet dans le recueil de questions/réponses réalisé par Sarle [SAR-1997].

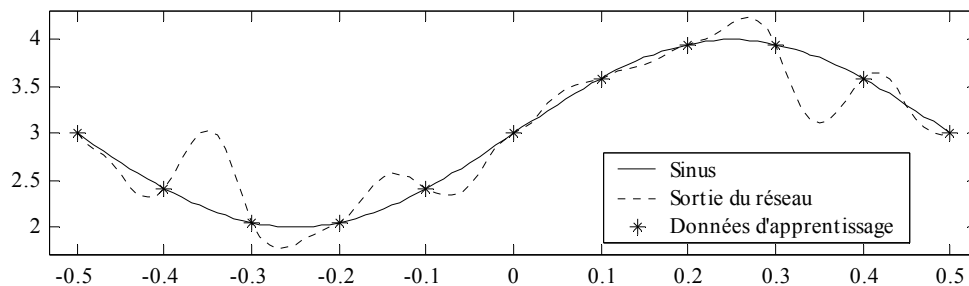


Figure 2.10: Sur-apprentissage dû au nombre trop élevé de paramètres.

c) Types de fonctions et choix des fonctions de sortie.

Il existe des fonctions qu'un réseau de neurones ne peut pas apprendre. Par exemple, la fonction $Y=1/X$ dans l'intervalle ouvert $(0,1)$. Avec n'importe quelle fonction d'activation bornée, l'erreur devient arbitrairement grande quand l'entrée tend vers zéro. Bien évidemment, nous pourrions choisir la fonction inverse comme fonction d'activation et obtenir facilement une approximation parfaite. Cependant, les réseaux de neurones sont le plus souvent utilisés dans le cas où nous ne disposons pas suffisamment de connaissances a priori pour choisir la fonction d'activation la plus appropriée.

d) Minima locaux.

Les méthodes d'apprentissage utilisant les informations locales (valeur de la fonction et le gradient en un point donné) présentent le risque de se caler sur des minima locaux de la surface décrite par la fonction de coût. Les paramètres que donne alors l'algorithme ne sont pas les valeurs recherchées.

Une solution à ce problème peut être obtenue en utilisant la régularisation de la fonction de coût qui lisse la surface obtenue et nous avons moins de risques de tomber sur un minimum local (voir la partie sur la généralisation ci-dessous).

e) Paralysie de l'apprentissage.

La paralysie du réseau est un autre problème qui peut survenir lors de l'apprentissage en utilisant le gradient.

Durant l'apprentissage, les paramètres peuvent devenir très grands, conduisant à des valeurs d'activation proches de la valeur limite de la fonction. Par exemple, la fonction d'activation sigmoïde de la figure 2.3 tend vers les valeurs -1 et 1 pour les valeurs d'entrée de magnitude très grande. Par la suite, les dérivées de cette fonction sont quasiment nulles pour ces entrées.

Les nouvelles valeurs des paramètres sont souvent calculées en utilisant le gradient. Cependant, certains éléments de celui-ci tendent vers zéro parce que les dérivées de la fonction de sortie aussi tendent vers zéro. Par conséquent, certains paramètres du réseau ne changent presque plus et on dit que les paramètres sont paralysés.

Il existe des méthodes qui évitent ce problème en utilisant le signe du gradient pour l'apprentissage au lieu d'utiliser sa valeur (l'algorithme RPROP [RIE-1993]). Dans ces algorithmes, la variation des paramètres est fixée indépendamment du gradient.

f) Généralisation.

La généralisation est une propriété recherchée dans les réseaux de neurones et qui leur permet d'extraire le comportement de la fonction à apprendre dans les cas où les données d'apprentissage seraient incomplètes ou bruitées (interpolation/extrapolation). Avec les figures 2.9a et 2.9b, nous avons montré qu'il convenait d'avoir des données les plus représentatives possibles pour améliorer la qualité du réseau retrouvé. Cependant, en utilisant des techniques provenant de la statistique, nous pouvons palier dans une certaine mesure ces défauts :

- Usage d'un filtrage préalable :

Le traitement des données avant l'apprentissage réduit le bruit.

- Régularisation de l'algorithme d'apprentissage :

Il s'agit d'un facteur d'oubli sur les paramètres qui a l'effet de filtrer les paramètres de manière à garder uniquement les caractéristiques stationnaires des données

d'apprentissage et à éviter le sur-apprentissage. A partir de l'équation (2.9), la fonction objectif est exprimée comme suit pour tenir compte de la régularisation :

$$F(\mathbf{E}, \mathbf{p}) = \alpha \cdot \sum_j \mathbf{e}_j^T \mathbf{e}_j + \beta \cdot \mathbf{p}^T \mathbf{p} \quad (2.11)$$

où α et β sont des coefficients de régularisation (souvent $\alpha=1$ et β est variable). Il est possible de coupler cette régularisation avec des méthodes Bayésiennes pour améliorer encore les propriétés de généralisation [FOR-1997].

- Utilisation des données de validation dans le critère d'arrêt :
Ceci repose sur une méthode d'arrêt prématuré en vérifiant que la fonction de coût n'augmente pas quand nous utilisons un jeu de données exclusivement utilisé pour valider les résultats obtenus avec les données d'apprentissage.

De multiples réinitialisations de l'algorithme avec des paramètres différents n'améliorent pas les propriétés de généralisation du réseau mais elles permettent de retrouver plus aisément l'optimum global et, théoriquement, le résultat recherché.

g) Initialisation du réseau de neurones.

Le mauvais choix des valeurs initiales des paramètres du réseau peut avoir deux effets. Si les paramètres initiaux sont très éloignés des valeurs recherchées, on risque de trouver un minimum local et il faut beaucoup plus d'itérations pour obtenir le résultat désiré. Nous rediscuterons de ces problèmes plus en détail dans la partie 2.4.2.4 traitant l'identification.

h) Vitesse de convergence.

Nous devons noter que la vitesse de convergence est fortement influencée par les limitations et les méthodes de généralisation que nous avons présentées. Le tableau 2.3 montre certains algorithmes qui sont couramment utilisés pour l'apprentissage d'un réseau de neurones

Méthode	Itérations	Opérations	Mémoire
Taux d'Apprentissage Variable	980	2.50	$\sim 3n+4$
Resilient Propagation (RPROP)	185	0.56	$\sim 6n+2$
Gradients Conjugués : Powell-Beale	74	0.59	$\sim 6n+8$
Quasi-Newton : BFGS	44	1.02	$\sim n^2+5n+4$
Levenberg-Marquardt	6	0.46	$\sim n^2+2n+n n_e$

Tableau 2.3: Comparaison des algorithmes d'apprentissage (n : nombre de paramètres, n_e : nombre d'échantillons d'entree).

Ces résultats ont été repris en partie du manuel d'utilisateur du logiciel MATLAB sur les réseaux de neurones [DEM-1998]. Les données que nous pouvons consulter sont le nombre d'itérations, le nombre d'opérations en virgule flottante (en millions) et la mémoire

approximative requise par l'algorithme. Les résultats ont été obtenus en faisant un moyenne sur vingt séances d'apprentissage pour un réseau de neurones 1-10-1 et 41 échantillons dans le jeu de données d'apprentissage ($n=31$, $n_e=41$) jusqu'à l'obtention d'une erreur au carré moyenne inférieure à 0,01. Les algorithmes ont été ordonnés par ordre croissant selon les contraintes de mémoire.

Le nombre d'itérations n'est pas un chiffre significatif car il ne représente pas fidèlement la tâche réalisée dans chaque itération. Le nombre d'opérations est un chiffre beaucoup plus fiable. Cependant, il faut prendre ce chiffre seulement à titre indicatif car il existe des cas particuliers où un algorithme dépasse l'autre et inversement.

2.2.5.4 Méthode retenue.

Nous allons détailler ensuite la méthode retenue pour l'apprentissage. Les raisons de ce choix seront exposées dans la partie 2.3.

a) L'optimisation par moindres carrés avec la méthode de Levenberg-Marquardt.

Nous allons introduire l'optimisation dans son cas général et puis l'appliquer aux moindres carrés car, de cette manière, nous pouvons faire ressortir les propriétés les plus importantes de cette méthode. L'optimisation dans son cadre général se ramène souvent à une minimisation car toute forme d'optimisation peut être mise sous la forme d'une minimisation en faisant un changement de signe.

L'optimisation part d'un jeu de paramètres initial \mathbf{p}_0 et d'une fonction de coût $F(\mathbf{p})$ auxquels nous appliquons itérativement un algorithme jusqu'à l'obtention d'un jeu de paramètres \mathbf{p} qui optimise la fonction. En d'autres termes :

```
Pendant que  $F(\mathbf{p}_k)$  non optimum répéter,
  Calculer nouveau pas  $\Delta\mathbf{p}_k$ 
  Actualiser les paramètres  $\mathbf{p}_{k+1} = \mathbf{p}_k + \Delta\mathbf{p}_k$ 
Fin.
```

Les critères habituels pour arrêter les itérations sont les suivants :

- La valeur du pas $\|\Delta\mathbf{p}_k\|$ est plus faible qu'un seuil de tolérance donné.
- La diminution de la fonction $\Delta F(\mathbf{p}_k)$ est plus faible qu'un seuil de tolérance donné.
- Nous avons atteint le nombre maximal d'itérations.

Le calcul du nouveau pas $\Delta\mathbf{p}_k$ peut être réalisé de plusieurs manières :

- La méthode la plus simple est la descente du gradient. Le gradient donne la direction suivant laquelle l'augmentation du coût est la plus importante. Ainsi, si nous fixons une magnitude pour cette direction, le pas peut être calculé selon la formule suivante :

$$\Delta \mathbf{p}_k = -lr \cdot F'_p(\mathbf{p}_k) \quad \text{avec} \quad F'_p(\mathbf{p}_k) = \left(\frac{\partial F(\mathbf{p})}{\partial \mathbf{p}} \right) \Big|_{\mathbf{p}=\mathbf{p}_k} = \left(\frac{\partial F(\mathbf{p})}{\partial p_1} \quad \dots \quad \frac{\partial F(\mathbf{p})}{\partial p_{Np}} \right) \Big|_{\mathbf{p}=\mathbf{p}_k} \quad (2.12)$$

Le vecteur $F'_p(\mathbf{p}_k)$ représente le gradient de la fonction $F(\mathbf{p})$ par rapport à \mathbf{p} au point $\mathbf{p}=\mathbf{p}_k$ et Np représente le nombre de paramètres. Le paramètre lr définit la taille du pas et il est souvent nommé « taux d'apprentissage » ou « learning rate » dans le contexte des réseaux de neurones.

Le paramètre lr définit la vitesse de convergence. Cependant une valeur trop grande fait diverger l'optimisation. Ainsi, il faut prévoir un système d'adaptation pour assurer la convergence : par exemple, en réduisant cette valeur quand l'erreur augmente et inversement.

- Une autre méthode de calcul du pas $\Delta \mathbf{p}_k$ repose sur la méthode de Newton. Cette méthode part d'un développement en série de Taylor de deuxième degré. Donc, nous avons :

$$F(\mathbf{p}_{k+1}) \approx F(\mathbf{p}_k) + F'_p(\mathbf{p}_k) \Delta \mathbf{p}_k + \frac{1}{2} \Delta \mathbf{p}_k^T F''_{pp}(\mathbf{p}_k) \Delta \mathbf{p}_k \quad (2.13)$$

$$F''_{pp}(\mathbf{p}_k) = \left(\frac{\partial^2 F(\mathbf{p})}{\partial p_i \partial p_j} \right) \Big|_{\mathbf{p}=\mathbf{p}_k} \quad (2.14)$$

Où la matrice symétrique $F''_{pp}(\mathbf{p}_k)$ représente le Hessian de la fonction $F(\mathbf{p})$ par rapport à \mathbf{p} au point $\mathbf{p}=\mathbf{p}_k$.

A partir de l'équation (2.13), nous pouvons calculer la valeur de $\Delta \mathbf{p}_k$ optimale en cherchant la valeur qui annule la dérivée première :

$$\frac{\partial F(\mathbf{p})}{\partial \mathbf{p}} \approx F'_p(\mathbf{p}_k) + F''_{pp}(\mathbf{p}_k) \Delta \mathbf{p}_k = \mathbf{0} \quad (2.15)$$

Nous obtenons la formule suivante :

$$\Delta \mathbf{p}_k = -F''_{pp}(\mathbf{p}_k)^{-1} F'_p(\mathbf{p}_k) \quad (2.16)$$

Cependant, cette formule n'est pas utilisable dans le cas où la matrice $F''_{pp}(\mathbf{p}_k)$ serait singulière. Marquardt a modifié l'équation précédente pour obtenir la loi de Levenberg-Marquardt qui est plus robuste [MAR-1963] :

$$\Delta \mathbf{p}_k = -(\mu \mathbf{I} + F''_{pp}(\mathbf{p}_k))^{-1} F'_p(\mathbf{p}_k) \quad (2.17)$$

Le paramètre μ et la matrice unitaire \mathbf{I} assurent que la matrice inversée sera non singulière. Dans le cas où la valeur de μ est très grande par rapport à $F''_{pp}(\mathbf{p}_k)$, nous pouvons approcher l'équation précédente comme suit :

$$\Delta \mathbf{p}_k \approx -\mu^{-1} F'_p(\mathbf{p}_k) \quad (2.18)$$

Nous pouvons constater que cette dernière formule est équivalente à la descente du gradient avec un taux d'apprentissage égale à μ^{-1} . Inversement, si la valeur de μ est très faible, nous retrouvons l'équation (2.16).

Pour compléter la méthode de Levenberg-Marquardt, il suffit de fixer une stratégie d'adaptation du paramètre μ . Une stratégie simple peut être réalisée comme suit :

- Si l'erreur diminue, le paramètre μ est diminué.
- Si l'erreur augmente, le paramètre μ est augmenté et le dernier pas est rejeté.

Cette démarche permet d'éviter la divergence de l'optimisation.

L'optimisation par moindres carrés est un cas spécial de l'optimisation générale. Il est typiquement le résultat d'une approximation de données (« data fitting » en anglais). Supposons qu'un système est modélisé par une fonction vectorielle $f(\bullet)$ qui dépend du vecteur des paramètres \mathbf{p} et d'un autre vecteur de valeurs ϕ_j . Si \mathbf{y}_j est la sortie correspondante du système pour le vecteur de valeurs ϕ_j alors le résidu

$$\mathbf{e}_j(\mathbf{p}) = f(\mathbf{p}, \phi_j) - \mathbf{y}_j \quad (2.19)$$

mesure l'erreur entre la prédiction et la sortie observée du système pour le vecteur de valeurs ϕ_j . Nous pouvons calculer l'erreur totale de la fonction comme suit:

$$F(\mathbf{p}) = \sum_j \mathbf{e}_j^T(\mathbf{p}) \mathbf{e}_j(\mathbf{p}) \quad (2.20)$$

La différence par rapport au cas général est que la fonction du coût a une forme quadratique.

Du point de vue algorithmique, la différence par rapport au cas général d'optimisation est la structure de la matrice Hessienne $F''_{pp}(\mathbf{p})$. Le gradient et le Hessien de $F(\mathbf{p})$ ont la forme suivante:

$$\begin{aligned}
F'_p(\mathbf{p}) &= \sum_j \mathbf{e}'_j(\mathbf{p})^T \mathbf{e}_j(\mathbf{p}) \\
F''_{pp}(\mathbf{p}) &= \sum_j \left[\mathbf{e}'_j(\mathbf{p})^T \mathbf{e}'_j(\mathbf{p}) + \sum_{n=1}^{Nel} e_{j,n}(\mathbf{p}) \mathbf{e}''_{j,n}(\mathbf{p}) \right] \approx \sum_j \mathbf{e}'_j(\mathbf{p})^T \mathbf{e}'_j(\mathbf{p}) \\
\mathbf{e}'_j(\mathbf{p}) &= \left(\frac{\partial \mathbf{e}_j(\mathbf{p})}{\partial \mathbf{p}} \right) = \left(\frac{\partial \mathbf{e}_j(\mathbf{p})}{\partial p_1} \quad \dots \quad \frac{\partial \mathbf{e}_j(\mathbf{p})}{\partial p_{Np}} \right) ; \quad \mathbf{e}''_{j,n}(\mathbf{p}) = \left(\frac{\partial^2 e_{j,n}(\mathbf{p})}{\partial p_i \partial p_j} \right)
\end{aligned} \tag{2.21}$$

où:

Nel : le nombre d'éléments du vecteur $\mathbf{e}_j(\mathbf{p})$.

$\mathbf{e}'_j(\mathbf{p})$: le Jacobien du vecteur $\mathbf{e}_j(\mathbf{p})$ par rapport au vecteur de paramètres \mathbf{p} .

$\mathbf{e}''_{j,n}(\mathbf{p})$: la matrice Hessienne de l'élément $e_{j,n}(\mathbf{p})$.

Souvent le terme $\mathbf{e}'_j(\mathbf{p})^T \mathbf{e}'_j(\mathbf{p})$ dans la matrice Hessienne $F''_{pp}(\mathbf{p})$ est beaucoup plus important que le deuxième terme, surtout quand les résidus $e_{j,n}(\mathbf{p})$ sont négligeables. Alors, en pratique, nous pouvons calculer le gradient et le Hessien de $F(\mathbf{p})$ en calculant le Jacobien $\mathbf{e}'_j(\mathbf{p})$.

Le cadre de la figure 2.11 résume l'algorithme complet de Levenberg-Marquardt :

$k = 0 / \mu = \mu_0$

$\mathbf{e}_j = f_{NN}(\mathbf{p}_k, \phi_j) - \mathbf{y}_j / sse_k = \sum_j \mathbf{e}_j^T \mathbf{e}_j / \mathbf{J}_j = \frac{\partial f_{NN}}{\partial \mathbf{p}}(\mathbf{p}_k, \phi_j) / \mathbf{g}_k = \sum_j \mathbf{J}_j^T \mathbf{e}_j$

Tant que $(sse_k > eg)$ et $(\mathbf{g}_k^T \mathbf{g}_k > g_{min}^2)$ et $(k < Iter_{max})$ et $(\mu < \mu_{max})$

$\mathbf{H}_k = \sum_j \mathbf{J}_j^T \mathbf{J}_j$

Repeter

$\Delta \mathbf{p}_k = (\mu \mathbf{I} + \mathbf{H}_k)^{-1} \mathbf{g}_k / \mathbf{p}_{k+1} = \mathbf{p}_k + \Delta \mathbf{p}_k$

$\mathbf{e}_j = f_{NN}(\mathbf{p}_{k+1}, \phi_j) - \mathbf{y}_j / sse_{k+1} = \sum_j \mathbf{e}_j^T \mathbf{e}_j$

$\mu = \mu \cdot \mu_{increment}$

Tant que $(sse_k < sse_{k+1})$ ou $(\mu < \mu_{max})$

Si $(\mu < \mu_{max})$

$k = k + 1$

$\mathbf{J}_j = \frac{\partial f_{NN}}{\partial \mathbf{p}}(\mathbf{p}_k, \phi_j) / \mathbf{g}_k = \sum_j \mathbf{J}_j^T \mathbf{e}_j$

$\mu = \mu \cdot \mu_{decrement}$

Fin Si

Fin Tant que

$\mathbf{p}^* = \mathbf{p}_k$

Figure 2.11: Algorithme d'optimisation de Levenberg-Marquardt.

2.3 Contraintes a priori des réseaux de neurones pour l'utilisation en temps réel.

Dans cette partie, nous allons détailler les contraintes dont nous avons dû tenir compte à l'heure de choisir certains aspects du réseau de neurones utilisé.

2.3.1 Limitations de temps de calcul.

Dans une première approche, il faut prendre en compte que la programmation du calcul du réseau de neurones sur un processeur séquentiel annule l'avantage du parallélisme des réseaux.

Dans notre application, la contrainte la plus importante est la limitation du temps de calcul. Le système que nous voulons contrôler à une dynamique très rapide (de l'ordre de 100 μ s, voir annexe C). Nous allons énumérer ensuite les facteurs nous permettant de réduire le temps de calcul des réseaux de neurones

2.3.1.1 Méthodes d'apprentissage.

Dans une deuxième approche, nous devons prendre en compte la charge de calcul introduite par l'apprentissage en ligne (voir la figure 2.12). Ainsi, nous avons choisi de ne pas utiliser des méthodes d'identification récursive ou des observateurs d'état parce que ces méthodes sont réalisées en ligne.

Parmi les deux autres groupes des méthodes d'apprentissage, les méthodes évolutionnistes ont un temps d'apprentissage très élevé car elles essaient de trouver l'optimum global. Par conséquent, il ne nous reste que les méthodes provenant de l'optimisation classique. D'après le tableau 2.3, nous pouvons constater que la méthode la plus rapide est celle de Levenberg-Marquardt. Cependant, cette méthode consomme le plus de mémoire.

2.3.1.2 Système à plusieurs périodes d'échantillonnage (apprentissage différé).

La figure 2.12 représente les diagrammes de temps de l'apprentissage en ligne et de l'apprentissage différé. Nous pouvons constater que le calcul du réseau de neurones prend toujours le même temps. Cependant, dans l'apprentissage en ligne, le temps de calcul de la tâche T_a est rajouté au temps de calcul de la tâche T_c et il limite la période de calcul de la tâche de contrôle. En dissociant la tâche de calcul du réseau de neurones de l'apprentissage,

nous pouvons réduire la périodicité de la tâche de contrôle au minimum. La contrepartie de cette démarche est que la durée d'apprentissage est beaucoup plus élevée.

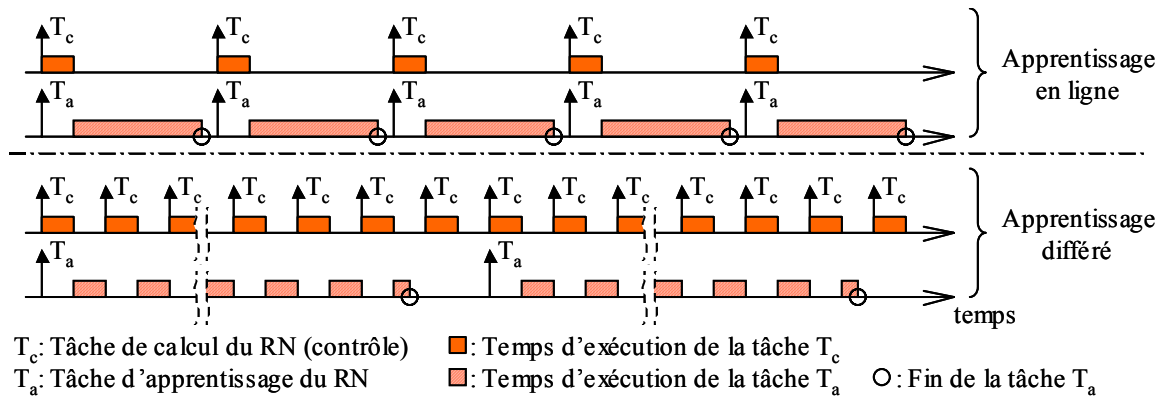


Figure 2.12: Diagramme de temps de l'apprentissage en ligne et l'apprentissage différé.

L'utilisation de l'apprentissage différé est restreinte aux applications où la dynamique ne varie pas beaucoup pendant deux étapes d'apprentissage.

2.3.2 Limitations de mémoire.

Dans notre application, nous n'avons pas eu de contraintes de mémoire très fortes car nous disposons d'une carte DSP avec une quantité de mémoire assez élevée (512ko).

Cependant dans une application embarquée où la mémoire serait un facteur limitatif, il faudrait envisager de remplacer l'algorithme de Levenberg-Marquardt par un autre comme les gradients conjugués [POW-1977] ou le "Resilient Propagation" (RPROP) [RIE-1993] lesquels utilisent moins de mémoire (voir le tableau 2.3).

2.4 Application à l'identification de l'axe.

2.4.1 Rôles du réseau de neurones.

Les rôles possibles d'un réseau de neurones sont l'identification, la réduction des oscillations de couple et la commande de l'axe.

a) Commande de l'axe.

Pour la commande de l'axe, l'intérêt des réseaux de neurones réside dans l'utilisation des propriétés d'apprentissage pour régler "automatiquement" le correcteur neuronal. La commande doit tenir compte des trois problèmes suivantes :

- La présence de la fréquence de résonance mécanique liée à la flexibilité de l'axe et ses évolutions avec les variations de la charge de l'axe ;
- les à-coups de couple au contact avec la pièce à usiner ;
- les harmoniques de couple générés par les imperfections du moteur (harmoniques de FEMs, couple de crantage), du variateur (dérive des capteurs) et de la transmission mécanique (voir chapitre 3).

b) Réduction des oscillations de couple.

Dans les harmoniques de couple, les plus gênants sont ceux qui se situent dans une fréquence relativement basse pour tomber dans la bande passante de la partie mécanique et nuire à la qualité du contrôle de positionnement de l'axe. Cependant, ils sont différents d'un entraînement à l'autre, ce qui justifie l'utilisation des capacités d'apprentissage d'un réseau de neurones pour les compenser.

Plus concrètement, le couple de crantage est un candidat pour ce type de compensation. En prenant un réseau non linéaire non récurrent à une entrée (la position de l'axe) ou deux entrées (position et vitesse de l'axe) nous pouvons calculer la correction que nous devons ajouter à la référence de couple pour compenser l'effet du couple de crantage. Avec ce réseau, nous obtenons un modèle du couple de crantage et le moyen de le corriger (voir §3.5.1).

c) Identification de l'axe.

L'obtention d'un modèle du comportement du système est très intéressante. Les usages possibles peuvent être : la synthèse de la loi de commande, l'observation et l'utilisation de

l'état estimé dans la commande, la substitution du modèle au système réel dans un processus itératif indépendant d'apprentissage de la loi de commande.

Dans ce rapport, nous allons surtout détailler cette dernière partie.

2.4.2 Identification de systèmes échantillonnés en utilisant des réseaux de neurones.

Nous allons commencer par donner les principes de l'identification et nous appliquerons ensuite ces principes aux réseaux de neurones. Nous citerons les similitudes entre les méthodes d'identification traditionnelles et les méthodes d'identification en utilisant les réseaux de neurones au fur et à mesure que nous introduirons la méthode utilisée. Finalement, nous montrerons les résultats obtenus en simulation.

2.4.2.1 Principes de l'identification de modèles dynamiques.

Les modèles de connaissance (reposant sur les lois de la physique, chimie, etc.) permettent une description assez complète des systèmes et sont utilisés pour leur simulation et leur conception. Ces modèles sont en général extrêmement complexes et rarement directement utilisables pour la commande.

Bien que des indications sur la structure des modèles de commande puissent s'obtenir à partir de la structure du modèle de connaissance, il est souvent très difficile de déterminer les valeurs des paramètres significatifs à partir de ces modèles. Par exemple, dans notre cas, nous n'avons pas de connaissance sur le modèle mécanique. C'est la raison pour laquelle nous avons été amenés à mettre en œuvre une méthodologie d'identification directe des modèles dynamiques.

Notons que les modèles dynamiques sont de deux types :

- **Modèles non paramétriques**
(par exemple : réponse fréquentielle, réponse à un échelon).
- **Modèles paramétriques**
(par exemple : fonction de transfert, équation différentielle ou équation aux différences).

Nous nous intéresserons par la suite à l'identification des modèles dynamiques paramétriques échantillonnés qui sont les plus appropriés pour la conception et l'ajustement des systèmes

numériques de commande et régulation. Plus particulièrement, le modèle paramétrique sera représenté par un réseau de neurones.

L'identification comporte quatre étapes :

1. Acquisition des entrées/sorties dans une procédure d'expérimentation.
2. Choix de la structure du "modèle" (complexité).
3. Estimation des paramètres du modèle.
4. Validation du modèle identifié (structure et valeur des paramètres).

Les méthodes spécifiques utilisées dans chaque phase dépendent du type de modèle recherché (paramétrique ou non paramétrique, continu ou échantillonné).

Le principe de l'estimation des paramètres des modèles échantillonnés est illustré à la figure 2.13.

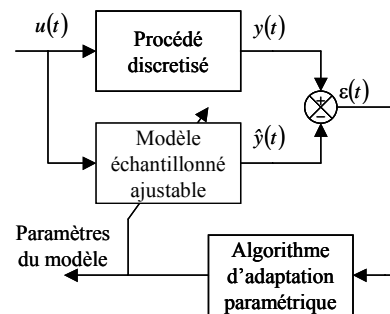


Figure 2.13: Principe de l'estimation des paramètres d'un modèle.

Un modèle échantillonné à paramètres ajustables est implanté sur le calculateur. L'erreur entre la sortie du système $y(t)$ et la sortie prédite par le modèle $\hat{y}(t)$, appelée erreur de prédiction, est utilisée sur un horizon de temps par l'algorithme d'identification qui modifie les paramètres afin de diminuer cette erreur. L'entrée $u(t)$ est souvent une séquence binaire pseudo-aléatoire de faible niveau (succession d'impulsions rectangulaires de durée aléatoire, voir §2.4.2.3). Une fois le modèle obtenu, une validation objective peut être faite par des tests statistiques sur l'erreur de prédiction $\varepsilon(t)$ et la sortie prédite $\hat{y}(t)$.

Un des éléments clés pour la mise en œuvre de cette approche pour l'identification est l'algorithme d'adaptation paramétrique qui pilote les paramètres du modèle ajustable de prédiction à partir des échantillons recueillis sur le système. Cet algorithme traite en bloc toutes les données d'entrée/sortie obtenues sur un horizon de temps.

Indiquons qu'il existe aussi des algorithmes récursifs d'identification paramétrique (qui traitent les données d'entrée/sortie à chaque pas d'échantillonnage) [LAN-1993]. Le tableau

2.4 présente la comparaison entre les algorithmes d'identification récursifs et ceux qui traitent en bloc toutes les données d'entrée/sortie.

Algorithmes de traitement en bloc	Algorithmes récursifs
<ul style="list-style-type: none"> ↳ Traitement itératif des données pour l'obtention d'une estimation plus fine des paramètres. ↳ Dissociation de l'algorithme d'identification de l'utilisation du prédicteur calculé (voir la figure 2.12). ↳ Temps de calcul total plus élevé. ↳ Nécessité d'une mémoire sensiblement plus importante. 	<ul style="list-style-type: none"> ↳ Estimation du modèle au fur et à mesure que le système évolue et poursuite des variations des paramètres. ↳ Compression importante des données. ↳ Mise en œuvre plus simple. ↳ Augmentation du temps de calcul du prédicteur (voir la figure 2.12).

Tableau 2.4: Comparaison des algorithmes récursifs d'identification et traitement en bloc.

2.4.2.2 Structure d'identification.

Dans cette partie, nous allons présenter l'adaptation aux réseaux de neurones des structures qui sont utilisées habituellement pour l'identification des systèmes.

a) Modèles paramétriques en forme d'espace d'état et de fonction de transfert.

La figure 2.14 montre la structure équivalente à un système d'équations d'état discret. La figure 2.15 montre la structure équivalente à une fonction de transfert discrète.

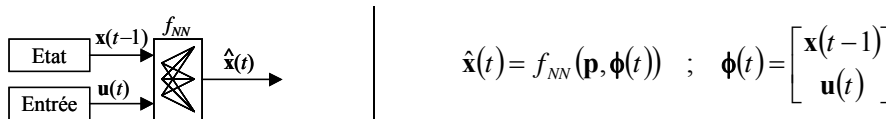


Figure 2.14: Structure équivalente à un système d'état discret.

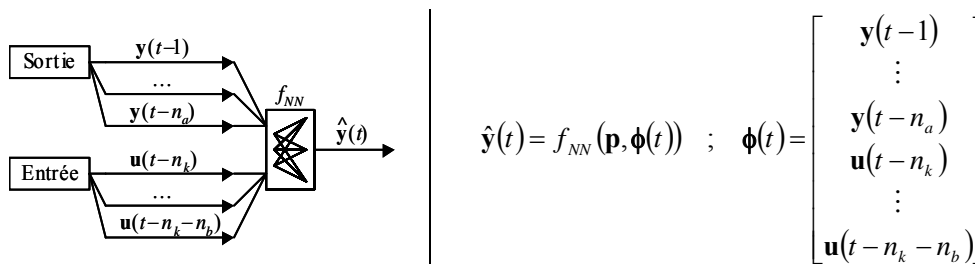


Figure 2.15: Structure équivalente à une fonction de transfert discrète.

$u(t)$, $x(t)$, $\hat{y}(t)$ représentent l'entrée du système, l'état et la sortie prédite à l'instant t . (n_b+1) et n_a représentent le nombre d'échantillons d'entrée et de sortie utilisés dans le modèle et n_k est le retard du modèle (voir l'équation (2.25)).

L'identification des structures reposant sur les équations d'état nécessite la connaissance des variables d'état du système. Ces structures ne sont donc pas appropriées pour l'identification

car la plupart des systèmes n'ont pas de capteurs pour tous les états. Par conséquent, nous utiliserons les structures fondées sur la fonction de transfert discrète. On peut montrer que, sous certaines conditions assez lâches, ce modèle reprend une large classe de systèmes non linéaires discrets [NAR-1990]. Ces conditions sont des conditions d'observabilité généralisée.

L'équation mathématique de ce modèle est :

$$\hat{y}(t) = \hat{f}(y(t-1), y(t-2), \dots, y(t-n_a), u(t-n_k), u(t-n_k-1), \dots, u(t-n_k-n_b)) \quad (2.22)$$

b) Modèle série-parallèle ou modèle parallèle.

L'identification des modèles paramétriques peut être réalisée de deux manières selon l'origine de la sortie que nous réinjectons sur le modèle :

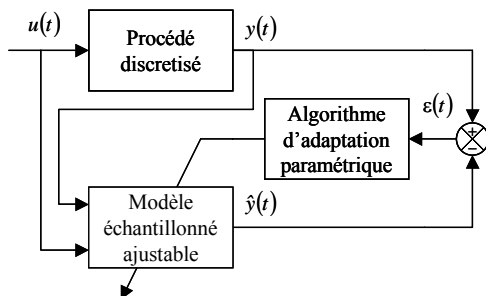


Figure 2.16: Modèle série-parallèle d'identification.

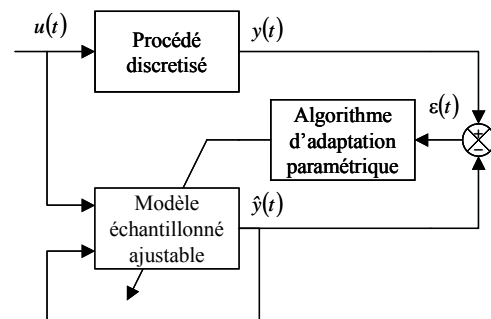


Figure 2.17: Modèle parallèle d'identification.

La figure 2.16 montre le modèle série-parallèle où nous utilisons la sortie réelle du système pour le calcul de la sortie prédite (le réseau équivalent serait non récurrent). Cette structure est préférable du point de vue de la stabilité (garantie sous certaines conditions assez générales [LJU-1987]), de la performance et de la simplicité des algorithmes. L'équation représentant ce modèle est l'équation (2.22). Cette équation est équivalente à la structure d'identification traditionnelle ARX quand la fonction $\hat{f}(\bullet)$ est linéaire (ARX : Auto-Regression with eXogenous input) [LJU-1987].

La figure 2.17 montre le modèle parallèle d'identification où nous utilisons la sortie prédite par le modèle (le réseau équivalent serait récurrent). Dans notre cas, cette structure est un cas particulier d'une structure plus générale de réseau récurrent (voir figure 2.6). L'équation représentant ce modèle est :

$$\hat{y}(t) = \hat{f}(\hat{y}(t-1), \hat{y}(t-2), \dots, \hat{y}(t-n_a), u(t-n_k), u(t-n_k-1), \dots, u(t-n_k-n_b)) \quad (2.23)$$

Cette équation est équivalente à la méthode du modèle (MM) quand la fonction $\hat{f}(\bullet)$ est linéaire (en terminologie anglaise la méthode MM est désignée par OE ou « Output Error ») [LJU-1987].

c) Effet du bruit de mesure.

Dans la réalité, les mesures sont affectées par des bruits et, par conséquent, il faut supprimer l'effet de ces bruits dans l'identification. Sinon, nous identifierons un modèle combiné du processus et du bruit.

L'identification traditionnelle linéaire tient compte de ce bruit de deux manières. Nous appliquerons ces mêmes principes aux réseaux de neurones.

- La première manière consiste en ajouter l'erreur de prédiction $\varepsilon(t)$ comme une entrée supplémentaire au modèle série-parallèle. Cette entrée représente l'erreur de mesure de la sortie du système et en l'ajoutant au modèle, nous pouvons compenser son effet dans l'estimation des paramètres. L'équation que nous obtenons a la forme suivante :

$$\hat{y}(t) = \hat{f}(y(t-1), \dots, y(t-n_a), u(t-n_k), \dots, u(t-n_k-n_b), \varepsilon(t-1), \dots, \varepsilon(t-n_\varepsilon)) \quad (2.24)$$

avec $\varepsilon(t) = y(t) - \hat{y}(t)$

Cette équation est équivalente à la structure d'identification ARMAX quand la fonction $\hat{f}(\bullet)$ est linéaire (ARMAX : Auto-Regressive Moving Average with eXogenous input [LJU-1987]).

- La deuxième manière consiste à utiliser le modèle parallèle (2.23). Dans ce cas la sortie prédite n'est pas affectée par le bruit de mesure de la sortie du système parce qu'elle est calculée de manière indépendante (parallèle). Si notre modèle a convergé vers ses valeurs théoriques, l'erreur de prédiction est égale à l'erreur de mesure. La propriété la plus remarquable est une meilleure estimation des paramètres en présence de bruit dans les signaux. En contrepartie, la stabilité de l'algorithme n'est pas assurée sans prendre de précautions, car les bruits présents dans la sortie du système peuvent rendre le modèle instable.

d) Choix du type de fonction d'approximation.

Le choix du nombre de couches et le type de fonction de sortie fixe le type de fonction $\hat{f}(\bullet)$ que nous pourrions obtenir avec le réseau de neurones. Pour les raisons décrites dans la partie 2.2.3, nous avons choisi d'utiliser uniquement les fonctions de sortie de type linéaire et sigmoïde. Par la suite, nous traiterons uniquement deux sortes de fonctions $\hat{f}(\bullet)$.

Le premier type de fonction est linéaire ; elle est obtenue avec un réseau à une seule couche et une fonction de sortie linéaire. Avec ce type de fonction, les équations (2.22), (2.23) et (2.24) deviennent des équivalentes des structures ARX, OE et ARMAX respectivement. Ce type de fonction nous permettra de valider la précision des modèles obtenus par rapport aux méthodes traditionnelles d'identification.

Le deuxième type de fonction est non linéaire. Elle est obtenue avec un réseau à deux couches et des fonctions sigmoïdes et linéaires. Avec ce type de fonction, les équations (2.22), (2.23) et (2.24) deviennent des équivalents non linéaires des structures ARX, OE et ARMAX respectivement, c'est-à-dire, NARX, NOE, et NARMAX. Ce type de fonction permet en principe d'approcher les non linéarités d'un système non linéaire.

e) Ordre du modèle (nombre de paramètres).

Dans le cas de l'identification linéaire, le nombre de paramètres du modèle peut être quantifié si nous avons des connaissances sur le système à identifier. Sinon, nous avons des méthodes qui permettent de calculer le nombre de paramètres du modèle. Pour être précis, les valeurs que nous pouvons calculer sont exactement :

- Le nombre d'échantillons d'entrée (n_b+1 dans les équations).
- Le nombre d'échantillons de la sortie prédite ou de la sortie du système (n_a dans les équations).
- Le nombre d'échantillons de retard sur l'entrée (n_k dans les équations).

Cependant dans l'identification non linéaire avec les réseaux de neurones, même si nous pouvons plus ou moins quantifier les valeurs présentées ci-dessus, il nous reste toujours à déterminer le nombre de neurones dans la couche cachée (appelé n_h). Cette valeur fixe les non linéarités possibles que nous pouvons identifier et par conséquent, son choix est critique.

2.4.2.3 Importance de la persistance de l'excitation.

Ce point est extrêmement important au moment de réaliser une identification correcte du système. En général, plus il y a des paramètres, plus le choix d'un signal d'excitation correct est contraignant.

Dans l'identification traditionnelle ce problème a été résolu en excitant le système avec des signaux spéciaux. Ces signaux sont des signaux aléatoires (Gaussiens ou uniformes) ou des

séquences binaires pseudo aléatoires (SBPA) qui sont plus simples à générer. Ces signaux présentent les caractéristiques d'un bruit blanc sur une plage de fréquence s'étendant de 0Hz jusqu'à la fréquence d'échantillonnage [LAN-1993]. Vous pouvez observer la forme de ces signaux à la figure 2.18.

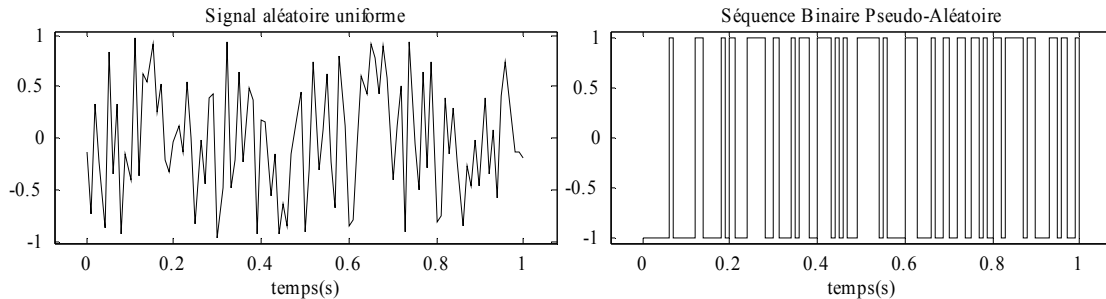


Figure 2.18: Signaux d'excitation pour l'identification.

Nous verrons que pour l'identification des systèmes non linéaires, le signal d'excitation doit aussi présenter une richesse dans les amplitudes (voir les résultats de simulation du §2.4.3.3).

2.4.2.4 Valeurs initiales des paramètres du réseau de neurones.

La vitesse de convergence de l'algorithme d'apprentissage dépend en grande partie de la proximité du vecteur de paramètres optimum et du vecteur des paramètres initial. De plus, il se peut que l'algorithme converge vers un optimum local au lieu de l'optimum global et par conséquent, trouve un jeu de paramètres qui n'est pas le meilleur possible (voir la partie 2.3.1).

Le choix des valeurs initiales dans un contexte d'identification peut être réalisé de deux manières :

1. De manière totalement aléatoire.
2. En utilisant un modèle ou une connaissance a priori sous forme d'une équation récurrente, pour initialiser les paramètres du réseau de neurones.

La première méthode est très simple à mettre en œuvre mais nous courrons le risque de spécifier un modèle initial instable et les algorithmes d'apprentissage ne sont pas capables de converger avec un réseau récurrent sous ces conditions. Par conséquent, nous devons nous assurer que le modèle initial est stable.

Une manière d'obtenir un modèle initial aléatoire stable consiste à choisir les valeurs des pôles de la fonction de transfert sous la condition de stabilité, puis de la convertir en une équation récurrente et finalement à utiliser cette équation pour obtenir les valeurs initiales du

jeu de paramètres. Cependant, la difficulté de cette approche réside dans la conversion d'une équation récurrente en un jeu de paramètres pour le réseau de neurones.

Le problème est le même pour transformer une connaissance a priori en un jeu de paramètres, c'est-à-dire, l'autre méthode pour initialiser les paramètres du réseau.

La deuxième méthode permet théoriquement de raccourcir la durée d'apprentissage en commençant avec un jeu de paramètres qui est plus proche de la valeur théorique. Ceci est vrai uniquement si la connaissance disponible n'est pas fausse.

a) Obtention des paramètres initiaux à partir d'un modèle.

Soit un modèle exprimé sous forme d'une fonction de transfert discrète que nous voulons utiliser pour initialiser les paramètres du réseau :

$$\frac{Y(z)}{U(z)} = F(z) = z^{-n_k} \frac{b_0 + b_1 z^{-1} + \dots + b_{n_b} z^{-n_b}}{1 + a_1 z^{-1} + \dots + a_{n_a} z^{-n_a}} \quad (2.25)$$

n_b ici représente l'ordre du polynôme numérateur et (n_b+1) est le nombre d'échantillons de l'entrée. n_a représente l'ordre du polynôme dénominateur et également le nombre d'échantillons de la sortie et n_k est le nombre de retards sur le signal d'entrée.

Nous pouvons transformer ce modèle en une équation récurrente :

$$y(t) = b_0 u(t - n_k) + b_1 u(t - 1 - n_k) + \dots + b_{n_b} u(t - n_b - n_k) - a_1 y(t - 1) - a_2 y(t - 2) - \dots - a_{n_a} y(t - n_a) \quad (2.26)$$

Nous utiliserons cette équation pour calculer le jeu de paramètres du réseau de neurones montré à la figure 2.15. Pour cela, la manière la plus simple est de linéariser la fonction du réseau de neurones :

$$\begin{aligned} b_0 &= \left. \frac{\partial f_{NN}(\mathbf{p}, \phi(t))}{\partial u(t - n_k)} \right|_{\phi(t)=0} & \dots & & b_{n_b} &= \left. \frac{\partial f_{NN}(\mathbf{p}, \phi(t))}{\partial u(t - n_k - n_b)} \right|_{\phi(t)=0} \\ -a_1 &= \left. \frac{\partial f_{NN}(\mathbf{p}, \phi(t))}{\partial y(t - 1)} \right|_{\phi(t)=0} & \dots & & -a_{n_a} &= \left. \frac{\partial f_{NN}(\mathbf{p}, \phi(t))}{\partial y(t - n_a)} \right|_{\phi(t)=0} \end{aligned} \quad (2.27)$$

Dans le cas d'un réseau de neurones avec une seule couche (voir l'équation (2.6)), les équations précédentes nous donnent :

$$\begin{aligned}
f_{NN}(\mathbf{p}, \boldsymbol{\phi}(t)) &= f^1(b^1 + \mathbf{W}^1 \boldsymbol{\phi}(t)) \\
&= f^1\left(b^1 + \sum_{r=0}^{n_b} W_{(1+r)}^1 u(t - n_k - r) + \sum_{r=1}^{n_a} W_{(n_b+1+r)}^1 y(t - r)\right) \\
b_0 &= \frac{\partial f_{NN}(\mathbf{p}, \boldsymbol{\phi}(t))}{\partial u(t - n_k)} = \frac{\partial f^1(b^1 + \mathbf{W}^1 \boldsymbol{\phi}(t))}{\partial u(t - n_k)} \Big|_{\boldsymbol{\phi}(t)=0} \cdot W_1 \Rightarrow W_1 \\
&\vdots \\
-a_{n_a} &= \frac{\partial f_{NN}(\mathbf{p}, \boldsymbol{\phi}(t))}{\partial y(t - n_a)} = \frac{\partial f^1(b^1 + \mathbf{W}^1 \boldsymbol{\phi}(t))}{\partial y(t - n_a)} \Big|_{\boldsymbol{\phi}(t)=0} \cdot W_{(n_a+n_b+1)} \Rightarrow W_{(n_a+n_b+1)}
\end{aligned} \tag{2.28}$$

Si nous fixons la valeur de b nous avons autant d'équations que d'inconnues et, par conséquent, nous trouvons directement les valeurs des paramètres.

Pour un réseau de neurones avec deux couches, les équations sont plus complexes parce que nous avons maintenant beaucoup plus de paramètres que d'équations et, par conséquent, nous devons choisir des conditions supplémentaires si nous voulons trouver les paramètres :

$$\begin{aligned}
f_{NN}(\mathbf{p}, \boldsymbol{\phi}(t)) &= f^2(b^2 + \mathbf{W}^2 \mathbf{a}^1) / \mathbf{a}^1 = f^1(b^1 + \mathbf{W}^1 \boldsymbol{\phi}(t)) \\
f_{NN}(\mathbf{p}, \boldsymbol{\phi}(t)) &= f^2\left(b^2 + \sum_{s=1}^{n_h} W_s^2 a_s^1\right) \\
a_s^1 &= f^1\left(b_s^1 + \sum_{r=0}^{n_b} W_{s,(1+r)}^1 u(t - n_k - r) + \sum_{r=1}^{n_a} W_{s,(n_b+1+r)}^1 y(t - r)\right) \\
b_0 &= \frac{\partial f^2(b^2 + \mathbf{W}^2 \mathbf{a}^1)}{\partial u(t - n_k)} \Big|_{\boldsymbol{\phi}(t)=0} \cdot \sum_{s=1}^{n_h} W_s^2 \frac{\partial a_s^1}{\partial u(t - n_k)} \Big|_{\boldsymbol{\phi}(t)=0} \\
&\vdots \\
-a_{n_a} &= \frac{\partial f^2(b^2 + \mathbf{W}^2 \mathbf{a}^1)}{\partial y(t - n_a)} \Big|_{\boldsymbol{\phi}(t)=0} \cdot \sum_{s=1}^{n_h} W_s^2 \frac{\partial a_s^1}{\partial y(t - n_a)} \Big|_{\boldsymbol{\phi}(t)=0}
\end{aligned} \tag{2.29}$$

Nous pouvons simplifier ces équations en faisant les hypothèses suivantes sur les fonctions de sortie des neurones :

$$f^c(x) \Big|_{x=0} = 0 \quad \frac{\partial f^c(x)}{\partial x} \Big|_{x=0} = 1 \tag{2.30}$$

Ces hypothèses sont valables avec les fonctions de sortie de type sigmoïde et linéaire (voir la figure 2.3). Si nous prenons les biais initiaux égaux à zéro ($b^c = 0$), les équations (2.29) deviennent alors :

$$\begin{aligned}
b_0 &= \sum_{s=1}^{n_h} W_s^{(2)} W_{s,1}^{(1)} \\
&\vdots \\
-a_{n_a} &= \sum_{s=1}^{n_h} W_s^{(2)} W_{s,(n_a+n_b+1)}^{(1)}
\end{aligned} \tag{2.31}$$

Le critère que nous avons adopté pour calculer les paramètres à partir de ces équations est une extension de l'équation (2.30) pour les cas où x ne serait plus zéro. En choisissant un coefficient K suffisamment grand et en initialisant les valeurs des paramètres avec les équations suivantes, nous pouvons effectivement considérer que la dérivée de la fonction de sortie est 1 :

$$W_s^{(2)} = \frac{K}{n_h}, \quad W_{s,1}^{(1)} = \frac{b_0}{K}, \dots, \quad W_{s,(n_a+n_b+1)}^{(1)} = -\frac{a_{n_a}}{K} \tag{2.32}$$

Il faut remarquer que ces équations donnent une matrice $\mathbf{W}^{(2)}$ de paramètres de rang 1, c'est-à-dire, que les lignes sont dépendantes.

2.4.3 Résultats de simulation.

Dans cette partie, nous allons présenter les résultats que nous avons obtenus. Pour cela, nous donnerons les modèles du système utilisés en simulation et ensuite les résultats obtenus.

2.4.3.1 Modèle du système linéaire

Cette fonction de transfert représente le gain entre la vitesse de rotation du moteur (rad/s) et le couple de référence donné aux régulateurs de courants :

$$\frac{\Omega(s)}{\Gamma_{ref}(s)} = F(s) = e^{-T_d s} \frac{1}{Js + f} \frac{s^2 + 2\zeta_a \omega_a s + \omega_a^2}{s^2 + 2\zeta_r \omega_r s + \omega_r^2} \frac{\omega_r^2}{\omega_a^2} \tag{2.33}$$

Et les valeurs des paramètres de ce premier modèle sont :

$$\begin{aligned}
J &= 1,501 \cdot 10^{-3} \text{ Kg} \cdot \text{m}^2 & \omega_r &= 697,7 \text{ rad / s} & \zeta_r &= 0,1173 & T_d &= 1,265 \text{ ms} \\
f &= 0,0233 \text{ Nms / rad} & \omega_a &= 463,9 \text{ rad / s} & \zeta_a &= 0,1736
\end{aligned}$$

Ce modèle a été obtenu à partir de la réponse fréquentielle en faisant une moyenne des spectres des signaux d'entrée et de sortie. Plus de détails sur ce modèle sont données dans la partie 3.4.2.1.

Nous utiliserons ce modèle pour valider nos algorithmes d'apprentissage dans le cas des réseaux linéaires par rapport aux méthodes d'identification traditionnelles (ARX, ARMAX et OE).

2.4.3.2 Modèles des systèmes non linéaires.

Nous avons utilisé deux modèles non linéaires en simulation. Le premier a été extrait du livre de J.M. Renders [REN-1994] où il étudie l'identification avec les réseaux de neurones avec des fonctions de sortie à base radiale. Son équation est la suivante :

$$y(k) = \frac{1}{5} \left[\frac{y(k-2)y(k-1)[y(k-1)+2,5]}{1 + y(k-1)^2 + y(k-2)^2} + u(k-2) \right] \quad (2.34)$$

Le deuxième modèle non linéaire repose sur le modèle linéaire présenté dans l'équation (2.33) mais en ajoutant les non linéarités de l'axe : le frottement sec, le couple de crantage de la machine synchrone, l'effet de la discrétisation dans les mesures de la vitesse et des courants, ... (voir chapitre 1).

Ce dernier modèle est un modèle représentatif de l'axe avec ses imperfections et nous permettra la validation de nos algorithmes pour cette application (voir la figure 3.2 pour un modèle complet de l'axe). Nous utiliserons ce modèle complet dans le chapitre 3 pour comparer les résultats de simulation avec ceux obtenus expérimentalement.

2.4.3.3 Tableaux comparatifs des résultats.

Pour chaque modèle du système, nous avons réalisé plusieurs essais en considérant tout d'abord que le système était affecté par un bruit de mesure habituellement suppose blanc. Dans un deuxième temps, nous considérons un cas plus réaliste quand le système est affecté par des perturbations.

a) Identification du système linéaire avec un modèle linéaire.

Pour les premiers essais, nous avons utilisé un modèle linéaire de réseau de neurones (une couche et linéaire). Nous avons plusieurs modèles possibles selon la présence ou non des bruits de mesure, la récurrence et la connaissance de l'ordre du système.

Pour vérifier la robustesse de l'algorithme face aux bruits, nous avons injecté un bruit qui représente les erreurs de mesure sur la sortie du système. La connaissance ou non de l'ordre du système nous permettra de valider l'algorithme d'apprentissage avec trop de paramètres.

Méthode		Levenberg-Marquardt	
		Sans récurrence	Avec récurrence
Système linéaire : eq.(2.33)			
Modèle linéaire (ordre connu et fixé en concordance) $n_a=3, n_b=3, n_k=2$	Pas de bruit	Parfait	Parfait
	Bruité	Biais dans les paramètres estimés à cause du bruit. Les modèles obtenus ne sont pas exploitables pour la synthèse de la commande. Modèle faux. (Equivalent à la méthode ARX).	Très bon et très proche du modèle réel si le bruit est réduit. Si nous commençons cet apprentissage avec un modèle sans récurrence, nous pouvons accélérer l'apprentissage postérieur, mais uniquement dans les cas où le modèle initial est stable. (Equivalent à la méthode OE).
Modèle linéaire (ordre inconnu, beaucoup de paramètres) $n_a=6, n_b=5, n_k=0$	Pas de bruit	Parfait	Vraiment très proche, mais les paramètres supplémentaires font perdre un peu de précision. Paramètres légèrement biaisés.
	Bruité	Modèle totalement faux.	Paramètres incorrects. Tests de corrélation bons. L'observation des coefficients indique le mauvais choix du retard et du nombre de paramètres.

Tableau 2.5: Identification du système linéaire de l'équation (2.33) avec un modèle linéaire.

En faisant un calcul rapide nous pouvons constater que nous avons 8 combinaisons au total présentées dans le tableau 2.5. Nous pouvons constater que s'il n'y a pas de bruits, tous les modèles sont valables. Cependant, ce contexte n'est pas très réaliste parce que nous ne pouvons pas ignorer les bruits existants dans la réalité, et encore moins sur une machine-outil.

En présence de bruits, la récurrence est nécessaire pour obtenir un modèle valable (voir la partie 2.4.2.2c). Cependant, si le nombre de paramètres est trop élevé, l'algorithme modélise le bruit également et les paramètres sont incorrects même si les tests de corrélation sont très bons. Si nous observons les pôles et les zéros du modèle identifié, nous pouvons distinguer les

pôles et les zéros du système d'une part et ceux de dynamiques très rapides d'autre part. Ces derniers essaient de modéliser le bruit.

La figure 2.19 montre les courbes de validation obtenues avec le modèle linéaire récurrent quand nous avons un bruit de mesure représenté par un signal aléatoire uniforme d'écart type 0,5rad/s et une entrée en SBPA.

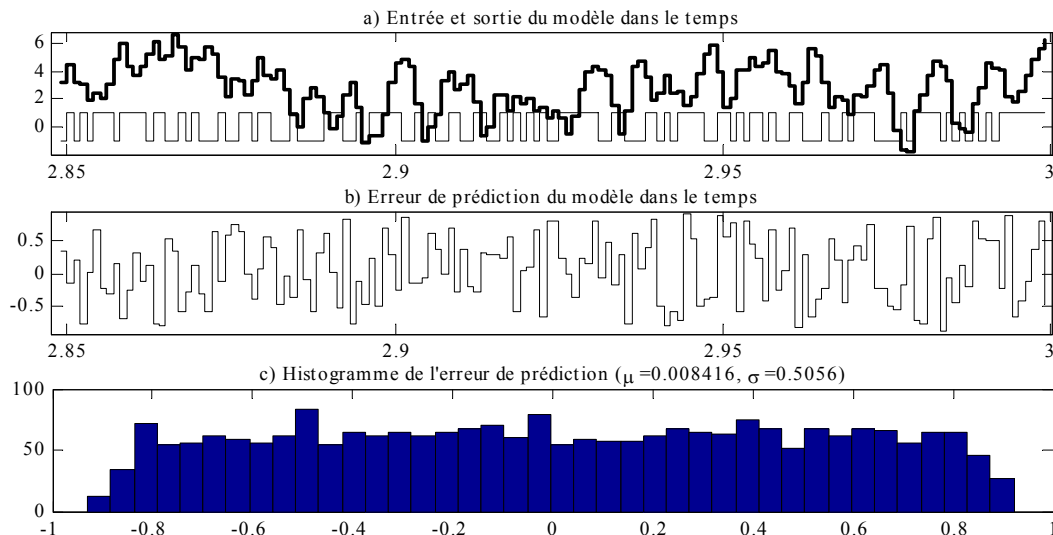


Figure 2.19: Validation de l'identification avec un modèle linéaire récurrent.

Si vous observez l'histogramme, vous pouvez vérifier que l'erreur est « presque » uniforme dans l'intervalle $\pm 0,86$ rad/s (dont l'écart type est $\sigma = 0,5056$ rad/s). Les tests de corrélation confirment la blancheur de l'erreur de prédiction et la decorrélation entre l'erreur et l'entrée.

Un point qu'il faut souligner dans ces essais est que les algorithmes d'apprentissage utilisés donnent des résultats qui sont équivalents aux méthodes d'identification traditionnelles « ARX » et « OE ». Par conséquent, nous avons validé nos algorithmes dans un contexte d'utilisation linéaire. Cependant, il faut aussi remarquer que ces résultats sont obtenus avec un algorithme plus lent que les méthodes traditionnelles car il est originalement conçu pour l'apprentissage non linéaire.

b) Identification du système linéaire avec un modèle non linéaire.

Dans le deuxième groupe d'essais, nous avons utilisé un modèle de réseau de neurones non linéaire (deux couches, sigmoïde dans la couche cachée et linéaire dans la couche de sortie). Nous avons fait une simulation pour chacune des 8 combinaisons précédentes (bruité ou non, récurrence ou non, trop de paramètres ou non). Cependant, cette fois, nous n'avons pas représenté tous les cas dans le tableau 2.6 parce que les résultats obtenus sont semblables au tableau 2.5.

Méthode		Levenberg-Marquardt	
		Sans récurrence	Avec récurrence
Système linéaire : eq.(2.33)			
Modèle Non linéaire	Pas de bruit	Avec un modèle initial reposant sur un système donné, il converge presque toujours, l'apprentissage est beaucoup plus rapide. Le modèle obtenu est valable uniquement dans le domaine d'apprentissage (voir commentaire).	Puisque nous commençons cet apprentissage avec un modèle sans récurrence et si le modèle obtenu a convergé, alors le modèle avec récurrence affine le modèle obtenu. Le modèle obtenu est valable uniquement dans le domaine d'apprentissage (voir commentaire).
	Bruité	Modèle faux.	Identique au cas sans bruit sauf que la précision du modèle n'est pas aussi bonne.

Tableau 2.6: Identification du système linéaire de l'équation (2.33) avec un modèle non linéaire.

Nous avons maintenant 90 paramètres à identifier pour le modèle non linéaire quand l'ordre est supposé connu ($n_a=3, n_b=3, n_k=2, n_h=10$) et 140 quand l'ordre est inconnu ($n_a=6, n_b=5, n_k=0, n_h=10$). Ce nombre élevé de paramètres couplé avec les non linéarités du modèle rend l'apprentissage plus lent (approximativement 3 à 10 fois plus lent par rapport au modèle linéaire).

Globalement, les résultats obtenus sont comparables à ceux obtenus dans le cas linéaire sauf que les modèles obtenus sont valables uniquement dans le domaine d'apprentissage. Illustrons ce problème par un exemple : la figure 2.20 montre les données d'apprentissage et le modèle identifié avec ces données. Le modèle identifié est assez bon (l'erreur de prédiction est blanche et d'écart type $\sigma=0.5\text{rad/s}$). Si vous observez la référence à suivre, son intervalle de variation est $\pm 7\text{rad/s}$ approximativement. Compte-tenu du fait que le réseau de neurones utilise les valeurs de la référence comme entrées du réseau pour la prédiction, le domaine

d'apprentissage pour ce réseau de neurones est dans l'intervalle fixé par la variation maximale de ses entrées ($\pm 7\text{rad/s}$).

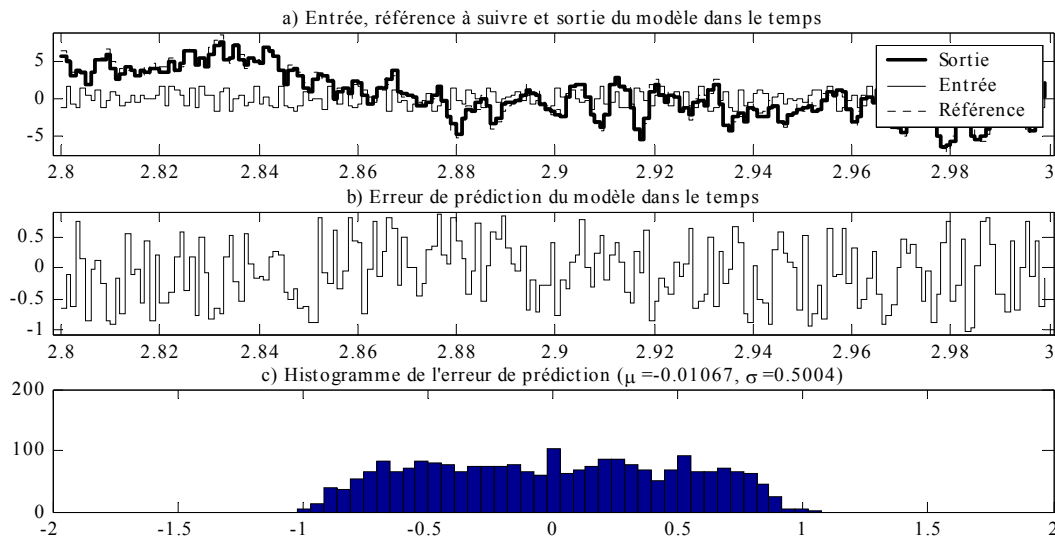


Figure 2.20: Données d'apprentissage et modèle résultant.

Si nous validons maintenant ce modèle avec une entrée en forme des dents de scie (figure 2.21), vous pouvez observer que l'intervalle de variation des entrées du réseau est beaucoup plus grand ($\pm 40\text{rad/s}$) et que l'erreur de prédiction est très grande.

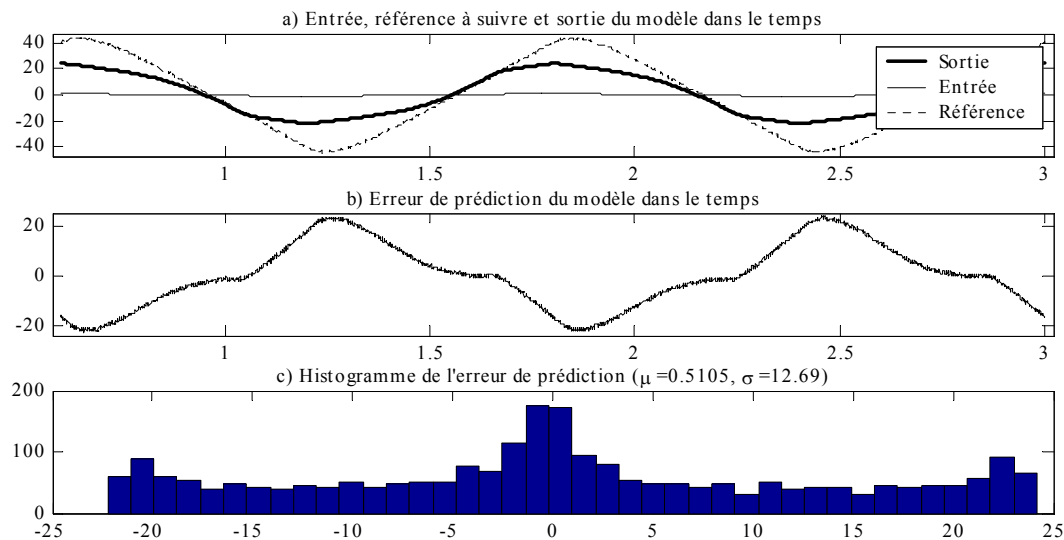


Figure 2.21: Validation du modèle trouvé de la figure 2.20 avec une entrée en forme des dents de scie.

D'où l'affirmation que le modèle est valable uniquement dans le domaine d'apprentissage. Nous sommes dans le même cas que l'exemple montré dans la figure 2.9a où le réseau doit extrapoler le comportement du système à partir des données d'apprentissage.

c) Identification du système non linéaire avec un réseau de neurones.

En suivant la même démarche qu'aux cas précédents, nous obtenons le tableau 2.7.

Méthode		Levenberg-Marquardt	
		Sans récurrence	Avec récurrence
Système non linéaire : eq.(2.34)			
Modèle non linéaire (ordre connu et fixé en concordance) $n_a=2, n_b=0, n_k=2,$ $n_h=25$	Pas de bruit	Parfait	Parfait
	Bruité	Biais dans les paramètres estimés à cause du bruit. Les modèles obtenus ne sont pas exploitables pour la synthèse de la commande. Modèle totalement faux.	Très bon et très proche du modèle réel si le bruit est réduit. Si nous commençons cet apprentissage avec un modèle sans récurrence, nous pouvons accélérer l'apprentissage postérieur, mais uniquement dans les cas où le modèle initial est stable.
Modèle non linéaire (ordre inconnu, beaucoup de paramètres) $n_a=4, n_b=3, n_k=0,$ $n_h=25$	Pas de bruit	Parfait	Vraiment très proche, mais les paramètres en plus font perdre un peu de précision. Paramètres légèrement biaisés.
	Bruité	Modèle totalement faux.	Tests de corrélation bons. L'observation des coefficients indique le mauvais choix du retard et du nombre de paramètres.

Tableau 2.7: Résultats d'identification du système non linéaire de l'équation (2.34).

Le tableau des résultats est presque identique aux cas précédents, mais dans ce cas, nous utilisons un modèle non linéaire pour identifier un système non linéaire. Cependant, le choix du signal d'excitation utilisé pour l'identification du modèle devient critique.

Si nous utilisons un signal de type SBPA, l'entrée prend deux valeurs uniquement et par conséquent le réseau doit interpoler les valeurs de sortie qu'aurait le système avec d'autres entrées. Dans le cas linéaire, ceci ne pose pas de problèmes justement à cause de la linéarité du système. Dans le cas non linéaire, avec seulement deux valeurs d'entrée, le réseau n'a pas suffisamment d'information pour pouvoir interpoler correctement et le modèle identifié n'est pas correct. Il en va de même pour tout type d'entrée qui n'assure pas une richesse de valeurs d'entrée (par exemple un signal carré).

Nous pouvons en conclure que pour l'identification des systèmes non linéaires, en plus de la richesse fréquentielle du signal d'excitation, nous avons besoin d'une richesse en ce qui concerne l'amplitude des valeurs d'entrée. Les signaux aléatoires constituent donc le type de

signal à préférer pour cette tâche. Les résultats du tableau ont été obtenus en utilisant un signal d'excitation aléatoire uniforme d'écart type $\sigma=1$.

La figure suivante montre les courbes de validation. Le signal d'entrée est maintenant en forme des dents de scie et nous avons un bruit de mesure Gaussien d'écart type $\sigma=0,05$.

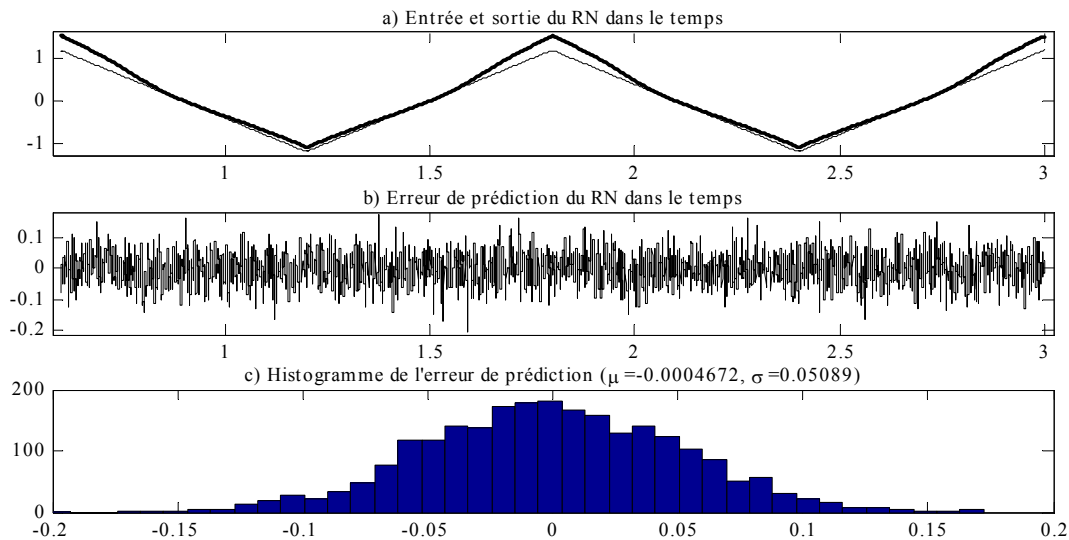


Figure 2.22: Validation de l'identification du système linéaire avec un modèle linéaire récurrent.

La figure 2.22a montre l'entrée en forme de dent de scie et l'évolution de la sortie du réseau avec cette entrée. La figure 2.22b montre l'erreur de prédiction du modèle. La figure 2.22c montre l'histogramme de l'erreur de prédiction.

Pour illustrer les conclusions sur la richesse du signal d'entrée, nous allons observer que la pente de la sortie du réseau change, ce qui est indicatif des non linéarités. Si l'algorithme utilise uniquement deux niveaux en entrée et les sorties correspondantes (par exemple une SBPA), il est clair qu'il n'aura pas d'informations sur le changement de pente qui existe dans le système non linéaire. Par conséquent, le modèle identifié sera incorrect.

L'évolution temporelle de l'erreur de prédiction ne nous offre pas beaucoup de renseignements. Cependant, l'histogramme a une forme gaussienne, la valeur moyenne du signal est $\mu=-4,6 \times 10^{-5}$ et son écart type est $\sigma=0,0509$. De plus, si nous examinons les tests de corrélation de l'erreur (dans la figure 2.23), nous pouvons en conclure que l'erreur de prédiction résultante est blanche. Avec tous ces renseignements, nous pouvons dire que l'erreur correspond au bruit de mesure du système et, par conséquent, l'identification est quasiment parfaite.

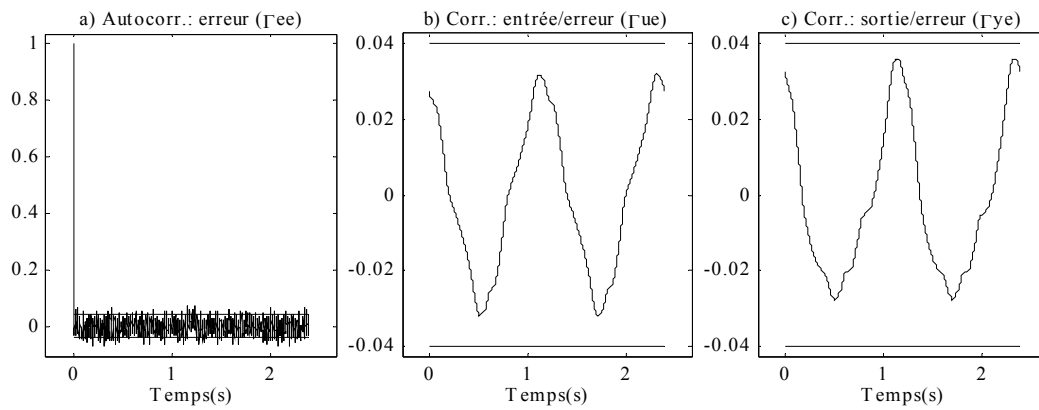


Figure 2.23: Tests de corrélation de l'erreur de prédiction de la figure 2.22b.

d) Identification des systèmes avec des perturbations externes.

Dans les cas précédents, nous avons considéré que l'erreur affectant le système était une erreur de mesure, souvent modélisée par un bruit blanc (figure 2.24). Dans la réalité, beaucoup de perturbations agissent directement sur l'entrée ou au sein du système.

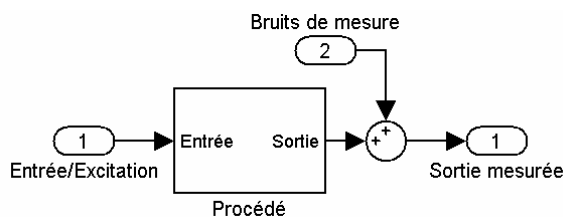


Figure 2.24: Modèle des bruits de mesure.

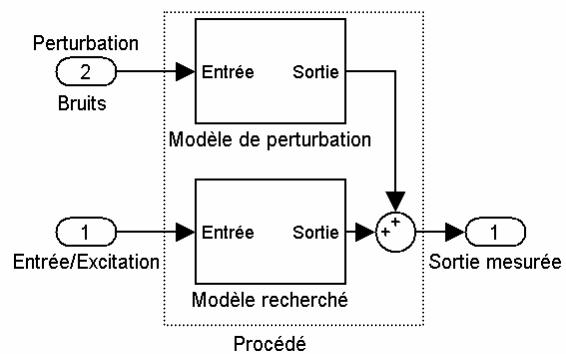


Figure 2.25: Modèle des bruits colorés et des perturbations.

Dans ce cas, le bruit à la sortie du modèle est appelé bruit coloré parce que le bruit blanc de l'entrée est filtré ou façonné par un modèle de bruit (figure 2.25).

Dans les méthodes d'identification traditionnelles, la structure ARMAX donne les meilleurs résultats, parce qu'en plus du modèle recherché, elle modélise aussi les bruits ou perturbations de manière à minimiser leur impact sur la qualité de l'identification (voir l'équation (2.24), [LAN-1993]).

Les modèles identifiés par le réseau ont la structure donnée par les équations (2.22) et (2.23). Par conséquent, nous devons nous attendre une diminution des performances face à des perturbations ou des bruits colorés.

La figure 2.26 montre le grand problème des structures ARX et OE face à des bruits colorés. Dans la pratique, les modèles du réseau de structure mentionnés sont capables d'obtenir un

modèle proche du système. Cependant, dans ces cas, l'erreur de prédiction résultante ne peut pas être blanche par définition et les tests de corrélation ne nous servent plus pour vérifier la qualité de l'identification.

Nous avons perdu un outil d'évaluation de la qualité de l'identification objectif (les tests de corrélation) et celui qui nous reste est une méthode subjective (l'appréciation des courbes).

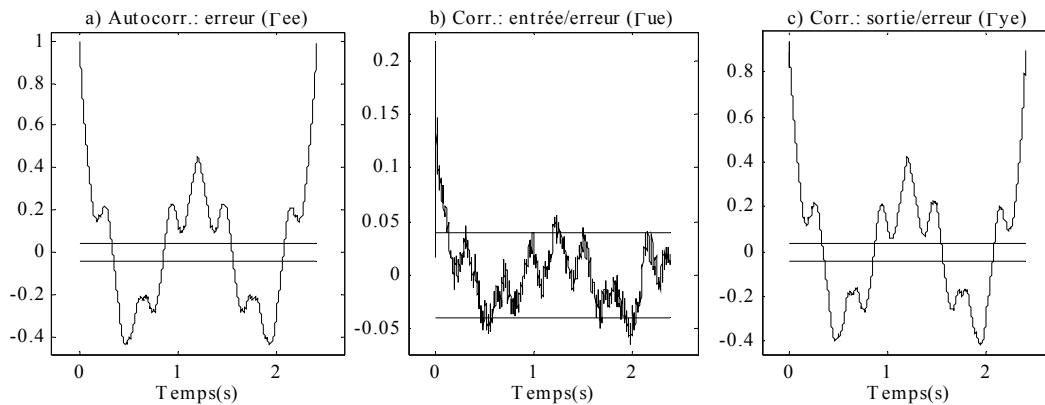


Figure 2.26: Tests de corrélation de l'erreur de prédiction d'une structure OE quand le bruit affectant le système est coloré.

Les courbes de la figure 2.26 ont été obtenues à partir de l'identification du système linéaire (2.33) avec un modèle linéaire et en ajoutant un bruit blanc uniforme filtré par la fonction de transfert suivante :

$$\frac{\Omega_p(s)}{B(s)} = \frac{\Omega_p(s)}{\Gamma_{res}(s)} = F(s) = \frac{1}{Js + f} \frac{\omega_r^2}{s^2 + 2\zeta_r \omega_r s + \omega_r^2} \quad (2.35)$$

Où $\Omega_p(s)$ est la perturbation de vitesse et $B(s)$ est le bruit blanc d'entrée. Les valeurs des paramètres de modèle de bruit sont égales à celles de l'équation (2.33). Ce modèle de bruit est justement le modèle de la relation entre la vitesse de l'axe et le couple résistant. Par conséquent, il correspond véritablement à une perturbation existant dans l'axe.

La figure 2.27 présente les résultats de l'identification avec les données d'apprentissage. Nous pouvons constater que l'amplitude de l'erreur obtenue est assez faible même si elle n'est pas blanche. La figure 2.28 montre maintenant le modèle identifié précédemment, avec un jeu de données sans bruit et une entrée différente. L'amplitude de l'erreur reste relativement faible par rapport à l'amplitude de la sortie et, par conséquent, nous pouvons considérer le modèle relativement bon (critère arbitraire et dépendant de l'application pour laquelle le modèle est utilisé).

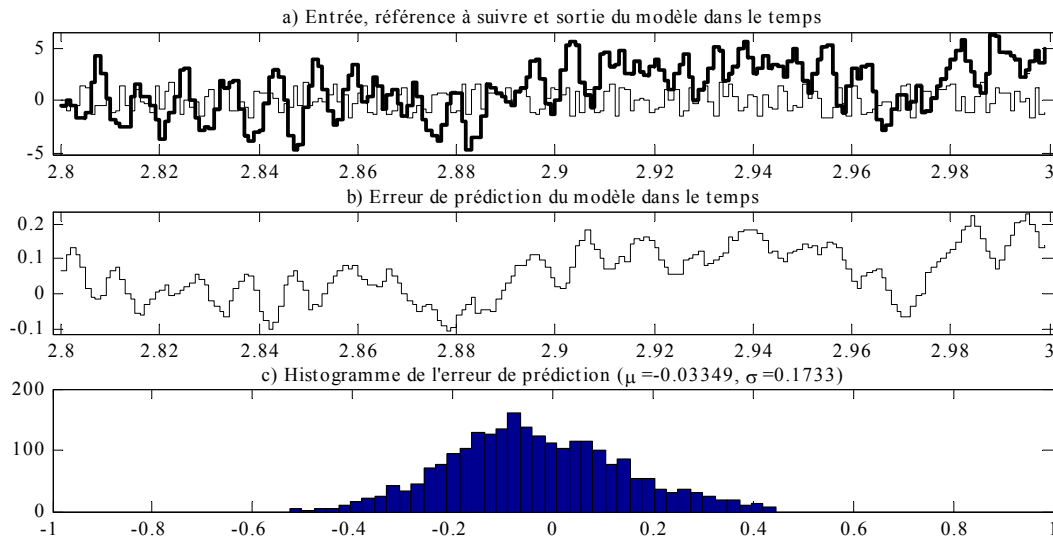


Figure 2.27: Validation du modèle linéaire identifié avec les données d'apprentissage (1^{er} étape).

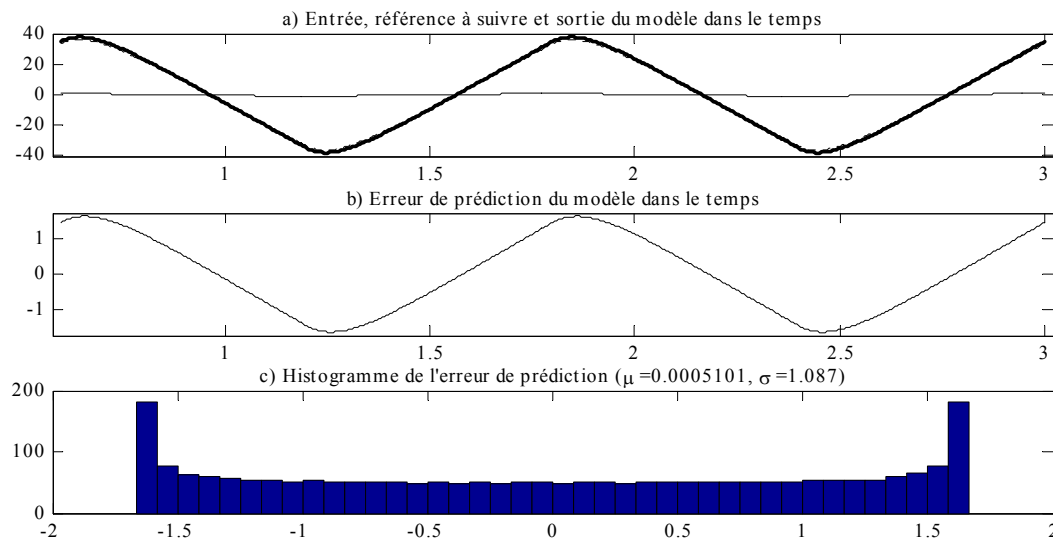


Figure 2.28: Validation du modèle linéaire identifié avec un autre jeu de données sans bruit (2^{ème} étape).

Il faut noter que les structures ARX et OE identifient un modèle qui combine le modèle recherché et le modèle de bruit selon l'importance des amplitudes de leur sortie. Ainsi, si la variance du bruit coloré est plus grande que la variance de la sortie du système, alors ces méthodes ont le risque de modéliser plutôt le bruit que le système.

Les cas d'identification non linéaire donnent des résultats semblables.

2.5 Bilan.

Les résultats obtenus ont été très bons (cas linéaire) ou bons (cas non linéaire) avec un bruit de sortie blanc. D'une part, les modèles linéaires obtenus sont tout à fait comparables à ceux obtenus avec les structures traditionnelles ARX et OE. D'autre part, le réseau est très capable de modéliser les non linéarités du système.

Cependant, dans les systèmes où les bruits de sortie ne sont pas des bruits blancs, le modèle de réseau choisi a plus de problèmes pour identifier le modèle réel. Malheureusement, l'axe que nous voulons modéliser est affecté par des perturbations et des bruits qui correspondent à ce cas et par conséquent, l'obtention d'un modèle non linéaire de l'axe avec ses imperfections semble extrêmement difficile. Il faudrait envisager la possibilité de réaliser un modèle de réseau de neurones avec la structure ARMAX pour pallier ce problème.

Dans le chapitre 3, nous allons vérifier avec ces conditions des résultats expérimentaux et étudier les problèmes d'un point de vue plus pratique.

Dans ce chapitre où tout est réalisé en simulation, nous n'avons pas les éléments permettant d'affirmer que les réseaux que nous avons choisis peuvent être implantés en temps réel (compatibilité entre le temps d'apprentissage et la vitesse d'évolution du système, compatibilité entre temps de calcul du réseau et période d'échantillonnage de la commande). Ces points seront explorés dans le chapitre 3 dédié à une mise en œuvre réelle des réseaux de neurones sur processeur de signal.

2.6 Bibliographie.

- [BER-1990]: L. BERKE, P. HAJELA
Application of artificial neural nets in structural mechanics.
Shape and layout optimization of structural systems (CISM Italy). 1990.
- [BER-1993]: H. BERSINI, J.P. NORDVIK, A. BONARINI
A simple direct adaptive fuzzy controller derived from its neural equivalent.
Proceedings of the IEEE International Conference on Fuzzy Systems, pages 345-350. 1993.
- [BRO-1970]: C.G. BROYDEN
The Convergence of a Class of Double-rank Minimization Algorithms.
J.Int. Maths.Applics., Vol.6, pages 76-90. 1970.
- [CAR-1993]: W.C. CARPENTER, J-F.M. BARTHELEMY
A comparison of polynomial approximations and artificial neural nets as response surfaces.
Structural Optimization, Vol.5, pages 166-174. 1990.
- [CHA-1992]: C. CHARALAMBOUS
Conjugate gradient algorithm for efficient training of artificial neural networks
IEEE Proceedings, Vol.139, n°3, pages 301-310. 1992.
- [DEM-1998] H. DEMUTH, M. BEALE
Neural Network Toolbox User's Guide.
The Mathworks, Inc. 1998
- [FAH-1990]: S.E. FAHLMAN, C. LEBIERE
The Cascade-Correlation Learning Architecture.
Advances in Neural Information Processing Systems. Morgan Kaufmann Publishers, pages 524-532. 1990.
- [FLE-1970]: R. FLETCHER
A New Approach to Variable Metric Algorithms.
Computer Journal, Vol.13, pages 317-322. 1970.
- [FOR-1997]: F.D. FORESEE, M.T. HAGAN
Gauss-Newton Approximation to Bayesian Regularization.
Proceedings of the International Joint Conference on Neural Networks, pages 1930-1935. 1997.
- [GOL-1970]: D. GOLDFARB
A Family of Variable Metric Updates Derived by Variational Means.
Mathematics of Computing, Vol.24, pages 23-26. 1970.
- [HOR-1989]: K. HORNIK, M. STINCHCOMBE, H. WHITE
Multilayered feedforward networks are universal approximators.
Neural Networks, Vol.2, pages 359-366. 1989.
- [HOR-1990]: K. HORNIK, M. STINCHCOMBE, H. WHITE
Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks.
Neural Networks, Vol.3, pages 551-560. 1990.

- [KAE-1996]: L.P. KAEHLING, M.L. LITTMAN, A.W. MOORE
Reinforcement learning: a survey.
Journal Of Artificial Intelligence Research, pages 237-285. 1996.
Adresse Internet : <http://www.ncstrl.org/>
- [KIR-1983]: S. KIRPATRICK, C.D. GELATT, M.P. VECCHI
Optimization by Simulated annealing
Science, Vol.220, n°4598, pages 671-680. 1983.
- [KOH-1987]: T. KOHONEN
Self-Organization and associative memory.
Springer. 1987.
- [Kwo-1997]: T.Y. KWOK, D.Y. YEUNG
Constructive algorithms for structure learning in feedforward networks for regression problems.
IEEE Transactions on Neural Networks, Vol.8, n°3, pages 630-645. 1997.
- [LAN-1993]: I.D. LANDAU
Identification et commande de systèmes.
Hermès. Paris. 1993.
- [LJU-1987]: L. LJUNG
System Identification : theory for the user.
Prentice-Hall. Englewood Cliffs. 1987.
- [McC-1943]: W.S. McCULLOCH, W.H. PITTS
A logical calculus of ideas immanent in nervous activity.
Bulletin of Mathematical Biophysics, Vol.5, pages 115-133. 1943.
- [MAR-1963]: D.W. MARQUARDT
An algorithm for least-squares estimation of nonlinear parameters.
SIAM Journal On Applied Mathematics, Vol.11, pages 431-441. 1963.
- [NAR-1990]: K.S. NARENDRA, K. PARTHASARATHY
Identification and control of dynamic systems using neural networks.
IEEE Transactions on Neural Networks, Vol.1, n°1, pages 4-27. 1990.
- [POW-1977]: M.J.D. POWELL
Restart procedures for conjugate gradient method.
Mathematical Programming, Vol.12, pages 241-254. 1977.
- [REE-1993]: R. REED
Pruning algorithms - A survey.
IEEE Transactions On Neural Networks. Vol.4, n°5, pages 740-747. 1993.
- [REN-1994]: J.M. RENDERS
Algorithmes génétiques et réseaux de neurones.
Hermès. Paris. 1995.
- [RIE-1993]: M. RIEDMILLER, H. BRAUN
A direct adaptive method for faster backpropagation learning: The RPROP algorithm.
Proceedings of the IEEE International Joint Conference on Neural Networks, 1993.

- [ROG-1993]: J.S. ROGER JANG, C.T. SUN
Functional equivalence between radial basis function networks and fuzzy inference systems.
IEEE Transactions on Neural Networks, Vol.4, n°1, pages 156-159. 1993.
- [ROS-1961]: F. ROSENBLATT
Principles of Neurodynamics.
Spartan Press. Washington DC. 1961.
- [ROS-1996]: F. ROSSI
Calcul de différentielles dans les réseaux de neurones généralisés : algorithmes, complexité,
implantation logicielle et applications.
Thèse de doctorat présentée à l'Université Paris IX Dauphine. 1996.
Adresse Internet : <http://www.ceremade.dauphine.fr/~rossi/thesis.html>
- [SAA-1998]: D. SAAD
On-line learning in Neural Networks
Cambridge University Press. 1998
- [SAR-1997]: W.S. SARLE et autres
Neural Network FAQ, part 1 of 7 : Introduction.
Periodic posting to the Usenet group comp.ai.neural-nets. 1997
Adresse Internet : <ftp://ftp.sas.com/pub/neural/FAQ.html>
- [SEA-1994]: J.R. SEARLE
The Rediscovery of the Mind
Bradford Books. 1994.
- [SHA-1970]: D.F. SHANNO
Conditioning of Quasi-Newton Methods for Function Minimization
Mathematics of Computing, Vol.24, pages 647-656. 1970.
- [SHE-1992]: S. SHEKARS, M.B. AMIN
Generalization by neural networks
IEEE Transaction on Knowledge and Data Engineering, Vol.4, pages 177-185. 1992.
- [SHE-1994]: G.M. SHEPHERD
Neurobiology, 3rd Edition.
Oxford University Press. 1994.
- [SIN-1989]: S. SINGAL, L. WU
Training multilayer perceptrons with extended Kalman filter.
Advances in Neural Information Processing Systems, pages 133-140. Morgan Kauffman. 1989.
- [SWI-1991]: R.A. SWIFT, S.M. BATILL
Application of neural networks to preliminary structure design.
32nd Structures, Structural Dynamics and Materials Conference, pages 335-343. Baltimore, 1991.
- [WAN-2000]: E.A. WAN, R. VAN DER MERWE
The Unscented Kalman Filter for Nonlinear Estimation.
Adaptive Systems for Signal Processing, Communications and Control Symposium. 2000.
Adresse Internet: http://cslu.cse.ogi.edu/publications/ps/wan_AS-SPCC_2000.ps.gz

Chapitre 3: Validation expérimentale des résultats.

3.1 Introduction.

Ce chapitre a pour but de montrer les résultats de l'application des réseaux de neurones à l'identification du modèle d'une machine-outil. Pour arriver à ce point, nous procéderons comme suit :

Tout d'abord, nous présenterons les modèles utilisés en simulation, le banc d'essais des moteurs et le banc d'essais machine-outil, avec les paramètres qui nous serviront pour la simulation. Puis, nous validerons les modèles de simulation en les comparant avec les résultats expérimentaux. Ensuite, nous montrerons les résultats de l'application des algorithmes d'identification développés sur ces bancs d'essai en faisant le lien avec les résultats de simulation. Nous verrons aussi les problèmes de mise en œuvre de ces algorithmes en temps réel.

Nous finirons en résumant les conclusions obtenues.

3.2 Outils de simulation.

La plupart des modèles de simulation développés à l'occasion de cette thèse ont été programmés en langage "C" et implantés sous le logiciel MATLAB/Simulink. Ce choix a été fait pour les raisons suivantes :

- Disponibilité de modèles déjà développés en langage "C" pour le logiciel de simulation Xmath[®] 5.0.
- Nécessité de passage au logiciel de simulation MATLAB[®] 4.2c (c'était le seul logiciel de simulation disponible en commun entre les deux centres de recherche de cotutelle).
- Avantage de la rapidité de calcul des modèles compilés par rapport aux modèles réalisés à partir des éléments Simulink.
- Rapidité de la mise en œuvre des algorithmes de commande dans le banc d'essais. Nous utilisons exactement les mêmes fichiers pour la simulation et les essais expérimentaux.

Malheureusement, ce choix nous a obligé à faire certains sacrifices :

- Nous n'avons pas utilisé le langage de plus haut niveau intégré dans le logiciel de simulation MATLAB.
- Nous avons dû programmer tous les outils nécessaires pour le calcul et l'apprentissage des réseaux de neurones.

3.2.1 Modèles MATLAB/Simulink.

L'interface entre notre code et le logiciel MATLAB a été réalisée en utilisant les fichiers MEX [MAT-1998]. Les fichiers MEX sont des bibliothèques de fonctions compilées qui peuvent être exploitées à partir de MATLAB.

La figure 3.1 montre le diagramme de blocs du modèle Simulink utilisé dans la simulation de l'axe avec les algorithmes d'identification et commande.

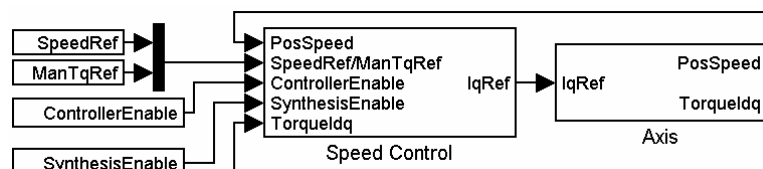


Figure 3.1: Modèle de simulation du servomécanisme entier.

Le bloc d'autorisation des correcteurs (« ControllerEnable ») a été programmé de manière à pouvoir faire les choix suivants :

- Pas de correcteur.

- Correcteur Proportionnel/Intégral (PI).
- Correcteur généralisé (RST).
- Correcteur Tout ou Rien.
- Addition d'une précompensation de couple.

Les blocs « SpeedRef » et « ManTqRef » imposent respectivement une référence de vitesse ou une référence de couple. En combinant avec le bloc de choix du correcteur, nous pourrions par exemple, commencer une simulation sans correcteur de vitesse en donnant la référence de couple et puis passer à un contrôle de vitesse avec RST suivant une référence de vitesse.

Le bloc d'autorisation synthèse (« SynthesisEnable ») est utilisé pour activer ou désactiver l'apprentissage des réseaux de neurones suivants :

- Réseau de neurones utilisé pour la modélisation des oscillations statiques du couple (par exemple le couple de crantage).
- Réseau de neurones pour l'identification en temps réel du modèle de l'axe.

Les structures des réseaux de neurones et les coefficients des correcteurs sont donnés dans le bloc de contrôle de vitesse (« Speed Control »). Par exemple, nous pouvons choisir entre un modèle parallèle, dont la sortie est estimée sans utiliser l'information sur la sortie réelle du système, ou un modèle série/parallèle dont la sortie est prédite en utilisant la sortie réelle du système (voir équations (2.26) et (2.27) de la partie 2.4.2.2).

La figure 3.2 présente le schéma des blocs de l'axe mécanique, du moteur, du variateur et de l'alimentation et du contrôle vectoriel des courants. Ces modèles ont été conçus pour représenter les perturbations et imperfections introduites au chapitre 1 de cette thèse.

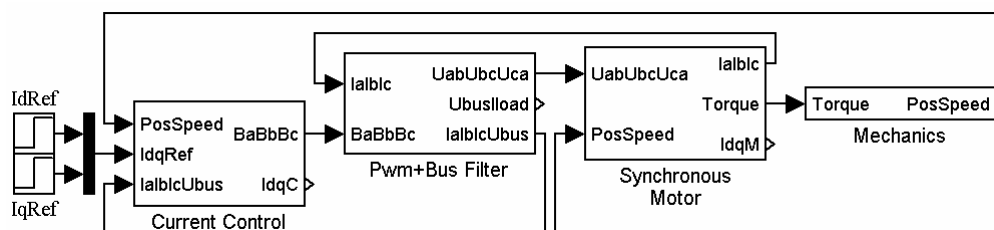


Figure 3.2: Modèle de simulation de l'axe et contrôle des courants.

Nous avons développé plusieurs modèles mécaniques sans jeux pour répondre à nos besoins :

- Un système mécanique qui impose une vitesse constante (très grande inertie).
- Un système mécanique correspondant à un axe rigide (premier ordre).
- Un système mécanique correspondant à un axe souple.

- Un système mécanique correspondant à un axe rigide et un filtre simulant la résonance et/ou l'antirésonance mécaniques. La différence entre l'axe souple et ce système mécanique est que, dans l'axe souple, la résonance et l'antirésonance sont liées par les paramètres de l'axe souple et, dans ce cas, elles sont complètement découplées.

De plus, tous ces modèles tiennent compte des frottements secs.

Le modèle de moteur synchrone prend en compte les perturbations et imperfections suivantes :

- Harmoniques du flux créés par les aimants permanents du rotor.
- Harmoniques de l'inductance statorique.
- Modèle approximatif du couple de crantage.

Différentes options peuvent être choisies pour simuler l'ensemble bus continu/onduleur MLI :

- Choix entre une tension de bus constante ou la simulation du redresseur triphasé et du filtre qui génèrent la tension de bus.
- Limitation de la tension de bus en utilisant une résistance de freinage avec un correcteur à hystérésis (seulement avec une tension de bus variable).
- Choix entre un modèle moyen de l'onduleur MLI ou un modèle topologique simulant les commutations.
- Modélisation de la perturbation introduite par les temps de garde avec la possibilité d'avoir des temps de garde déséquilibrés.
- Mesure de courant synchrone ou décalée par rapport aux instants de commutation de la MLI.

Le contrôle de courant possède les caractéristiques suivantes :

- Réglage des correcteurs PI de contrôle des courants.
- Modélisation des erreurs des capteurs de courant par un décalage et une erreur sur le gain du capteur.
- Possibilité d'ajouter sélectivement une quantification sur la mesure des courants, la position et la référence de tension fournie à la MLI.
- Activation de la compensation de la FEM du moteur.
- Choix entre le découplage des courants par mesure directe des courants, par estimation des courants et fonctionnement sans découplage.
- Possibilité d'utiliser une mesure de la tension de bus pour compenser ses variations.

- Choix entre deux types de surmodulation ou désactivation de la gestion de la limitation de tension.

Le dernier modèle utilisé pendant cette thèse a été développé pour simuler des perturbations de l'axe en boucle ouverte. Par conséquent, il utilise la même structure que la figure 3.2 mais en changeant le bloc de contrôle des courants par un bloc générant des références de tension sinusoïdales et triphasées.

Un effort considérable a été fait pour simplifier l'utilisation de ces modèles. Ainsi chaque modèle a une boîte de dialogue propre pour l'entrée de paramètres. La figure 3.3 montre la boîte de dialogue du bloc mécanique de l'axe rigide.

Par ailleurs, des séries de simulations peuvent être réalisées sans intervention de l'utilisateur grâce à un fichier de paramétrage.

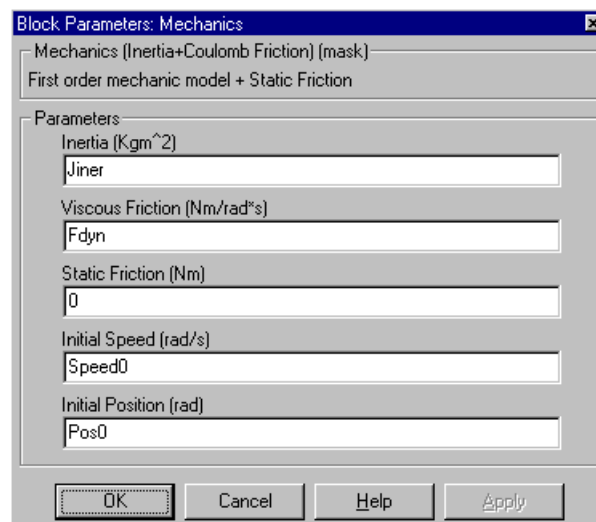


Figure 3.3: Fenêtre de dialogue pour l'entrée des paramètres du modèle d'un axe mécanique rigide.

3.2.1.1 Modèle simplifié.

Le modèle simplifié est construit à partir des modèles que nous avons déjà présentés. Il comprend :

- Modèle mécanique d'un axe rigide sans frottements secs.
- Pas d'harmoniques dans l'inductance statorique et le flux créé par les aimants permanents.
- Pas de couple de crantage dans la machine.
- Tension de bus constante.
- Mesures synchronisées de courant.

- Mesure parfaite des courants (sans discrétisation, décalage, ni erreur de gain).

Ce modèle a été utilisé surtout au chapitre 2 pour les simulations montrant les perturbations et les harmoniques produits sur le couple. A ce modèle, nous avons ajouté les perturbations dont nous voulions montrer l'effet.

Cependant, dans les cas où nous avons montré les perturbations concernant les parties électriques, nous avons choisi d'utiliser une mécanique à vitesse constante pour simplifier l'obtention des données en régime permanent et éviter les effets mécaniques.

3.2.1.2 Modèle détaillé.

Ce modèle a été conçu pour représenter le plus fidèlement possible l'axe étudié avec ses imperfections. Les caractéristiques principales sont :

- Modèle mécanique rigide avec un filtre pour simuler un axe souple.
- Moteur avec harmoniques de flux créés par les aimants permanents.
- Présence du couple de crantage au moteur.
- Filtre de bus avec résistance de dissipation en freinage.
- Mesure non-synchrone des courants.
- Retards de mesure des signaux pour le correcteur de vitesse.

3.3 Mise en œuvre du banc d'essais moteur.

3.3.1 Objectifs et description.

Le premier objectif de ce banc d'essais est la vérification des différents algorithmes avant de les implanter sur la machine-outil.

L'autre objectif de ce banc d'essais est la caractérisation du contrôle de couple et la mesure des paramètres pour la simulation. Enfin, à l'aide de ce banc, nous pouvons réaliser des essais qui ne sont pas possibles dans la machine-outil à cause du parcours limité des axes.

Le banc d'essais, tel que vous pouvez l'observer dans la figure 3.4, est constitué de deux moteurs synchrones à aimants permanents FAGOR AXM 3.09.6 qui sont montés l'un face à l'autre sur un support métallique en forme de 'O'. Un accouplement mécanique fait l'union entre les axes des deux moteurs pour le transfert du mouvement et des forces. Les tensions sont fournies par deux variateurs FAGOR AXD 2.50.



Figure 3.4: Banc d'essais des algorithmes.

Les algorithmes d'identification sont appliqués et testés sur un des moteurs. Le deuxième moteur travaille comme charge variable et configurable.

La commande de vitesse, la génération des signaux d'excitation pour l'identification, l'acquisition des données et les algorithmes des réseaux de neurones sont réalisés sur une carte DSP DS1102 de dSPACE qui est insérée dans un PC. Cet ordinateur fait à la fois l'interfaçage entre l'utilisateur et les algorithmes et l'acquisition des données pour la visualisation et le traitement postérieur des résultats. La figure 3.5 montre la structure interne de la carte DS1102. Les quatre convertisseurs analogiques/numériques permettent

l'acquisition des vitesses et des images de couple fournies par les variateurs et 2 des 4 convertisseurs numériques /analogiques permettent la sortie des consignes vers les variateurs.

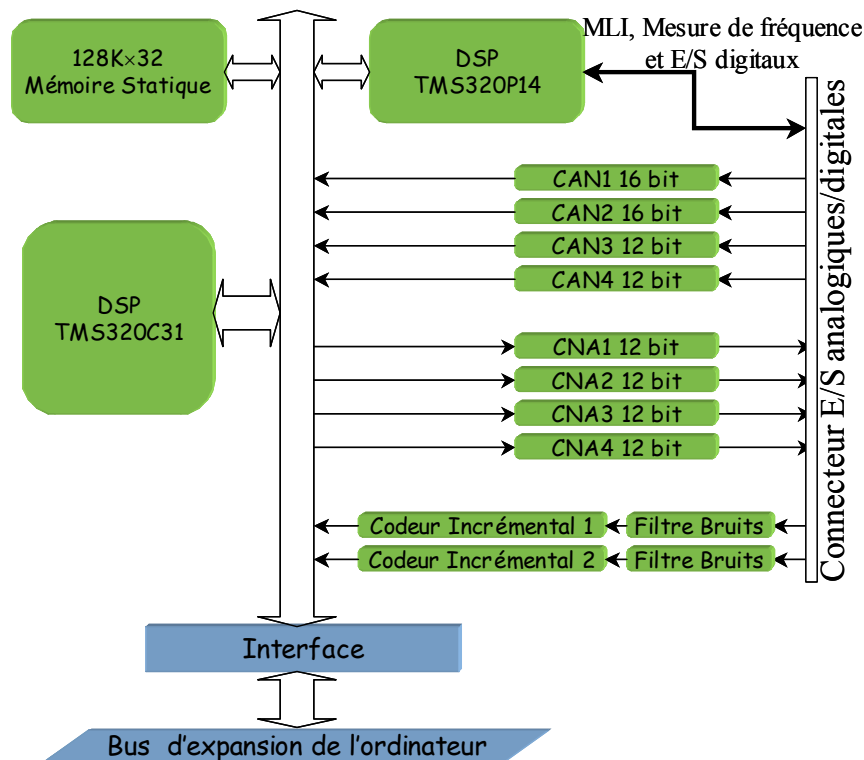


Figure 3.5: Structure interne de la carte DSP DS1102 de dSPACE.

Autres détails de la carte : elle dispose de 512Ko de mémoire ultrarapide, d'un processeur de signal secondaire pour la génération des signaux de MLI et la mesure de fréquences, de deux circuits de lecture du signal des codeurs de position et d'une mémoire à double accès pour la communication avec l'ordinateur.

La communication des données entre la carte DSP et les variateurs est réalisée par voie analogique (avec toutes ses conséquences sur la qualité de passage de l'information). La ligne série est utilisée pour deux fonctions :

- acquisitions de données à basse fréquence d'échantillonnage et de nombre d'échantillons réduits directement à partir de l'ordinateur.
- configuration du variateur : désactivation du contrôle de vitesse et choix des signaux transférés par voie analogique.

Malheureusement, nous n'avons pas pu récupérer la position de manière digitale parce que le variateur ne fournissait pas les signaux des codeurs de position et par conséquent, dans toutes les mesures où nous avons besoin de la position, nous avons dû l'obtenir en intégrant la mesure analogique de la vitesse. Nous verrons que ceci a été une source très importante

d'erreurs, qui pour certains essais expérimentaux, ont beaucoup dégradé la qualité des mesures.

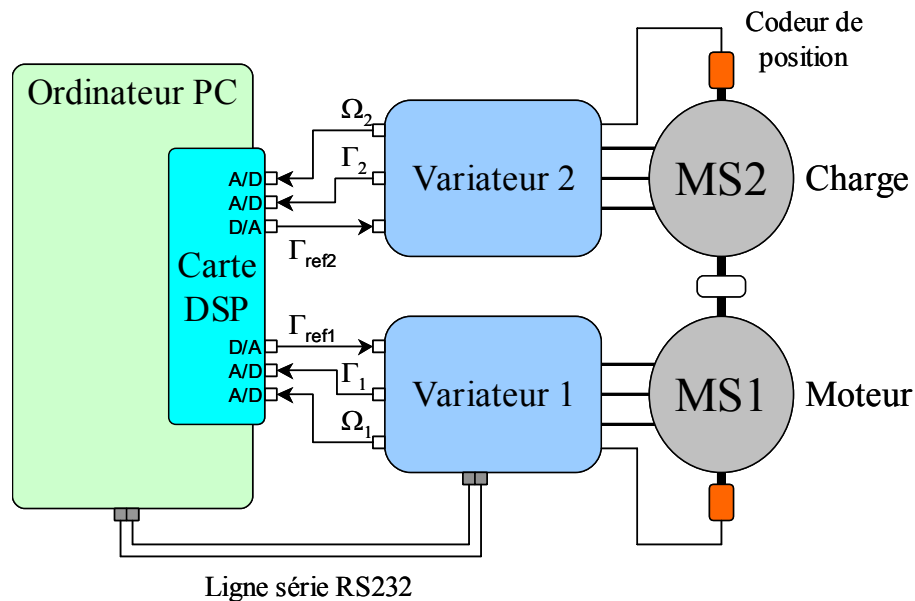


Figure 3.6: Diagramme d'interconnexion du banc d'essais des moteurs.

La figure 3.6 montre le diagramme d'interconnexion du banc d'essais. L'image de couple ou le couple estimé est obtenue par le variateur en mesurant le courant de quadrature du contrôle vectoriel et en appliquant le coefficient de couple du moteur.

3.3.2 Validation des modèles de simulation du moteur/variateur.

3.3.2.1 Paramètres mécaniques.

a) Inertie, frottement sec et frottement visqueux.

Les tests réels vont être conduits dans la machine-outil et les paramètres mécaniques dans ce cas sont différents. Par conséquent, nous avons davantage cherché des ordres de grandeur dans les cas du banc d'essai moteur que des valeurs précises.

Pour cela, nous avons mesuré la vitesse de l'axe et le couple du moteur quand la référence de couple est un signal carré et nous avons pris un modèle de mécanique rigide (3.1).

$$\Gamma_m = J \frac{d\Omega}{dt} + f\Omega + \Gamma_s \quad (3.1)$$

En régime établi, nous obtenons les valeurs des frottements et en analysant le régime transitoire nous obtenons la valeur de l'inertie.

$$J = 1,66 \cdot 10^{-3} \text{ Kg} \cdot \text{m}^2 \quad f = 7,24 \cdot 10^{-3} \text{ Nms / rad} \quad \Gamma_s \approx 0 \text{ Nm} \quad (3.2)$$

La précision de calcul de ces valeurs est très faible. Par exemple, selon une estimation grossière, nous avons 80% d'erreur pour le frottement visqueux.

3.3.2.2 Paramètres du moteur et du variateur.

a) Paramètres du moteur FAGOR AXM 3.09.6.

La plupart des paramètres du moteur sont donnés dans la documentation du constructeur du moteur. Les exceptions ont été les harmoniques de FEM et le couple de crantage.

$$\begin{aligned} N &= 6000 \text{ } \frac{\text{v}}{\text{min}} & I_{\text{nominal}} &= 7,4\text{A} & \Gamma_{\text{nominal}} &= 6,5\text{Nm} \\ p &= 4 & I_{\text{maximum}} &= 36\text{A} & \Gamma_{\text{maximum}} &= 32\text{Nm} \\ R &= 1\Omega & L_0 &= 1,5\text{mH} & M_0 &= -0,75\text{mH} \end{aligned} \quad (3.3)$$

Où p est le nombre de paires de pôles, R et L_0 sont la résistance et l'inductance par phase et M_0 est l'inductance mutuelle entre deux phases (voir l'équation (1.2)).

Nous avons considéré que la machine n'a pas d'harmoniques d'inductance parce que l'entrefer est constant. Cependant, nous n'avons pas les moyens nécessaires pour vérifier ces hypothèses (la mesure de la valeur de l'inductance en fonction de la position du moteur requiert des essais assez sophistiqués pour avoir une précision acceptable [FLE-1987]).

Pour obtenir la constante de FEM avec ses harmoniques, nous avons fait tourner la machine à vitesse constante et à vide et nous avons mesuré les tensions entre phases. A partir de celles-ci, nous avons retrouvé la tension d'une des phases avec l'aide de l'équation (3.4) et nous avons fait une analyse spectrale de cette tension.

$$\hat{V}_a = \frac{V_{ab} - V_{ca}}{3} \quad (3.4)$$

L'équation (3.4) ne peut pas être utilisée pour retrouver les harmoniques d'ordres multiples de 3 parce que la tension estimée ne les contient pas. Cependant, pour un moteur symétrique et équilibré, l'équation du flux (l'équation (1.9)) n'a pas ces harmoniques parce que la somme des flux créés doit être nulle par respect de la loi de conservation du flux.

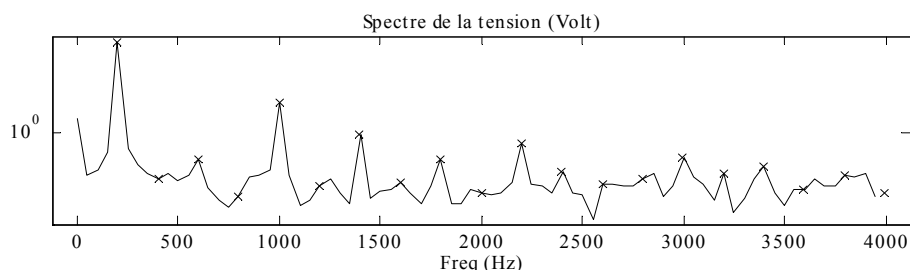


Figure 3.7: Spectre de la tension de phase.

La figure 3.7 montre le spectre de cette tension. A partir de ce spectre, nous obtenons les valeurs des harmoniques et nous retrouvons les constantes de FEM en combinant les équations (1.3) et (1.9) (en Volt/rad/s) :

$$\begin{aligned} K_1 &= 0,0967355 & K_3 &= 0 & K_5 &= 0,00370967 \\ K_7 &= 0,000806143 & K_9 &= 0 & K_{11} &= 0,00041164 \end{aligned} \quad (3.5)$$

Dans l'obtention des paramètres du couple de crantage, nous avons pu compter le nombre d'encoches en tournant l'axe de la machine à la main. Ce moteur a un couple de crantage suffisamment large pour le faire. Nous avons compté $N_e=24$ encoches au stator.

Selon le modèle présenté par l'équation (1.11) du chapitre 1, il nous faut encore trouver l'amplitude et le déphasage de la fréquence fondamentale du couple de crantage. Pour cela, nous avons réalisé l'acquisition du couple de référence fourni par le correcteur de vitesse quand la machine tourne à vitesse constante, suffisamment basse pour que les fréquences significatives du couple de crantage soient dans la bande passante du correcteur de vitesse. Théoriquement, le correcteur de vitesse compense alors la perturbation produite par le couple de crantage en fournissant une référence de couple opposée à cette perturbation.

Cependant, cette acquisition est complexe à réaliser avec précision. Le fondamental du couple de crantage est d'ordre relativement élevé (N_e fois la fréquence de rotation de la machine). Par conséquent, la seule solution est de faire tourner la machine à une très basse vitesse pour que les harmoniques du couple de crantage soient dans la bande passante du correcteur de vitesse. Néanmoins, le variateur a une erreur de mesure très grande à basses vitesses et par conséquent, la position estimée en intégrant la vitesse n'est pas fiable. Il faut trouver une vitesse de rotation de la machine qui réalise le compromis entre la qualité de l'estimation de la position et l'estimation du couple de perturbation la machine.

La figure 3.8 montre le couple de référence par rapport à la position quand la machine tourne à 30tour/min approximativement. Avec la courbe du couple, il apparaît aussi le couple de crantage reconstruit avec les paramètres que nous avons trouvés :

$$\Gamma_{cr} = 0,15\text{Nm} \pm 35\% \quad \delta_{cr} = -127^\circ \pm 25\% \quad (3.6)$$

Dans le spectre du couple de référence, nous pouvons observer que la valeur maximale de la perturbation du couple de référence arrive à 24 fois par tour de la machine, c'est-à-dire, le nombre d'encoches.

Il faut noter que la valeur du déphasage du couple de crantage doit être identifiée à chaque démarrage de la machine parce que les capteurs de position sont relatifs par rapport à une position initiale.

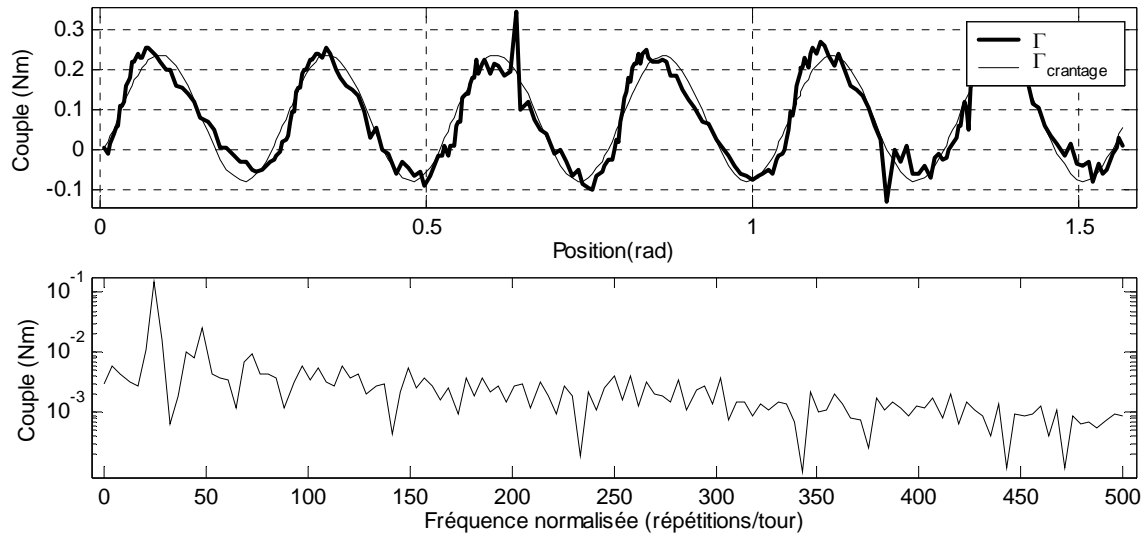


Figure 3.8: Couple mesuré et estimé et les harmoniques de couple mesuré.

b) Paramètres du variateur FAGOR AXD 2.50.

Le constructeur FAGOR nous a aimablement fourni des données internes au variateur pour la suite de ce travail. De plus, il a fait des modifications au programme du variateur pour que nous puissions inhiber le contrôle de vitesse intégré dans le variateur et agir directement sur la référence de couple de la machine.

Voici la liste des paramètres du variateur (voir figure 1.14) :

Paramètres du redresseur et du filtre du bus :

$$\begin{aligned}
 U_{bus} &= 540\text{Volt} & R_{freinage} &= 18\Omega / 370W \\
 R_{bus} &= 1m\Omega & L_{bus} &= 1mH & C_{bus} &= 680\mu F
 \end{aligned}
 \tag{3.7}$$

Seuils de l'hystérésis d'activation de la résistance de freinage :

$$U_{bus}|_{MIN} = 545\text{Volt} \quad U_{bus}|_{MAX} = 580\text{Volt}$$

Paramètres de la MLI (fréquence et temps de garde) :

$$F_{MLI} = 8kHz \quad t_g = 2\mu s \tag{3.8}$$

Paramètres des mesures des courants, de la position et de la vitesse (voir §1.3.3.6) :

$$\begin{aligned}
 \Delta I_{min} &= 0,019721A & \Delta \theta_{min} &= 5,99 \times 10^{-6} rad & \Delta \Omega_{min} &= 0,095874 rad / s \\
 T_e &= 62,5\mu s & t_{synchro} &= 12,5\mu s
 \end{aligned}
 \tag{3.9}$$

Où les valeurs Δx_{\min} sont les résolutions des mesures, T_e est la période d'échantillonnage et t_{synchro} est l'avancement de la mesure des courants par rapport au point de synchronisme de la MLI (voir figure 1.26).

Paramètres des correcteurs PI de contrôle des courants :

$$\begin{aligned} K_P &= 4,6 \text{ Volt/Nm} & K_I &= 2000 \text{ Volt/Nm/s} \\ U_{\text{lim}} &= 510 \text{ Volt} \end{aligned} \quad (3.10)$$

Où U_{lim} représente la tension maximale demandée en sortie du correcteur du courant et K_P et K_I sont respectivement les gains proportionnel et intégral du correcteur.

3.3.2.3 Comparaison des résultats expérimentaux et de simulation.

Maintenant que nous avons trouvé tous les paramètres, nous allons essayer de déterminer le domaine de validité du modèle de simulation par rapport au système réel. En pratique, il nous suffit qu'il soit suffisamment représentatif pour que les conclusions obtenues à partir des simulations soient extrapolables au système réel.

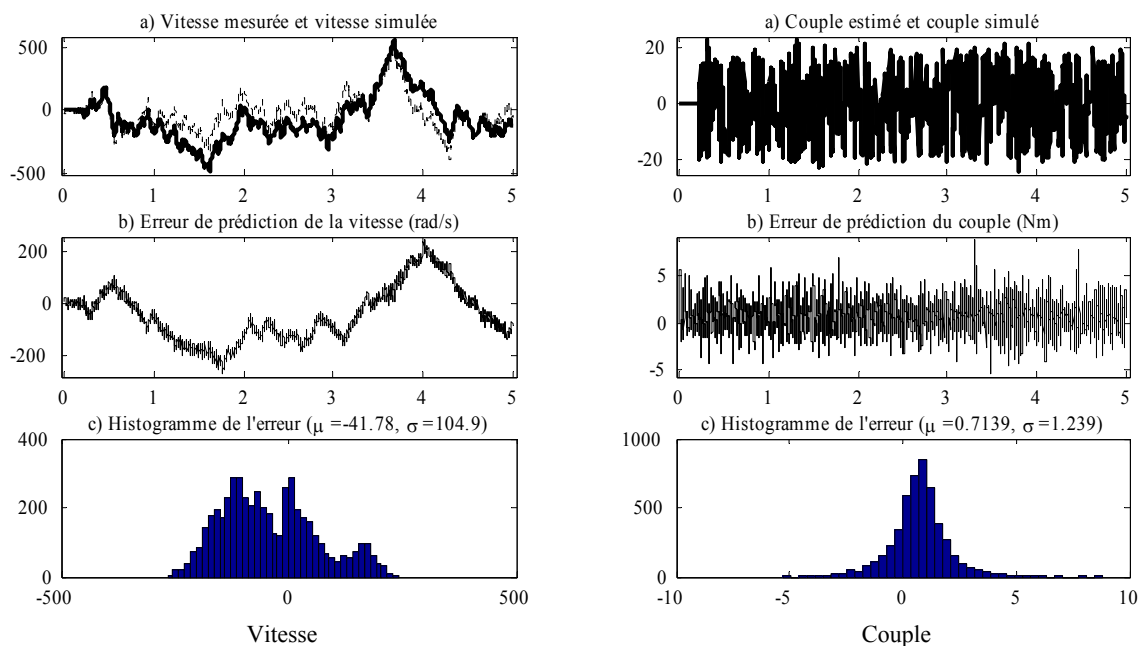


Figure 3.9: Comparaison de l'allure de la vitesse et du couple.

La figure 3.9 montre la vitesse et le couple « mesurés » expérimentalement et en simulation (couple estimé par le variateur à partir du courant dans l'axe q, et couple recalculé en simulation de la même manière) pour un couple de référence de type aléatoire. Nous pouvons constater que les réponses temporelles obtenues ont des allures semblables mais elles ne sont pas égales. Comme nous avons signalé auparavant, la faible précision des paramètres

mécaniques peut expliquer ces l'écart apparaissant dans la vitesse. Cependant, nous pouvons observer que les couples estimés sont très proches.

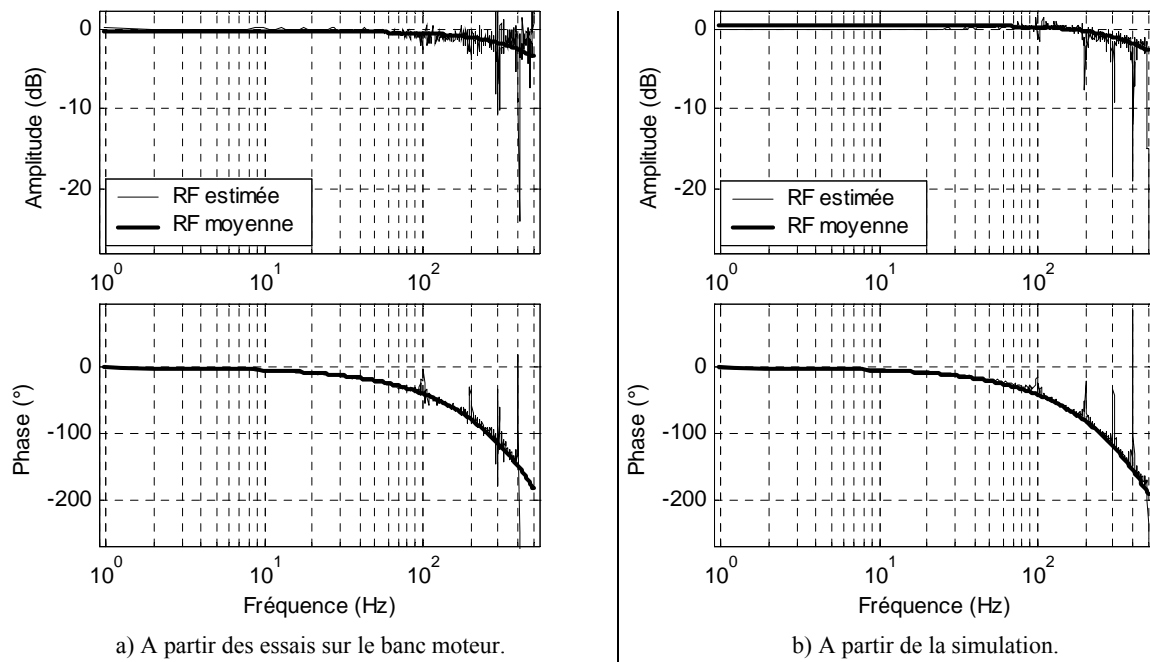


Figure 3.10: Réponses fréquentielles du boucle de courant.

La figure 3.10 montre les réponses fréquentielles (RF) des boucles de courant obtenues à partir des essais sur le banc moteur (figure 3.10a) et de la simulation (figure 3.10b). En d'autres termes, la réponse fréquentielle entre le couple de référence et le couple estimé. Avec les réponses fréquentielles de la boucle, la figure montre le modèle moyen estimé dans le logiciel MATLAB.

En regardant ces réponses, nous pouvons constater qu'elles sont très proches. Cependant, il faut faire attention ; la bande passante du correcteur du boucle de courant est de 500Hz minimum. La période d'échantillonnage (de 1kHz) ne nous permet pas d'obtenir des estimations au delà de 500Hz et introduit des phénomènes comme la chute d'amplitude et des retards dans la phase. Par conséquent, nous ne pouvons pas assurer que nous avons bien modélisé les phénomènes à haute fréquence (>500Hz).

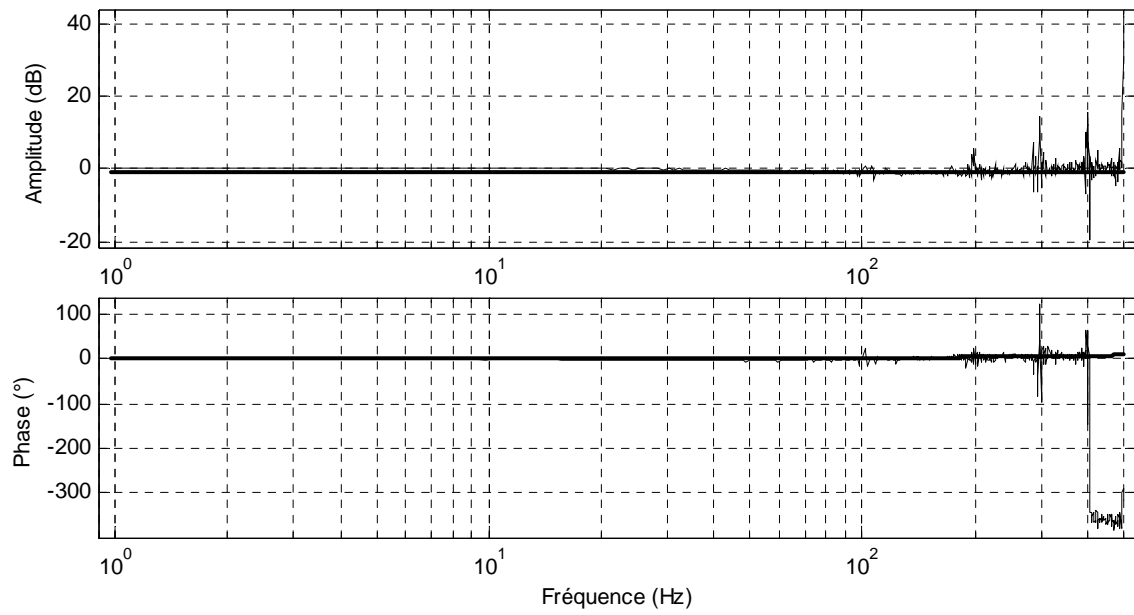


Figure 3.11: Différence entre les réponses fréquentielles montrés dans la figure 3.9.

La figure 3.11 confirme ce que nous avons mentionné au paragraphe précédent. Les paramètres modélisent bien le système sauf à proximité des 500Hz.

3.4 Mise en œuvre du banc d'essais avec la machine-outil.

3.4.1 Objectifs et description.

La figure 3.12 montre la fraiseuse que nous avons utilisée pour réaliser les expérimentations. Sur cette machine-outil, nous avons installé un accéléromètre sur chaque axe pour la phase d'obtention du modèle mécanique.



Figure 3.12: Machine-outil ANAYAK utilisé dans les essais expérimentaux.

La machine est dotée d'un CNC 8050 du constructeur FAGOR. Les axes X,Y et Z ont des moteurs synchrones FAGOR AXM 3.06.6 et les variateurs sont les mêmes que ceux utilisées pour l'autre banc d'essais (FAGOR AXD 2.50). La broche a un moteur asynchrone FAGOR SPM 112 LE et un variateur FAGOR SPD 2.50. Notre étude étant limitée aux axes, le moteur et le variateur de la broche n'ont aucune influence.

L'inclusion des accéléromètres sur les axes introduit des changements dans le diagramme d'interconnexion. La figure 3.13 montre ce diagramme.

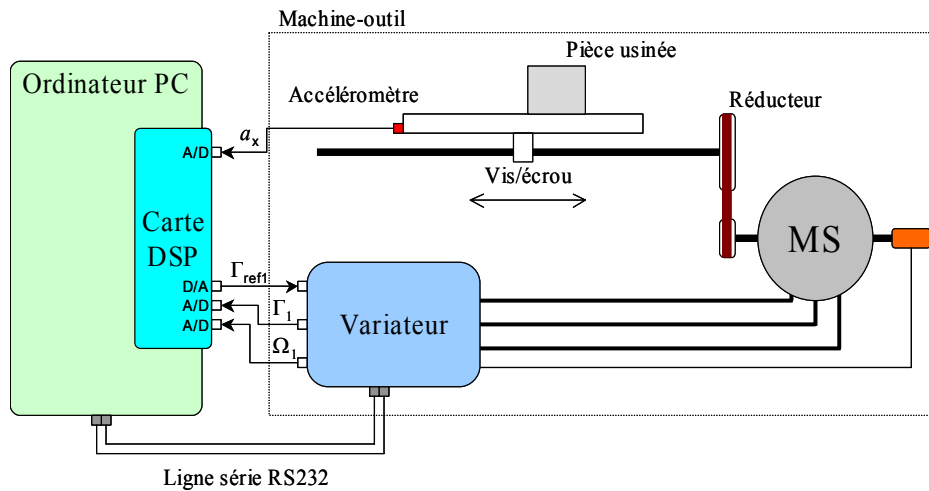


Figure 3.13: Diagramme d'interconnexion du banc d'essais de la machine-outil.

Il faut remarquer que pour utiliser la commande externe par le DSP, il faut tout d'abord désactiver les alarmes et la supervision du contrôle numérique. Le CN détecte les mouvements de l'axe et si le mouvement de l'axe dépasse une limite d'erreur de poursuite alors la puissance de la machine est coupée. Ceci ajoute une difficulté au moment de faire les essais sur la machine, car l'utilisateur *doit* faire attention pour que les axes ne dépassent pas les limites de son parcours sous peine d'endommager la machine (par exemple 695mm de parcours sur l'axe X).

3.4.2 Validation des modèles de simulation mécanique.

3.4.2.1 Paramètres mécaniques.

Nous avons utilisé une méthode d'identification dans le domaine fréquentiel pour retrouver le modèle mécanique que nous avons utilisé dans la simulation [PIN-2001]. Nous avons fourni un signal d'excitation SBPA ou aléatoire uniforme comme référence de couple et nous avons mesuré le couple estimé par le variateur et la vitesse de la machine.

Avec ces données, nous avons obtenu les spectres de puissance et le spectre croisé de puissance entre la vitesse et le couple :

$$\begin{aligned}
 \Psi_{T_q T_q}(\omega) &= \|T_q(\omega)\|^2 && \leftarrow \text{spectre de puissance du couple} \\
 \Psi_{\Omega \Omega}(\omega) &= \|\Omega(\omega)\|^2 && \leftarrow \text{spectre de puissance de la vitesse} \\
 \Psi_{\Omega T_q}(\omega) &= T_q^*(\omega)\Omega(\omega) && \leftarrow \text{spectre croisé de puissance entre vitesse et couple} \\
 &&& (T_q^*(\omega) \text{ représente le complexe conjugué de } T_q(\omega))
 \end{aligned}
 \tag{3.11}$$

Dans ces formules, $T_q(\omega)$ et $\Omega(\omega)$ représentent les spectres du couple et de la vitesse respectivement.

Pour minimiser les erreurs dues à la mesure, nous faisons une moyenne des spectres de puissance obtenus à différents instants de temps. Par exemple, dans les figures, les spectres sont obtenus avec 1024 échantillons et l'obtention d'un nouveau spectre est réalisée chaque 256 échantillons.

Notre but étant chercher la fonction de transfert entre la vitesse et le couple, nous pouvons l'obtenir en faisant la division entre le spectre croisé et le spectre du couple :

$$\begin{aligned} \Omega(\omega) &= F(\omega)T_q(\omega) \\ \frac{\Psi_{\Omega T_q}(\omega)}{\Psi_{\Omega\Omega}(\omega)} &= \frac{T_q^*(\omega)\Omega(\omega)}{\|T_q(\omega)\|^2} = \frac{T_q^*(\omega)F(\omega)T_q(\omega)}{\|T_q(\omega)\|^2} = F(\omega) \end{aligned} \quad (3.12)$$

La raison du calcul de la fonction de transfert avec les spectres de puissance au lieu de faire la division directe des spectres des signaux est que, d'une part, nous obtenons un meilleur filtrage des bruits et que, d'autre part, la fonction de cohérence :

$$C_{\Omega T_q}(\omega) = \frac{\|\Psi_{\Omega T_q}(\omega)\|^2}{\Psi_{T_q T_q}(\omega)\Psi_{\Omega\Omega}(\omega)} \leftarrow \text{fonction de cohérence} \quad (3.13)$$

nous indique le niveau des bruits ou des non linéarités présents au moment d'obtenir le spectre croisé. Dans un cas idéal sans bruits et avec une fonction de transfert linéaire, la valeur de la cohérence est 1.

Nous identifions ensuite la réponse fréquentielle avec celle d'une fonction de transfert du type suivant :

$$F(s) = e^{-T_d s} \frac{b_n s^n + \dots + b_1 s + b_0}{a_n s^n + \dots + a_1 s + a_0} \quad (3.14)$$

Pour obtenir les paramètres de cette fonction, nous avons utilisé une méthode de moindres carrés non linéaire du logiciel MATLAB implantée avec la fonction « lscurvefit » [OPT-1999].

Dans notre cas, la réponse fréquentielle de la mécanique présente une résonance et une antirésonance (voir figure 3.14). Par conséquent, le modèle que nous adoptons est le suivant :

$$F(s) = e^{-T_d s} \frac{1}{Js + f} \cdot \frac{\omega_r^2}{\omega_a^2} \cdot \frac{s^2 + 2\omega_a z_a s + \omega_a^2}{s^2 + 2\omega_r z_r s + \omega_r^2} \quad (3.15)$$

Où les paramètres sont :

J, f Inertie et frottement visqueux équivalents de la partie rigide.

ω_a, z_a Pulsation naturelle et amortissement de l'antirésonance.

ω_r, z_r Pulsation naturelle et amortissement de la résonance.

T_d Retard total du système.

La figure 3.14 montre la cohérence, la réponse fréquentielle estimée de la fonction de transfert que nous cherchons et la réponse fréquentielle moyenne obtenue à partir du modèle approché.

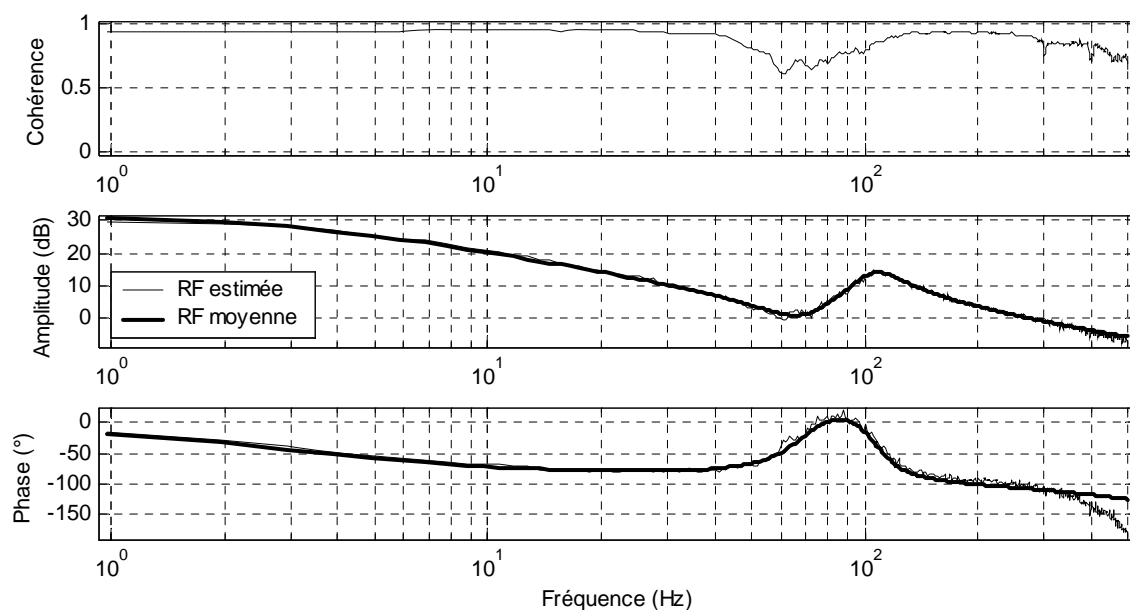


Figure 3.14: Réponses fréquentielles de l'axe X à partir des essais et modèle identifié.

Vous pouvez constater que la cohérence est aux alentours de 0,9. Cette valeur indique que même si la mesure est entachée de bruit, l'estimation du spectre croisé est bonne. Au niveau de l'antirésonance, vous pouvez observer que la cohérence diminue. Ceci n'est pas indicatif d'une non linéarité, mais l'amplitude ayant diminué et le niveau de bruit restant constant, l'influence relative du bruit augmente. Ce même phénomène se renouvelle aux hautes fréquences.

Paramètre	Valeur Max.	Valeur fig.	Valeur Min.
J (Kg.m ²)	$1,623 \times 10^{-3}$	$1,460 \times 10^{-3}$	$1,443 \times 10^{-3}$
F (Nm/rad/s)	$3,753 \times 10^{-2}$	$2,972 \times 10^{-2}$	$2,113 \times 10^{-2}$
ω_r (rad/s)	728,8	663,52	662
Z_r	0,1767	0,1593	0,09508
ω_a (rad/s)	483,8	433,16	424,1
Z_a	0,2256	0,2175	0,1339
T_d (μs)	134	130,7	116

Tableau 3.1: Paramètres du modèle mécanique de l'axe X et valeurs maximales et minimales.

Les valeurs des paramètres de l'équation (3.15) pour cet axe en milieu de course sont données sur le tableau 3.1 ainsi que les valeurs maximales et minimales que nous avons trouvées pour cet axe pour différentes positions et signaux d'excitation.

Les variations des fréquences de résonance et antirésonance ne sont pas grandes (<12% entre les valeurs maximale et minimale). Ces écarts peuvent être expliqués par la variabilité dans les conditions des essais (différents états de charge de la machine, différentes positions initiales) et avec l'incertitude de la méthode d'optimisation utilisée pour obtenir ces valeurs.

Les variations des amortissements sont grandes mais leur effet final dans la réponse fréquentielle est relativement réduit. Ces valeurs sont très sensibles au niveau de bruits présent dans la mesure et les raisons mentionnées ci-dessus peuvent aussi justifier ces variations.

Cependant, l'écart qui ne peut pas être expliqué par ces raisons est la variation du frottement visqueux (>50%). Dans le modèle choisi, nous ne tenons pas compte des frottements secs. Or les frottements secs du système introduisent une non linéarité à basses fréquences qui réduit effectivement le gain statique du système. Dans le modèle dont nous faisons l'approximation, le frottement visqueux est l'inverse du gain statique. Cependant, l'obtention de ce gain à partir des réponses fréquentielles est quasiment impossible parce que la valeur à fréquence 0 n'est pas calculée. Le seul moyen d'obtenir cette valeur serait en réalisant des essais à vitesse constante mais ils sont impossibles à réaliser à cause du parcours limité des axes.

Pour la mesure du frottement sec, nous avons fait varier lentement la valeur du couple de référence à partir de zéro jusqu'au mouvement de la machine. Cette méthode pose un certain nombre de problèmes connus mais c'était la méthode la plus simple que nous pouvions mettre en œuvre pendant le temps de disponibilité de la machine. Le problème qui nous affecte le plus est la surestimation de la valeur du frottement parce que la mesure a dû être faite pratiquement à l'arrêt pour les raisons de sécurité évoquées précédemment ; la valeur du frottement sec statique est plusieurs fois plus grande que sa valeur dynamique.

Les valeurs des retards sont présents uniquement pour tenir compte des petits retards introduits dans le variateur suite à l'acquisition de la vitesse et au calcul du couple estimé. Néanmoins, c'est un paramètre que nous pouvons ignorer pour l'obtention du modèle mécanique.

3.4.2.2 Paramètres du système d'acquisition des données.

Le retard entre le passage de données de la carte DSP au variateur et le retour des mesures affecte aussi notre modèle de simulation. La figure 3.15 montre la fonction de transfert estimée entre le couple de référence et le couple estimé dans le banc d'essais.

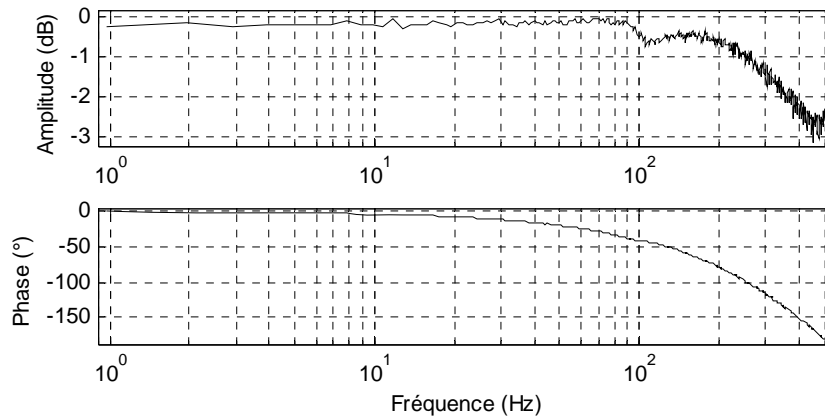


Figure 3.15: Réponse fréquentielle estimée de la fonction de transfert entre le couple de référence et le couple mesuré.

Nous observons deux phénomènes :

- Une chute du gain avec la fréquence qui correspond à la fonction de transfert en boucle fermée du dispositif du contrôle vectoriel des courants ; la bande passant à 3dB est au minimum de 500Hz si nous tenons compte de l'effet de l'échantillonnage qui accentue la chute du gain (fréquence d'échantillonnage 1kHz).
- Une chute de la phase, bien supérieure à celle associée à la chute du gain. Elle est révélatrice d'un retard pur que nous pouvons évaluer à 1,04ms.

La figure 3.16 montre la réponse fréquentielle de la fonction de transfert entre le couple de référence et le couple estimé, mais cette fois obtenue à partir des données de simulation. Nous pouvons constater que le retard est semblable dans les deux cas et que la chute de gain n'est pas tout à fait équivalente (dû probablement aux différences dans les paramètres du correcteur).

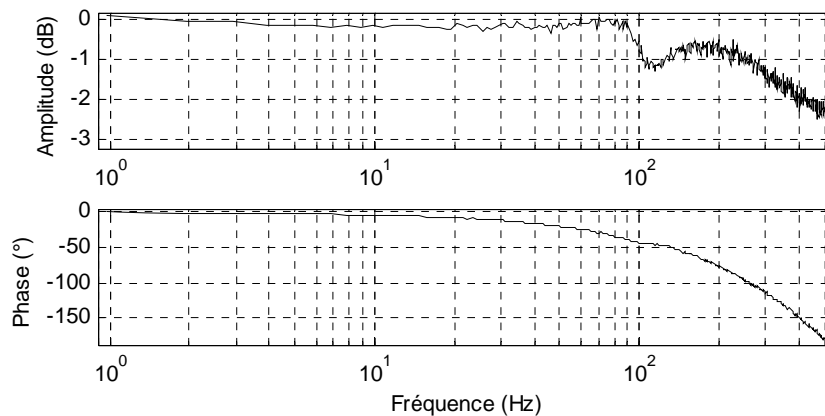


Figure 3.16: Réponse fréquentielle estimée à partir des données de simulation de la fonction de transfert entre le couple de référence et le couple estimé.

3.4.2.3 Comparaison des résultats expérimentaux et de simulation.

Nous avons fait une simulation de l'axe complet avec les différents modèles explicités dans les paragraphes précédents (modèle détaillé du moteur et du variateur tel que défini au §3.2.1.2 avec les paramètres donnés ou identifiés au §3.3.2.2 ; modèle de la mécanique de l'axe tel que défini au §3.4.2.1 avec les paramètres identifiés, axe mi course). Comme sur la machine réelle, nous fournissons à la simulation un signal d'excitation SBPA ou aléatoire uniforme comme référence de couple et nous relevons le couple estimé par le variateur et la vitesse de la machine.

La figure 3.17 montre les réponses fréquentielles estimés à partir des données expérimentales et de simulation.

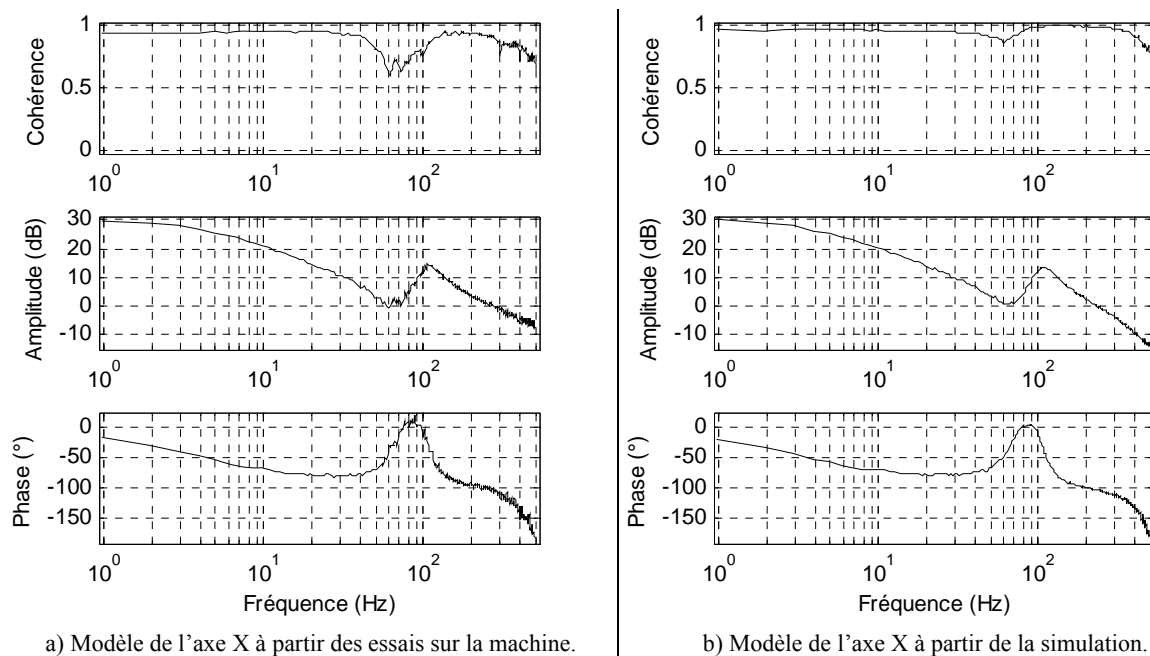


Figure 3.17: Réponses fréquentielles estimés à partir des données expérimentales et de simulation.

Nous pouvons constater que les deux réponses fréquentielles sont très proches. Si nous observons les paramètres des modèles obtenus avec ces réponses fréquentielles (voir tableau 3.2), nous pouvons constater qu'il existe des écarts assez importants entre les deux modèles.

L'écart sur l'inertie et le frottement visqueux existe principalement parce les frottements secs ne sont pas pris en compte dans la simulation et ces frottements introduisent des non linéarités à basses fréquences. D'autre part, la variabilité des valeurs des résonances et des amortissements provient principalement de l'algorithme d'optimisation qui peut engendrer des variations très importants (spécialement pour les amortissements).

Paramètre	Expérimentation (figure 3.17a)	Simulation (figure 3.17b)	Ecart (%)
J (Kg.m ²)	$1,481 \times 10^{-3}$	$1,793 \times 10^{-3}$	+21,1%
F (Nm/rad/s)	$2,822 \times 10^{-2}$	$2,433 \times 10^{-2}$	-13,8%
ω_r (rad/s)	666,02	690,57	+3,69%
Z_r	0,1518	0,1102	-27,3%
ω_a (rad/s)	436,42	478,81	+9,71%
Z_a	0,2237	0,3358	+50,1%
T_d (μ s)	197	144,2	-26,8%

Tableau 3.2: Comparaison de paramètres du modèle obtenus de l'expérimentation et la simulation.

En recalculant le tableau 3.2 avec une valeur approximative du frottement sec obtenu à partir des caractéristiques mécaniques de l'axe (masse des axes et coefficient de frottement sec), nous obtenons tableau 3.3 :

Paramètre	Expérimentation	Simulation	Ecart (%)
J (Kg.m ²)	$1,481 \times 10^{-3}$	$1,793 \times 10^{-3}$	+21,1%
F (Nm/rad/s)	$2,822 \times 10^{-2}$	$2,734 \times 10^{-2}$	-3,1%
ω_r (rad/s)	666,02	689,8	+3,57%
Z_r	0,1518	0,1105	-27,2%
ω_a (rad/s)	436,42	477,9	+9,51%
Z_a	0,2237	0,3343	+49,5%
T_d (μ s)	197	144,5	-26,7%

Tableau 3.3: Comparaison de paramètres du modèle obtenus de l'expérimentation et la simulation avec l'inclusion du frottement sec.

Nous pouvons constater que effectivement l'écart sur le frottement visqueux est plus faible.

3.5 Validation des algorithmes d'identification.

3.5.1 Identification et compensation du couple de crantage.

La figure 3.18 montre les résultats de compensation obtenus en simulation avec le réseau de neurones mentionné au §2.4.1b et le modèle détaillé du moteur et du variateur. Le réseau de neurones a concrètement une entrée, la position du rotor ; une sortie, le couple de crantage et 24 neurones dans la couche cachée. Vous pouvez observer que le fondamental du couple de crantage (arrivant 24 fois par tour de la machine) est réduit de 0,15Nm à 0,009Nm après la correction, compensant ainsi l'effet qu'il avait dans la vitesse.

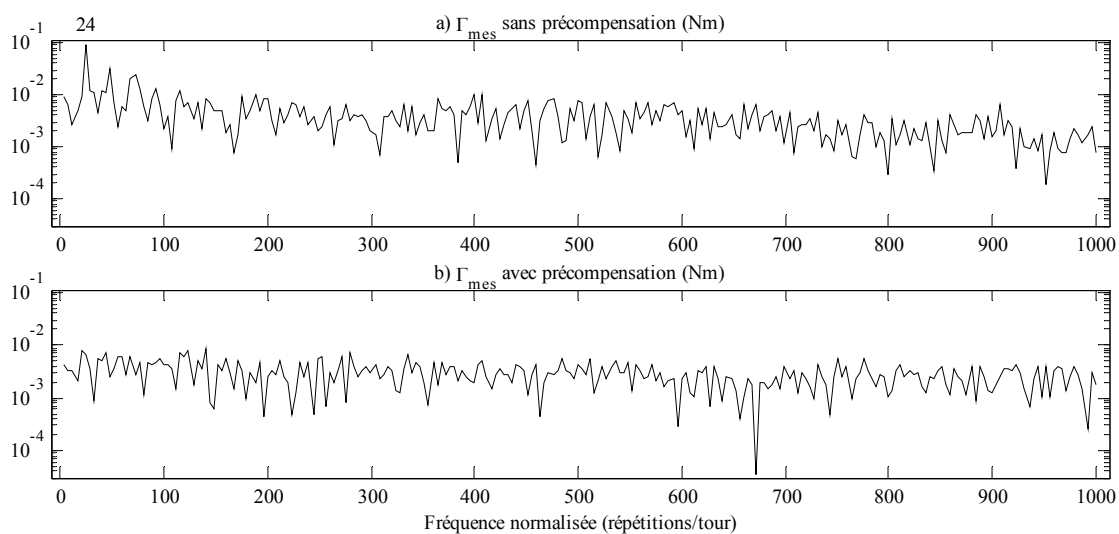


Figure 3.18: Harmoniques de couple avant la précompensation de couple (a) et après (b).

L'algorithme d'apprentissage converge progressivement vers l'allure de la perturbation du couple de crantage et réduit les harmoniques dus au couple de crantage jusqu'à l'annulation complète (en régime permanent).

En pratique, la méthode de précompensation pose un certain nombre de problèmes :

- Elle suppose le couple de crantage indépendant du régime de fonctionnement de la machine (vitesse, charge), ce qui n'est pas le cas en réalité.
- Pour chaque position, le réseau de neurones détermine le couple de crantage à ajouter au couple de référence. Cette correction est prise en compte par le variateur à la période d'échantillonnage suivante, donc pour une position différente de celle pour laquelle nous avons calculé la correction. Le phénomène est d'autant plus important que la vitesse est grande. Nous pouvons ainsi obtenir le résultat inverse de celui souhaité. On peut théoriquement minimiser ce problème en corrigeant la position pour laquelle on calcule la

correction en fonction de la vitesse de la machine et connaissant le retard d'application de la correction.

Expérimentalement, les résultats que nous avons obtenus sont très décevants. La principale explication réside dans les problèmes de mesure de position évoqués au §3.4.1.

3.5.2 Identification de la mécanique de l'axe.

En simulation, nous avons montré les performances des réseaux de neurones linéaires et non linéaires sur le modèle linéaire de la mécanique de l'axe §2.4.3.3. Compte-tenu du caractère non linéaire de l'axe réel, c'est le réseau non linéaire que nous avons mis en œuvre sur la machine.

La figure 3.19 montre les résultats expérimentaux obtenus.

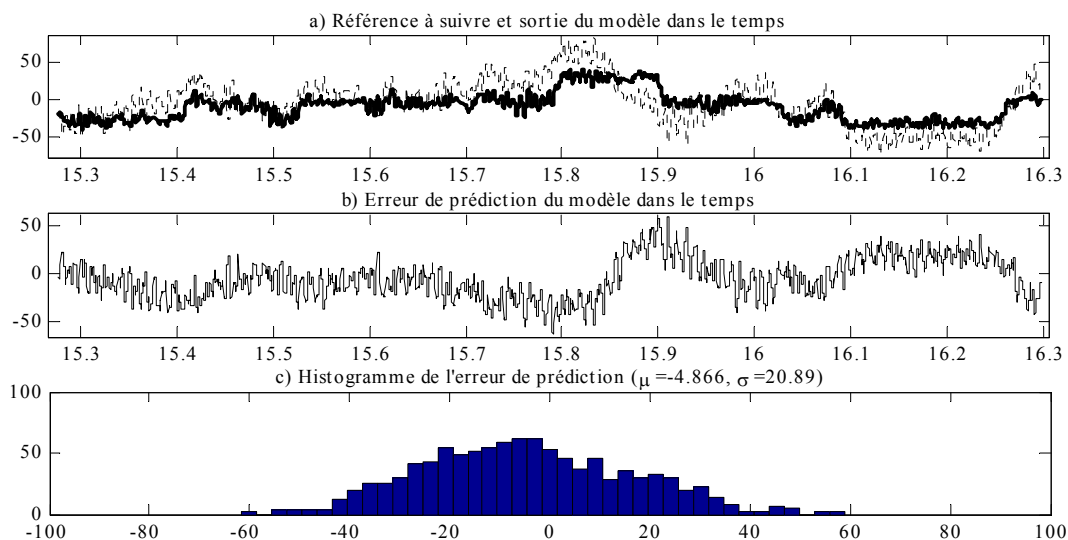


Figure 3.19: Validation du modèle identifié en prédiction: réponses temporelles.

La figure 3.20 montre les tests de corrélation des résultats de la figure 3.19.

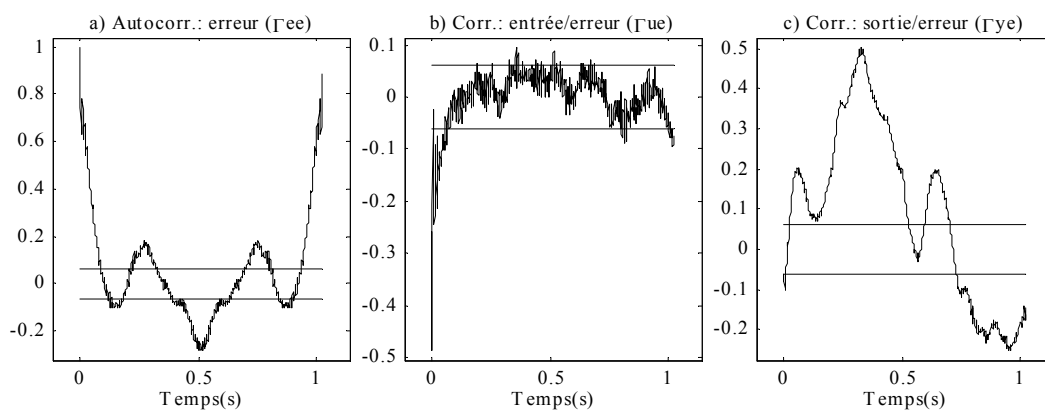


Figure 3.20: Validation du modèle identifié en prédiction: corrélations et marges de confiance.

Tel que nous pouvons observer dans les figures, nous sommes confrontés au problème du bruit de sortie coloré (voir §2.4.3.3d) ; par conséquent, nous retrouvons les conclusions que nous avons données :

- Les structures de type ARX et OE ne peuvent pas décorrélérer l'erreur de prédiction et l'excitation du système. Par conséquent, les paramètres obtenus sont biaisés.
- La structure ARMAX est indispensable si nous avons besoin de paramètres non-biaisés. Autrement, si le bruit coloré est réduit, alors les paramètres obtenus avec les structures ARX et OE peuvent être valables. Ceci n'est pas le cas pour une machine-outil.

3.5.3 Utilisation « en temps réel » des réseaux de neurones.

Pour évaluer l'utilisation en temps réel, nous avons évalué le nombre d'opérations élémentaires du processeur DSP pour le calcul du réseau de neurones et pour le calcul de l'algorithme d'apprentissage.

En ce qui concerne le calcul du réseau de neurones, nous pouvons décomposer les opérations réalisées à chaque couche de la façon suivante :

$$\begin{aligned}
 \text{Sommes et soustractions :} & \quad n_c \times n_{c-1} \\
 \text{Multiplications :} & \quad n_c \times n_{c-1} \\
 \text{Stockages en mémoire :} & \quad 2 \times n_c \\
 \text{Calculs de la fonction de sortie :} & \quad n_c
 \end{aligned} \tag{3.16}$$

n_c étant le nombre de neurones de la couche c et n_{c-1} étant le nombre de neurones de la couche précédant ($c-1$).

Habituellement, sur un DSP, la somme, la soustraction et le stockage en mémoire sont des opérations élémentaires car elles prennent le même temps. La division est approximativement équivalente à 4 opérations élémentaires. Exceptée la fonction de sortie linéaire, où nous n'avons aucune opération, le calcul de la fonction de sortie est réalisé en utilisant des tableaux d'interpolation qui peuvent nécessiter de 4 à 10 opérations élémentaires (en supposant qu'il est très optimisé).

Avec ces hypothèses, et prenant le pire des cas, le nombre d'opérations élémentaires pour calculer une couche d'un réseau de neurones est :

$$N_{op} = n_c(2n_{c-1} + 12) \tag{3.17}$$

Ainsi, pour un réseau de neurones avec une sortie linéaire, n_h neurones dans la couche cachée avec la fonction sigmoïde et n_u entrées le nombre d'opérations nécessaire pour calculer sa sortie est :

$$OP_{NN} \approx \underbrace{1 \times (2n_h + 1)}_{\text{couche de sortie}} + \underbrace{n_h (2n_u + 12)}_{\text{couche cachée}} = 1 + 14n_h + 2n_u n_h \quad (3.18)$$

En ce qui concerne l'algorithme d'apprentissage, il y a plusieurs calculs (voir la figure 2.11) :

$$\begin{aligned} \text{Erreur de prédiction } (e_j) : & \sim N_j \times (3 + 14n_h + 2n_u n_h) \\ \text{Fonction objectif } (sse_k) : & \sim 2N_j \\ \text{Jacobien } (J_j, \text{ sans récurrence}) : & \sim N_j \times 13n_h (n_u + 1) \\ \text{Jacobien } (J_j, \text{ avec récurrence}) : & \sim N_j \times [13n_h (n_u + 1) + 2N_p \times n_l] \\ \text{Gradient } (g_k) : & \sim N_j \times N_p \\ \text{Hessien } (H_k) : & \sim N_j \times N_p^2 \\ \text{Incrément du pas } (\Delta p_k) : & \sim (N_p^2 + 9N_p + 4)N_p / 3 \\ \text{Nouveau pas } (p_{k+1}) : & \sim 2 \times N_p \end{aligned} \quad (3.19)$$

N_j étant le nombre de jeux de données, $N_p = n_h (n_u + 2)$ étant le nombre de paramètres et n_l étant le nombre d'échantillons des instants précédents.

Le nombre d'opérations des boucles de l'algorithme est :

$$\begin{aligned} \text{Boucle interne } (\mu) : & \sim (\Delta p_k) + (p_{k+1}) + (e_j) + (sse_{k+1}) + 3 \\ \text{Boucle externe } (k) : & \sim 5 + N_p + (H_k) + Iter_\mu \times (\mu) + (J_j) + (g_k) \end{aligned} \quad (3.20)$$

$Iter_\mu$ étant le nombre d'itérations de la boucle interne.

Et finalement, le nombre d'opérations de l'algorithme est :

$$OP_{algorithme} \approx 2 + (e_j) + (sse_k) + (J_j) + (g_k) + Iter_k \times (k) + N_p \quad (3.21)$$

$Iter_k$ étant le nombre d'itérations de la boucle externe.

Nous pouvons constater que dans cette dernière équation les valeurs $Iter_k$ et $Iter_\mu$ dépendent de la convergence de l'algorithme et de la proximité de la solution recherchée. Par conséquent, leurs valeurs sont très variables. Pour trouver le nombre d'opérations réalisées par l'algorithme, nous devons prendre des hypothèses supplémentaires. Dans ce cas, nous avons considéré les deux extrêmes : la boucle externe réalise une seule itération ou le nombre

maximum d'itérations. Pour la boucle interne, nous avons supposé qu'en moyenne 2 itérations sont suffisantes pour qu'elle converge.

Ainsi, pour le réseau de neurones utilisé sur la machine pour l'identification, nous obtenons le tableau suivant ($n_h=25$, $n_u=7$, $N_j=1024$, $N_p=225$, $n_f=0$ ou 3) :

	Nombre d'opérations
Réseau de neurones	701
Algorithme d'apprentissage	
(sans récurrence, 1 itération)	$67,7 \times 10^6$
(sans récurrence, 50 itérations)	$3207,4 \times 10^6$
(avec récurrence, 1 itération)	$70,5 \times 10^6$
(avec récurrence, 50 itérations)	$3277,9 \times 10^6$

Tableau 3.4: Nombre d'opérations élémentaires dans le calcul du réseau de neurones et de l'algorithme d'apprentissage.

En supposant un processeur DSP qui exécute une opération élémentaire chaque 100ns, nous pouvons constater que le calcul du réseau de neurones peut être réalisé en 70 μ s. Cependant, il lui faut 6,7 secondes pour l'apprentissage dans le meilleur des cas et 5 minutes dans le pire des cas. Il faut dire que ce réseau est le plus grand que nous avons utilisé. Le réseau que nous avons utilisé normalement avait moins de neurones dans la couche cachée.

De toute façon, nous pouvons en tirer que même si le calcul du réseau de neurones est tout à fait possible dans un contexte « temps réel », son apprentissage ne l'est pas du tout. Par conséquent, l'algorithme d'apprentissage utilisé n'est pas du tout viable pour un système variant dans le temps rapidement.

3.6 Bilan.

L'apport des RN par rapport aux méthodes mettant à profit directement les connaissances sur le système est l'intégration de l'ensemble du noyau de calcul (le réseau proprement dit) et des méthodes d'adaptation des paramètres à l'environnement de travail (l'apprentissage). Cependant, les résultats obtenus nous montrent que le choix de la structure d'identification du réseau n'est pas adapté aux bruits présents dans une machine-outil. La présence de perturbations externes nous oblige à avoir une structure de type ARMAX.

La carte DSP utilisée peut calculer la sortie du réseau de neurones relativement complexe dans un temps relativement court et nous permet d'utiliser les réseaux de neurones en temps réel sans problèmes. Cependant, elle n'est pas assez puissante pour que l'apprentissage du réseau de neurones soit fait dans un horizon de temps suffisamment court. L'apprentissage différé a besoin de plusieurs centaines de millisecondes, voire de secondes, avant de donner un résultat.

En revenant à notre application, les principales non linéarités de notre banc d'essais proviennent :

1. Du couple de crantage.
2. Des frottements secs.

Et dans une moindre mesure :

3. Des variations de charge sur l'axe.

Sachant que les deux premières non linéarités dépendent principalement de la position et la vitesse, il est certainement plus aisé de mettre en œuvre un correcteur de précompensation de ces perturbations que de substituer le correcteur principal par un correcteur global qui prend en compte toutes les non linéarités (en plus d'assurer les performances de la boucle de commande).

Nous pouvons conclure qu'un correcteur global par réseau de neurones n'a d'intérêt que lorsque nous ne disposons d'aucune connaissance a priori sur le processus à contrôler. Par exemple, dans notre cas concret, nous aurions pu également substituer la précompensation de couple par réseau de neurones par une fonction sinusoïdale de la même forme que le modèle du couple de crantage en identifiant l'amplitude au démarrage et en ajustant la phase en ligne pour compenser l'effet du point de fonctionnement.

3.7 Bibliographie.

- [FLE-1987] R. FLETCHER
Practical Methods of optimization.
Wiley & Sons, 1987.
- [MAT-1998] MATLAB manuals.
Application Program Interface Guide. Version 5.
MathWorks. 1998
- [OPT-1999] T. COLEMAN, M.A. BRANCH, A. GRACE
MATLAB Optimization Toolbox User's Guide. Version 2 (release 11).
MathWorks. 1999
- [PIN-2001] R. PINTELON, J. SCHOUKENS
System Identification: A Frequency Domain Approach.
IEEE Press, Piscataway. 2001

Conclusions et perspectives.

Conclusions.

Les modèles mécaniques des axes de machine-outil utilisés généralement pour la synthèse des lois de commande, sont très simples ; ils supposent les axes rigides et ils ne prennent pas en compte les non linéarités telles que variations d'inertie, de frottements et a fortiori de rigidité mécanique avec la position de l'axe. Ces modèles sont insuffisants dans le contexte de l'usinage rapide pour lequel on recherche un suivi de trajectoire précis sur toute la course des axes, avec des vitesses et des accélérations importantes, sans générer de vibrations mécaniques. Les lois de commande doivent donc pouvoir être recalculées en ligne à partir de modèles non linéaires suffisamment précis, également identifiés en ligne pendant l'exploitation de la machine.

Pour atteindre les objectifs de vitesse et de qualité d'usinage, un travail important concerne également les structures et les matériaux de machines mais également la réduction des perturbations inhérentes à la structure elle-même (jeux, frottements..) et aux systèmes électrique d'entraînement des axes. Les moteurs électriques, leur alimentation et les dispositifs de contrôle des courants ont des imperfections qui génèrent de telles perturbations. Certaines peuvent être minimisées à la conception (couple d'encoche) ou facilement corrigées en exploitation (différentiels d'erreur de gain capteur), d'autres non (erreur de décalage capteur). Il faut donc chercher à compenser en ligne les perturbations résultantes après les avoir identifiées.

En théorie, les réseaux de neurones peuvent synthétiser et approcher n'importe quelle fonction non-linéaire. Cependant, il n'existe pas de méthodologie de choix de réseau (structure, nombre de couches et de neurones, type des neurones ...) à partir des caractéristiques de l'application et des contraintes. Le problème de l'apprentissage, en particulier de la richesse des jeux de données d'apprentissage, est également complexe. Un réseau n'est pas capable d'apprendre une propriété qui n'est pas contenue dans ces données.

Un certain nombre de connaissances a priori sur le système peuvent beaucoup simplifier le choix de réseau et la tâche d'apprentissage des réseaux de neurones (quel type de non linéarités, plage de variation des paramètres, répétitivité des comportements, ...). A contrario, si le système est suffisamment bien connu et modélisable (par des modèles quantitatifs ou

qualitatifs, déterministes ou stochastiques, linéaires ou non, ...), les réseaux de neurones, éventuellement plus lents, ne sont pas forcément compétitifs par rapport à des solutions plus classiques (identification, contrôle linéaire ou non linéaire, contrôle flou ...).

Dans ce travail, nous avons cherché à approfondir nos connaissances sur les imperfections de la chaîne électromécanique d'entraînement d'un axe de machine outil, afin de déterminer leur importance relative et les éventuelles difficultés dans l'identification des perturbations qu'elles génèrent, afin également de disposer d'un modèle assez précis de simulation. Nous avons exploré les possibilités que pouvaient offrir les réseaux de neurones dans l'identification et la compensation en ligne des perturbations préjudiciables à la qualité du contrôle d'axe. Nous avons exploré les possibilités théoriques des réseaux de neurones non linéaires et récurrents à l'identification en temps différé de fonctions non linéaires. Nous avons cherché à les exploiter pour l'identification de l'axe. Le choix de la structure du réseau de neurones et des fonctions d'activation des neurones était principalement dicté par le souci d'identifier des fonctions non linéaires dans un contexte de mise en œuvre en temps réel. Nous n'avons pas la prétention d'avoir fait le choix optimal, d'avoir exploré toutes les possibilités ni d'avoir fait progresser les connaissances théoriques sur les réseaux de neurones. Il s'agit davantage d'une « expérimentation » sur un cas concret peu étudié. Par conséquent, les résultats sont encore partiels et dans certains cas décevants sans que l'on puisse tirer de conclusion définitive ; on est encore loin de la mise en œuvre d'un système complet de commande d'axe par réseau de neurones incluant compensation de perturbation, identification en ligne de l'axe et génération de la référence de couple. Il n'est d'ailleurs pas évident que cet objectif soit réaliste.

Perspectives.

En ce qui concerne les perturbations, pour pouvoir réellement valider ou non le principe de la compensation, il est important de réaliser les expérimentations dans de meilleures conditions au niveau des mesures de position et de vitesse, des périodes d'échantillonnage et du réglage du correcteur de vitesse de façon à avoir une bonne image du couple de perturbation sur une plage de fréquence plus significative.

Outre une exploitation plus complète des résultats expérimentaux, avec des conditions d'essais améliorées, les perspectives par rapport aux réseaux de neurones se situent dans trois grands axes :

- L'étude d'autres types de structures d'identification comme la structure ARMAX, qui devrait donner de meilleurs résultats d'identification en présence des perturbations qui affectent réellement l'axe d'une machine outils.
- L'exploration de méthodes d'apprentissage en ligne, indispensables pour l'identification des systèmes évoluant rapidement dans le temps. Les méthodes que nous avons mises en œuvre ne permettent que le traitement différé. Parmi les méthodes adaptatives qui font actuellement l'objet de nombreux travaux, le filtrage par les variantes du filtre de Kalman étendu présente un intérêt particulier. Les propriétés de ces méthodes adaptatives nous assurent une convergence des paramètres dans le cas où les bruits intervenant sur le système sont non-gaussiens. Les réseaux de neurones à base radiale, qui ont la propriété de converger rapidement, peuvent être opportunément associés à ces méthodes.
- L'étude de la faisabilité et de l'intérêt du contrôle par réseaux de neurones pour des procédés variant dans le temps.

Annexe A: Etude des harmoniques du couple de la machine synchrone.

A.1 Définitions de base.

(Référence depuis chapitre 2).

Voici quelques grandeurs auxiliaires et propriétés que nous utiliserons par la suite dans les transformations complexes des équations de la machine :

$$a = e^{j\frac{2\pi}{3}} \quad b = e^{j\delta} \quad a^{3n} = a^{-3n} = 1 \quad 1 + a + a^2 = 0 \quad a^* = a^{-1} = a^2 \quad (\text{A.1})$$

Ainsi, le flux créé par les aimants permanent du rotor dans sa forme complexe s'écrit :

$$\mathbf{K}_s(\theta) = \begin{bmatrix} k_a(\theta) \\ k_b(\theta) \\ k_c(\theta) \end{bmatrix} = \sum_{\substack{k=1 \\ n=2k-1}} K_n \begin{bmatrix} \cos(n\theta) \\ \cos[n(\theta - 2\pi/3)] \\ \cos[n(\theta - 4\pi/3)] \end{bmatrix} = \sum_{\substack{k=1 \\ n=2k-1}} \frac{K_n}{2} \begin{bmatrix} e^{jn\theta} + e^{-jn\theta} \\ e^{jn\theta} a^{-n} + e^{-jn\theta} a^n \\ e^{jn\theta} a^{-2n} + e^{-jn\theta} a^{2n} \end{bmatrix} \quad (\text{A.2})$$

et sa dérivée :

$$\frac{d\mathbf{K}_s(\theta)}{d\theta} = - \sum_{\substack{k=1 \\ n=2k-1}} n \frac{K_n}{j2} \begin{bmatrix} e^{jn\theta} - e^{-jn\theta} \\ e^{jn\theta} a^{-n} - e^{-jn\theta} a^n \\ e^{jn\theta} a^{-2n} - e^{-jn\theta} a^{2n} \end{bmatrix} \quad (\text{A.3})$$

La matrice des inductances statoriques et sa dérivée :

$$\mathbf{L}_s(\theta) = \begin{bmatrix} L_{s0} & M_{s0} & M_{s0} \\ M_{s0} & L_{s0} & M_{s0} \\ M_{s0} & M_{s0} & L_{s0} \end{bmatrix} + \sum_{\substack{k=1 \\ n=2k}} L_n \begin{bmatrix} \cos(n\theta) & \cos[n(\theta + 2\pi/3)] & \cos[n(\theta - 2\pi/3)] \\ \cos[n(\theta + 2\pi/3)] & \cos[n(\theta - 2\pi/3)] & \cos(n\theta) \\ \cos[n(\theta - 2\pi/3)] & \cos(n\theta) & \cos[n(\theta + 2\pi/3)] \end{bmatrix} \quad (\text{A.4})$$

$$\mathbf{L}_s(\theta) = \mathbf{L}_{s0} + \sum_{\substack{k=1 \\ n=2k}} \frac{L_n}{2} \begin{bmatrix} e^{jn\theta} + e^{-jn\theta} & e^{jn\theta} a^n + e^{-jn\theta} a^{-n} & e^{jn\theta} a^{-n} + e^{-jn\theta} a^n \\ e^{jn\theta} a^n + e^{-jn\theta} a^{-n} & e^{jn\theta} a^{-n} + e^{-jn\theta} a^n & e^{jn\theta} + e^{-jn\theta} \\ e^{jn\theta} a^{-n} + e^{-jn\theta} a^n & e^{jn\theta} + e^{-jn\theta} & e^{jn\theta} a^n + e^{-jn\theta} a^{-n} \end{bmatrix}$$

$$\frac{d\mathbf{L}_s(\theta)}{d\theta} = - \sum_{\substack{k=1 \\ n=2k}} n \frac{L_n}{j2} \begin{bmatrix} e^{jn\theta} - e^{-jn\theta} & e^{jn\theta} a^n - e^{-jn\theta} a^{-n} & e^{jn\theta} a^{-n} - e^{-jn\theta} a^n \\ e^{jn\theta} a^n - e^{-jn\theta} a^{-n} & e^{jn\theta} a^{-n} - e^{-jn\theta} a^n & e^{jn\theta} - e^{-jn\theta} \\ e^{jn\theta} a^{-n} - e^{-jn\theta} a^n & e^{jn\theta} - e^{-jn\theta} & e^{jn\theta} a^n - e^{-jn\theta} a^{-n} \end{bmatrix} \quad (\text{A.5})$$

A.2 Harmoniques de couple avec un courant non sinusoïdal.

Nous allons considérer le développement en série de Fourier du courant dans sa forme complexe en tenant compte de ses possibles harmoniques :

$$\mathbf{i}_s = \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} = \sum_{z=1} i_z \begin{bmatrix} \sin[z\theta + \delta_z] \\ \sin[z(\theta - 2\pi/3) + \delta_z] \\ \sin[z(\theta - 4\pi/3) + \delta_z] \end{bmatrix} = \sum_{z=1} \frac{i_z}{j2} \begin{bmatrix} e^{j(z\theta + \delta_z)} - e^{-j(z\theta + \delta_z)} \\ e^{j(z\theta + \delta_z)} a^{-z} - e^{-j(z\theta + \delta_z)} a^z \\ e^{j(z\theta + \delta_z)} a^{-2z} - e^{-j(z\theta + \delta_z)} a^{2z} \end{bmatrix} \quad (\text{A.6})$$

$$i_{3o} = 0, \quad \forall o \in N^+$$

Les trois courants de phase étant équilibrés alors les harmoniques multiples de 3 du fondamental du courant sont nuls (facilement vérifiable en faisant la somme des trois courants).

Le couple total de la machine synchrone est donné par l'équation suivante :

$$\Gamma = p \left(\frac{1}{2} \mathbf{i}_s^T \frac{d\mathbf{L}_s(\theta)}{d\theta} \mathbf{i}_s + \mathbf{i}_s^T \frac{d\mathbf{K}_s(\theta)}{d\theta} - \frac{dW_{rotor}(\theta)}{d\theta} \right) \quad (\text{A.7})$$

Nous allons séparer les différents termes de cette équation pour simplifier le développement et nous substituons le courant donné par l'équation (A.6).

Ainsi le couple créé par l'interaction avec les aimants rotoriques avec un courant non sinusoïdal dévient :

$$\begin{aligned} \Gamma_{flux} &= \mathbf{i}_s^T \frac{d\mathbf{K}_s(\theta)}{d\theta} p = -p \mathbf{i}_s^T \sum_{\substack{k=1 \\ n=2k-1}} n \frac{K_n}{j2} \begin{bmatrix} e^{jn\theta} - e^{-jn\theta} \\ e^{jn\theta} a^{-n} - e^{-jn\theta} a^n \\ e^{jn\theta} a^{-2n} - e^{-jn\theta} a^{2n} \end{bmatrix} \\ \Gamma_{flux} &= p \sum_{\substack{k=1 \\ n=2k-1}} \sum_{z=1} n \frac{i_z K_n}{4} \begin{bmatrix} e^{j[(z+n)\theta + \delta_z]} + e^{-j[(z+n)\theta + \delta_z]} - e^{j[(z-n)\theta + \delta_z]} - e^{-j[(z-n)\theta + \delta_z]} \\ + e^{j[(z+n)\theta + \delta_z]} a^{-(z+n)} + e^{-j[(z+n)\theta + \delta_z]} a^{(z+n)} \\ - e^{j[(z-n)\theta + \delta_z]} a^{-(z-n)} - e^{-j[(z-n)\theta + \delta_z]} a^{(z-n)} \\ + e^{j[(z+n)\theta + \delta_z]} a^{-2(z+n)} + e^{-j[(z+n)\theta + \delta_z]} a^{2(z+n)} \\ - e^{j[(z-n)\theta + \delta_z]} a^{-2(z-n)} - e^{-j[(z-n)\theta + \delta_z]} a^{2(z-n)} \end{bmatrix} = \\ \Gamma_{flux} &= p \sum_{\substack{k=1 \\ n=2k-1}} \sum_{z=1} n \frac{i_z K_n}{4} \left[\begin{aligned} & \left(e^{j[(z+n)\theta + \delta_z]} + e^{-j[(z+n)\theta + \delta_z]} \right) \left(1 + a^{(z+n)} + a^{2(z+n)} \right) - \\ & \left(e^{j[(z-n)\theta + \delta_z]} + e^{-j[(z-n)\theta + \delta_z]} \right) \left(1 + a^{(z-n)} + a^{2(z-n)} \right) \end{aligned} \right] = \\ \Gamma_{flux} &= p \sum_{\substack{k=1 \\ n=2k-1}} \sum_{z=1} n \frac{i_z K_n}{2} \left[\begin{aligned} & \left(1 + a^{(z+n)} + a^{2(z+n)} \right) \cos[(z+n)\theta + \delta_z] - \\ & \left(1 + a^{(z-n)} + a^{2(z-n)} \right) \cos[(z-n)\theta + \delta_z] \end{aligned} \right] = \end{aligned}$$

Il faut remarquer que, suite à la longueur des équations, nous avons été obligés de les écrire en plusieurs lignes. Ainsi les termes entre crochets peuvent être soit des vecteurs soit des valeurs scalaires. Par la suite, nous considérerons que s'il n'y a aucun signe de somme ou soustraction entre deux termes en deux lignes différents, alors ces deux termes font partie de deux éléments différents d'un vecteur.

Pour pouvoir simplifier cette dernière équation, nous devons développer les termes en puissances de a en multiples de 3 pour pouvoir utiliser les formules simplificatrices données par l'équation (A.1). Cette somme en k et z peut être séparée en 9 sommes différentes avec le développement suivant en faisant un changement de variable :

$$\begin{aligned} \sum_{n=1} \sum_{z=1} f(n, z) &= \sum_{k=1} \sum_{m=1} \\ & f(3k-1, 3m-1) + f(3k-2, 3m-1) + f(3k-3, 3m-1) + \\ & f(3k-1, 3m-2) + f(3k-2, 3m-2) + f(3k-3, 3m-2) + \\ & f(3k-1, 3m-3) + f(3k-2, 3m-3) + f(3k-3, 3m-3) \end{aligned} \quad (\text{A.9})$$

En appliquant cette dernière équation dans l'équation (A.8), nous obtenons :

$$\begin{aligned} \Gamma_{flux} &= p \sum_{k=1} \sum_{m=1} \\ & (6k-5) \frac{i_{(3m-2)} k_{(6k-5)}}{2} \left[\frac{(1 + a^{(3m-2+6k-5)} + a^{2(3m-2+6k-5)}) \cos[(3m-2+6k-5)\theta + \delta_{(3m-2)}] -}{(1 + a^{(3m-2-6k+5)} + a^{2(3m-2-6k+5)}) \cos[(3m-2-6k+5)\theta + \delta_{(3m-2)}]} \right] + \\ & (6k-5) \frac{i_{(3m-1)} k_{(6k-5)}}{2} \left[\frac{(1 + a^{(3m-1+6k-5)} + a^{2(3m-1+6k-5)}) \cos[(3m-1+6k-5)\theta + \delta_{(3m-1)}] -}{(1 + a^{(3m-1-6k+5)} + a^{2(3m-1-6k+5)}) \cos[(3m-1-6k+5)\theta + \delta_{(3m-1)}]} \right] + \\ & (6k-5) \frac{i_{3m} k_{(6k-5)}}{2} \left[\frac{(1 + a^{(3m+6k-5)} + a^{2(3m+6k-5)}) \cos[(3m+6k-5)\theta + \delta_{(3m)}] -}{(1 + a^{(3m-6k+5)} + a^{2(3m-6k+5)}) \cos[(3m-6k+5)\theta + \delta_{(3m)}]} \right] + \\ & (6k-3) \frac{i_{(3m-2)} k_{(6k-3)}}{2} \left[\frac{(1 + a^{(3m-2+6k-3)} + a^{2(3m-2+6k-3)}) \cos[(3m-2+6k-3)\theta + \delta_{(3m-2)}] -}{(1 + a^{(3m-2-6k+3)} + a^{2(3m-2-6k+3)}) \cos[(3m-2-6k+3)\theta + \delta_{(3m-2)}]} \right] + \\ & (6k-3) \frac{i_{(3m-1)} k_{(6k-3)}}{2} \left[\frac{(1 + a^{(3m-1+6k-3)} + a^{2(3m-1+6k-3)}) \cos[(3m-1+6k-3)\theta + \delta_{(3m-1)}] -}{(1 + a^{(3m-1-6k+3)} + a^{2(3m-1-6k+3)}) \cos[(3m-1-6k+3)\theta + \delta_{(3m-1)}]} \right] + \\ & (6k-3) \frac{i_{3m} k_{(6k-3)}}{2} \left[\frac{(1 + a^{(3m+6k-3)} + a^{2(3m+6k-3)}) \cos[(3m+6k-3)\theta + \delta_{(3m)}] -}{(1 + a^{(3m-6k+3)} + a^{2(3m-6k+3)}) \cos[(3m-6k+3)\theta + \delta_{(3m)}]} \right] + \\ & (6k-1) \frac{i_{(3m-2)} k_{(6k-1)}}{2} \left[\frac{(1 + a^{(3m-2+6k-1)} + a^{2(3m-2+6k-1)}) \cos[(3m-2+6k-1)\theta + \delta_{(3m-2)}] -}{(1 + a^{(3m-2-6k+1)} + a^{2(3m-2-6k+1)}) \cos[(3m-2-6k+1)\theta + \delta_{(3m-2)}]} \right] + \\ & (6k-1) \frac{i_{(3m-1)} k_{(6k-1)}}{2} \left[\frac{(1 + a^{(3m-1+6k-1)} + a^{2(3m-1+6k-1)}) \cos[(3m-1+6k-1)\theta + \delta_{(3m-1)}] -}{(1 + a^{(3m-1-6k+1)} + a^{2(3m-1-6k+1)}) \cos[(3m-1-6k+1)\theta + \delta_{(3m-1)}]} \right] + \\ & (6k-1) \frac{i_{3m} k_{(6k-1)}}{2} \left[\frac{(1 + a^{(3m+6k-1)} + a^{2(3m+6k-1)}) \cos[(3m+6k-1)\theta + \delta_{(3m)}] -}{(1 + a^{(3m-6k+1)} + a^{2(3m-6k+1)}) \cos[(3m-6k+1)\theta + \delta_{(3m)}]} \right] = \end{aligned} \quad (\text{A.10})$$

En appliquant les équations simplificatrices (A.1) :

$$\begin{aligned}
\Gamma_{flux} &= p \sum_{k=1} \sum_{m=1} \\
& (6k-5) \frac{i_{(3m-2)} k_{(6k-5)}}{2} \left[\frac{(1+a^{(-7)}+a^{2(-7)}) \cos[(3m+6k-7)\theta + \delta_{(3m-2)}]}{(1+a^{(3)}+a^{2(3)}) \cos[(3m-6k+3)\theta + \delta_{(3m-2)}]} \right] + \\
& (6k-5) \frac{i_{(3m-1)} k_{(6k-5)}}{2} \left[\frac{(1+a^{(-6)}+a^{2(-6)}) \cos[(3m+6k-6)\theta + \delta_{(3m-1)}]}{(1+a^{(4)}+a^{2(4)}) \cos[(3m-6k+4)\theta + \delta_{(3m-1)}]} \right] + \\
& (6k-5) \frac{i_{3m} k_{(6k-5)}}{2} \left[\frac{(1+a^{(-5)}+a^{2(-5)}) \cos[(3m+6k-5)\theta + \delta_{(3m)}]}{(1+a^{(5)}+a^{2(5)}) \cos[(3m-6k+5)\theta + \delta_{(3m)}]} \right] + \\
& (6k-3) \frac{i_{(3m-2)} k_{(6k-3)}}{2} \left[\frac{(1+a^{(-5)}+a^{2(-5)}) \cos[(3m+6k-5)\theta + \delta_{(3m-2)}]}{(1+a^{(1)}+a^{2(1)}) \cos[(3m-6k+1)\theta + \delta_{(3m-2)}]} \right] + \\
& (6k-3) \frac{i_{(3m-1)} k_{(6k-3)}}{2} \left[\frac{(1+a^{(-4)}+a^{2(-4)}) \cos[(3m+6k-4)\theta + \delta_{(3m-1)}]}{(1+a^{(2)}+a^{2(2)}) \cos[(3m-6k+2)\theta + \delta_{(3m-1)}]} \right] + \\
& (6k-3) \frac{i_{3m} k_{(6k-3)}}{2} \left[\frac{(1+a^{(-3)}+a^{2(-3)}) \cos[(3m+6k-3)\theta + \delta_{(3m)}]}{(1+a^{(3)}+a^{2(3)}) \cos[(3m-6k+3)\theta + \delta_{(3m)}]} \right] + \\
& (6k-1) \frac{i_{(3m-2)} k_{(6k-1)}}{2} \left[\frac{(1+a^{(-3)}+a^{2(-3)}) \cos[(3m+6k-3)\theta + \delta_{(3m-2)}]}{(1+a^{(-1)}+a^{2(-1)}) \cos[(3m-6k-1)\theta + \delta_{(3m-2)}]} \right] + \\
& (6k-1) \frac{i_{(3m-1)} k_{(6k-1)}}{2} \left[\frac{(1+a^{(-2)}+a^{2(-2)}) \cos[(3m+6k-2)\theta + \delta_{(3m-1)}]}{(1+a^{(0)}+a^{2(0)}) \cos[(3m-6k)\theta + \delta_{(3m-1)}]} \right] + \\
& (6k-1) \frac{i_{3m} k_{(6k-1)}}{2} \left[\frac{(1+a^{(-1)}+a^{2(-1)}) \cos[(3m+6k-1)\theta + \delta_{(3m)}]}{(1+a^{(1)}+a^{2(1)}) \cos[(3m-6k+1)\theta + \delta_{(3m)}]} \right] = \\
\Gamma_{flux} &= \frac{3}{2} p \sum_{k=1} \sum_{m=1} \left[\begin{aligned} & -(6k-5) \cdot k_{(6k-5)} i_{(3m-2)} \cos[(3m-6k+3)\theta + \delta_{(3m-2)}] + \\ & (6k-5) \cdot k_{(6k-5)} i_{(3m-1)} \cos[(3m+6k-6)\theta + \delta_{(3m-1)}] + \\ & (6k-3) \cdot k_{(6k-3)} i_{3m} \cos[(3m+6k-3)\theta + \delta_{(3m)}] - \\ & (6k-3) \cdot k_{(6k-3)} i_{3m} \cos[(3m-6k+3)\theta + \delta_{(3m)}] + \\ & (6k-1) \cdot k_{(6k-1)} i_{(3m-2)} \cos[(3m+6k-3)\theta + \delta_{(3m-2)}] - \\ & (6k-1) \cdot k_{(6k-1)} i_{(3m-1)} \cos[(3m-6k)\theta + \delta_{(3m-1)}] \end{aligned} \right] \quad (A.12)
\end{aligned}$$

Cette dernière équation peut encore être simplifiée davantage en sachant que les harmoniques multiples de 3 du courant sont nuls :

$$i_{3m} = 0, \quad \forall m \in N^+$$

$$\Gamma_{flux} = \frac{3}{2} p \sum_{k=1} \sum_{m=1} \begin{bmatrix} -(6k-5) \cdot k_{(6k-5)} i_{(3m-2)} \cos[(3m-6k+3)\theta + \delta_{(3m-2)}] + \\ (6k-5) \cdot k_{(6k-5)} i_{(3m-1)} \cos[(3m+6k-6)\theta + \delta_{(3m-1)}] + \\ (6k-1) \cdot k_{(6k-1)} i_{(3m-2)} \cos[(3m+6k-3)\theta + \delta_{(3m-2)}] - \\ (6k-1) \cdot k_{(6k-1)} i_{(3m-1)} \cos[(3m-6k)\theta + \delta_{(3m-1)}] \end{bmatrix} \quad (\text{A.13})$$

Le courant n'ayant que des harmoniques impairs généralement, il est intéressant de séparer les harmoniques pairs et impairs. Pour cela, nous avons utilisé une variante du développement comme il est montré dans l'équation (A.9) :

$$\Gamma_{flux} = \frac{3}{2} p \sum_{k=1} \sum_{r=1} \begin{bmatrix} -(6k-5) \cdot k_{(6k-5)} i_{(6r-5)} \cos[(6r-6k)\theta + \delta_{(6r-5)}] \\ + (6k-5) \cdot k_{(6k-5)} i_{(6r-4)} \cos[(6r+6k-9)\theta + \delta_{(6r-4)}] \\ + (6k-1) \cdot k_{(6k-1)} i_{(6r-5)} \cos[(6r+6k-6)\theta + \delta_{(6r-5)}] \\ - (6k-1) \cdot k_{(6k-1)} i_{(6r-4)} \cos[(6r-6k-3)\theta + \delta_{(6r-4)}] \\ - (6k-5) \cdot k_{(6k-5)} i_{(6r-2)} \cos[(6r-6k+3)\theta + \delta_{(6r-2)}] \\ + (6k-5) \cdot k_{(6k-5)} i_{(6r-1)} \cos[(6r+6k-6)\theta + \delta_{(6r-1)}] \\ + (6k-1) \cdot k_{(6k-1)} i_{(6r-2)} \cos[(6r+6k-3)\theta + \delta_{(6r-2)}] \\ - (6k-1) \cdot k_{(6k-1)} i_{(6r-1)} \cos[(6r-6k)\theta + \delta_{(6r-1)}] \end{bmatrix} \quad (\text{A.14})$$

$$\Gamma_{flux} = \frac{3}{2} p \sum_{k=1} \sum_{r=1} \begin{bmatrix} -(6k-5) \cdot k_{(6k-5)} i_{(6r-5)} \cos[(6r-6k)\theta + \delta_{(6r-5)}] \\ + (6k-1) \cdot k_{(6k-1)} i_{(6r-5)} \cos[(6r+6k-6)\theta + \delta_{(6r-5)}] \\ + (6k-5) \cdot k_{(6k-5)} i_{(6r-1)} \cos[(6r+6k-6)\theta + \delta_{(6r-1)}] \\ - (6k-1) \cdot k_{(6k-1)} i_{(6r-1)} \cos[(6r-6k)\theta + \delta_{(6r-1)}] \\ + (6k-5) \cdot k_{(6k-5)} i_{(6r-4)} \cos[(6r+6k-9)\theta + \delta_{(6r-4)}] \\ - (6k-1) \cdot k_{(6k-1)} i_{(6r-4)} \cos[(6r-6k-3)\theta + \delta_{(6r-4)}] \\ - (6k-5) \cdot k_{(6k-5)} i_{(6r-2)} \cos[(6r-6k+3)\theta + \delta_{(6r-2)}] \\ + (6k-1) \cdot k_{(6k-1)} i_{(6r-2)} \cos[(6r+6k-3)\theta + \delta_{(6r-2)}] \end{bmatrix} \quad (\text{A.15})$$

Cette dernière équation représente les harmoniques de couple générés par l'interaction du flux rotorique avec un courant non sinusoïdal. Nous pouvons constater que le couple n'a que des harmoniques multiples de 3. Cependant, si le courant n'a que des harmoniques impairs alors le couple n'a que des harmoniques multiples de 6.

Pour trouver les harmoniques de couple générés dues par les variations de réluctance dans le circuit magnétique, nous suivons exactement la même démarche :

$$\Gamma_{réel} = p \frac{1}{2} \mathbf{i}_s^T \frac{d\mathbf{L}_s(\theta)}{d\theta} \mathbf{i}_s \quad (\text{A.16})$$

$$\Gamma_{réel} = p \frac{1}{8} \mathbf{i}_s^T \sum_{k=1} \sum_{\substack{z=1 \\ n=2k}} n L_n i_z \left[\begin{array}{l} e^{j[(z+n)\theta+\delta_z]} (1 + a^{-(z-n)} + a^{(z-n)}) + e^{-j[(z+n)\theta+\delta_z]} (1 + a^{(z-n)} + a^{-(z-n)}) - \\ e^{j[(z-n)\theta+\delta_z]} (1 + a^{(z+n)} + a^{-(z+n)}) - e^{-j[(z-n)\theta+\delta_z]} (1 + a^{(z+n)} + a^{-(z+n)}) \\ e^{j[(z+n)\theta+\delta_z]} (a^n + a^{-(z+n)} + a^z) + e^{-j[(z+n)\theta+\delta_z]} (a^{-n} + a^{(z+n)} + a^{-z}) - \\ e^{j[(z-n)\theta+\delta_z]} (a^{-n} + a^{-(z-n)} + a^z) - e^{-j[(z-n)\theta+\delta_z]} (a^n + a^{(z-n)} + a^{-z}) \\ e^{j[(z+n)\theta+\delta_z]} (a^{-n} + a^{-z} + a^{(z+n)}) + e^{-j[(z+n)\theta+\delta_z]} (a^n + a^z + a^{-(z+n)}) - \\ e^{j[(z-n)\theta+\delta_z]} (a^n + a^{-z} + a^{(z-n)}) - e^{-j[(z-n)\theta+\delta_z]} (a^{-n} + a^z + a^{-(z-n)}) \end{array} \right] \quad (\text{A.17})$$

$$\Gamma_{réel} = p \frac{1}{8} \sum_{m=1} \sum_{k=1} \sum_{\substack{z=1 \\ n=2k}} n \frac{L_n i_m i_z}{j2} \left[\begin{array}{l} + e^{j[(n+m+z)\theta+\delta_m+\delta_z]} (1 + a^{-(z-n)} + a^{(z-n)}) \\ + e^{-j[(n-m+z)\theta-\delta_m+\delta_z]} (1 + a^{(z-n)} + a^{-(z-n)}) \\ - e^{-j[(n-m-z)\theta-\delta_m-\delta_z]} (1 + a^{(z+n)} + a^{-(z+n)}) \\ - e^{j[(n+m-z)\theta+\delta_m-\delta_z]} (1 + a^{(z+n)} + a^{-(z+n)}) \\ - e^{j[(n-m+z)\theta-\delta_m+\delta_z]} (1 + a^{-(z-n)} + a^{(z-n)}) \\ - e^{-j[(n+m+z)\theta+\delta_m+\delta_z]} (1 + a^{(z-n)} + a^{-(z-n)}) \\ + e^{-j[(n+m-z)\theta+\delta_m-\delta_z]} (1 + a^{(z+n)} + a^{-(z+n)}) \\ + e^{j[(n-m-z)\theta-\delta_m-\delta_z]} (1 + a^{(z+n)} + a^{-(z+n)}) \\ + e^{j[(n+m+z)\theta+\delta_m+\delta_z]} (a^n + a^{-(z+n)} + a^z) a^{-m} \\ + e^{-j[(n-m+z)\theta-\delta_m+\delta_z]} (a^{-n} + a^{(z+n)} + a^{-z}) a^{-m} \\ - e^{-j[(n-m-z)\theta-\delta_m-\delta_z]} (a^{-n} + a^{-(z-n)} + a^z) a^{-m} \\ - e^{j[(n+m-z)\theta+\delta_m-\delta_z]} (a^n + a^{(z-n)} + a^{-z}) a^{-m} \\ - e^{j[(n-m+z)\theta-\delta_m+\delta_z]} (a^n + a^{-(z+n)} + a^z) a^m \\ - e^{-j[(n+m+z)\theta+\delta_m+\delta_z]} (a^{-n} + a^{(z+n)} + a^{-z}) a^m \\ + e^{-j[(n+m-z)\theta+\delta_m-\delta_z]} (a^{-n} + a^{-(z-n)} + a^z) a^m \\ + e^{j[(n-m-z)\theta-\delta_m-\delta_z]} (a^n + a^{(z-n)} + a^{-z}) a^m \\ + e^{j[(n+m+z)\theta+\delta_m+\delta_z]} (a^{-n} + a^{-z} + a^{(z+n)}) a^m \\ + e^{-j[(n-m+z)\theta-\delta_m+\delta_z]} (a^n + a^z + a^{-(z+n)}) a^m \\ - e^{-j[(n-m-z)\theta-\delta_m-\delta_z]} (a^n + a^{-z} + a^{(z-n)}) a^m \\ - e^{j[(n+m-z)\theta+\delta_m-\delta_z]} (a^{-n} + a^z + a^{-(z-n)}) a^m \\ - e^{j[(n-m+z)\theta-\delta_m+\delta_z]} (a^{-n} + a^{-z} + a^{(z+n)}) a^{-m} \\ - e^{-j[(n+m+z)\theta+\delta_m+\delta_z]} (a^n + a^z + a^{-(z+n)}) a^{-m} \\ + e^{-j[(n+m-z)\theta+\delta_m-\delta_z]} (a^n + a^{-z} + a^{(z-n)}) a^{-m} \\ + e^{j[(n-m-z)\theta-\delta_m-\delta_z]} (a^{-n} + a^z + a^{-(z-n)}) a^{-m} \end{array} \right] \quad (\text{A.18})$$

$$\Gamma_{réel} = p \frac{1}{8} \sum_{m=1} \sum_{k=1} \sum_{z=1} n \frac{L_n i_m i_z}{j2}$$

$$\left[\begin{array}{l} + e^{j[(n+m+z)\theta + \delta_m + \delta_z]} \left(1 + a^{-(z-n)} + a^{(z-n)} + a^{-(z-m)} + a^{(z-m)} + a^{-(n-m)} + a^{(n-m)} + a^{-(z+n+m)} + a^{(z+n+m)} \right) \\ - e^{-j[(n+m+z)\theta + \delta_m + \delta_z]} \left(1 + a^{(z-n)} + a^{-(z-n)} + a^{-(z-m)} + a^{(z-m)} + a^{-(n-m)} + a^{(n-m)} + a^{-(z+n+m)} + a^{(z+n+m)} \right) \\ - e^{j[(n-m+z)\theta - \delta_m + \delta_z]} \left(1 + a^{-(z-n)} + a^{(z-n)} + a^{-(z+m)} + a^{-(n+m)} + a^{(z+n-m)} + a^{(z+m)} + a^{(n+m)} + a^{-(z+n-m)} \right) \\ + e^{-j[(n-m+z)\theta - \delta_m + \delta_z]} \left(1 + a^{(z-n)} + a^{-(z-n)} + a^{-(z+m)} + a^{-(n+m)} + a^{(z+n-m)} + a^{(z+m)} + a^{(n+m)} + a^{-(z+n-m)} \right) \\ - e^{j[(n+m-z)\theta + \delta_m - \delta_z]} \left(1 + a^{(z+n)} + a^{-(z+n)} + a^{(n-m)} + a^{-(z+m)} + a^{(-m+z-n)} + a^{-(n-m)} + a^{(z+m)} + a^{(m-z+n)} \right) \\ + e^{-j[(n+m-z)\theta + \delta_m - \delta_z]} \left(1 + a^{(z+n)} + a^{-(z+n)} + a^{(n-m)} + a^{-(z+m)} + a^{(-m+z-n)} + a^{-(n-m)} + a^{(z+m)} + a^{(m-z+n)} \right) \\ + e^{j[(n-m-z)\theta - \delta_m - \delta_z]} \left(1 + a^{(z+n)} + a^{-(z+n)} + a^{-(n+m)} + a^{(z-m)} + a^{-(m+z-n)} + a^{(n+m)} + a^{-(z-m)} + a^{(m+z-n)} \right) \\ - e^{-j[(n-m-z)\theta - \delta_m - \delta_z]} \left(1 + a^{(z+n)} + a^{-(z+n)} + a^{-(n+m)} + a^{(z-m)} + a^{-(m+z-n)} + a^{(n+m)} + a^{-(z-m)} + a^{(m+z-n)} \right) \end{array} \right] \quad (\text{A.19})$$

$$\Gamma_{réel} = p \frac{1}{8} \sum_{m=1} \sum_{k=1} \sum_{z=1} n L_n i_m i_z$$

$$\left[\begin{array}{l} + \left(1 + a^{-(z-n)} + a^{(z-n)} + a^{-(z-m)} + a^{(z-m)} + a^{-(n-m)} + a^{(n-m)} + a^{-(z+n+m)} + a^{(z+n+m)} \right) \\ \quad \sin[(n+m+z)\theta + \delta_m + \delta_z] \\ - \left(1 + a^{(z-n)} + a^{-(z-n)} + a^{-(z+m)} + a^{-(n+m)} + a^{(z+n-m)} + a^{(z+m)} + a^{(n+m)} + a^{-(z+n-m)} \right) \\ \quad \sin[(n-m+z)\theta - \delta_m + \delta_z] \\ - \left(1 + a^{(z+n)} + a^{-(z+n)} + a^{(n-m)} + a^{-(z+m)} + a^{(-m+z-n)} + a^{-(n-m)} + a^{(z+m)} + a^{(m-z+n)} \right) \\ \quad \sin[(n+m-z)\theta + \delta_m - \delta_z] \\ + \left(1 + a^{(z+n)} + a^{-(z+n)} + a^{-(n+m)} + a^{(z-m)} + a^{-(m+z-n)} + a^{(n+m)} + a^{-(z-m)} + a^{(m+z-n)} \right) \\ \quad \sin[(n-m-z)\theta - \delta_m - \delta_z] \end{array} \right] \quad (\text{A.20})$$

Dans ce cas, nous avons une somme triple en m , k et z . La division de cette somme en harmoniques multiples de trois est faite en suivant une variante du développement donné dans l'équation (A.9). Cependant cette division donne 27 termes ! Cette équation est trop grande pour figurer ici et, par conséquent, nous avons opté pour faire la simplification ailleurs et rapporter uniquement le résultat simplifié :

$$\Gamma_{\text{réel}} = p \frac{9}{8} \sum_{k=1} \sum_{o=1} \sum_{r=1} \left[\begin{aligned} &+ (6k-4)L_{(6k-4)}i_{(3o-1)}i_{(3r-1)} \sin[(6k+3o+3r-6)\theta + \delta_{(3o-1)} + \delta_{(3r-1)}] \\ &+ (6k-2)L_{(6k-2)}i_{(3o-2)}i_{(3r-2)} \sin[(6k+3o+3r-6)\theta + \delta_{(3o-2)} + \delta_{(3r-2)}] \\ &+ (6k)L_{(6k)}i_{(3o)}i_{(3r)} \sin[(6k+3o+3r)\theta + \delta_{(3o)} + \delta_{(3r)}] \\ &- (6k-4)L_{(6k-4)}i_{(3o-2)}i_{(3r-1)} \sin[(6k-3o+3r-3)\theta - \delta_{(3o-2)} + \delta_{(3r-1)}] \\ &- (6k-2)L_{(6k-2)}i_{(3o-1)}i_{(3r-2)} \sin[(6k-3o+3r-3)\theta - \delta_{(3o-1)} + \delta_{(3r-2)}] \\ &- (6k)L_{(6k)}i_{(3o)}i_{(3r)} \sin[(6k-3o+3r)\theta - \delta_{(3o)} + \delta_{(3r)}] \\ &- (6k-4)L_{(6k-4)}i_{(3o-1)}i_{(3r-2)} \sin[(6k+3o-3r-3)\theta + \delta_{(3o-1)} - \delta_{(3r-2)}] \\ &- (6k-2)L_{(6k-2)}i_{(3o-2)}i_{(3r-1)} \sin[(6k+3o-3r-3)\theta + \delta_{(3o-2)} - \delta_{(3r-1)}] \\ &- (6k)L_{(6k)}i_{(3o)}i_{(3r)} \sin[(6k+3o-3r)\theta + \delta_{(3o)} - \delta_{(3r)}] \\ &+ (6k-4)L_{(6k-4)}i_{(3o-2)}i_{(3r-2)} \sin[(6k-3o-3r)\theta - \delta_{(3o-2)} - \delta_{(3r-2)}] \\ &+ (6k-2)L_{(6k-2)}i_{(3o-1)}i_{(3r-1)} \sin[(6k-3o-3r)\theta - \delta_{(3o-1)} - \delta_{(3r-1)}] \\ &+ (6k)L_{(6k)}i_{(3o)}i_{(3r)} \sin[(6k-3o-3r)\theta - \delta_{(3o)} - \delta_{(3r)}] \end{aligned} \right] \quad (\text{A.21})$$

En supprimant les harmoniques multiples de 3 du courant :

$$i_{3o} = 0, i_{3r} = 0, \quad \forall o, r \in N^+$$

$$\Gamma_{\text{réel}} = p \frac{9}{8} \sum_{k=1} \sum_{o=1} \sum_{r=1} \left[\begin{aligned} &+ (6k-4)L_{(6k-4)}i_{(3o-1)}i_{(3r-1)} \sin[(6k+3o+3r-6)\theta + \delta_{(3o-1)} + \delta_{(3r-1)}] \\ &+ (6k-2)L_{(6k-2)}i_{(3o-2)}i_{(3r-2)} \sin[(6k+3o+3r-6)\theta + \delta_{(3o-2)} + \delta_{(3r-2)}] \\ &- (6k-4)L_{(6k-4)}i_{(3o-2)}i_{(3r-1)} \sin[(6k-3o+3r-3)\theta - \delta_{(3o-2)} + \delta_{(3r-1)}] \\ &- (6k-2)L_{(6k-2)}i_{(3o-1)}i_{(3r-2)} \sin[(6k-3o+3r-3)\theta - \delta_{(3o-1)} + \delta_{(3r-2)}] \\ &- (6k)L_{(6k)}i_{(3o)}i_{(3r)} \sin[(6k+3o+3r)\theta + \delta_{(3o)} + \delta_{(3r)}] \\ &- (6k-4)L_{(6k-4)}i_{(3o-1)}i_{(3r-2)} \sin[(6k+3o-3r-3)\theta + \delta_{(3o-1)} - \delta_{(3r-2)}] \\ &- (6k-2)L_{(6k-2)}i_{(3o-2)}i_{(3r-1)} \sin[(6k+3o-3r-3)\theta + \delta_{(3o-2)} - \delta_{(3r-1)}] \\ &+ (6k-4)L_{(6k-4)}i_{(3o-2)}i_{(3r-2)} \sin[(6k-3o-3r)\theta - \delta_{(3o-2)} - \delta_{(3r-2)}] \\ &+ (6k-2)L_{(6k-2)}i_{(3o-1)}i_{(3r-1)} \sin[(6k-3o-3r)\theta - \delta_{(3o-1)} - \delta_{(3r-1)}] \end{aligned} \right] \quad (\text{A.22})$$

Cette dernière équation représente les harmoniques de couple générés par l'interaction des variations de la réluctance avec un courant non sinusoïdal.

Si le courant n'a que des harmoniques impairs, alors le couple devient :

$$\Gamma_{\text{réel}} = p \frac{9}{8} \sum_{k=1} \sum_{o=1} \sum_{r=1} \left[\begin{aligned} &+ (6k-4)L_{(6k-4)}i_{(6o-1)}i_{(6r-1)} \sin[(6k+6o+6r-6)\theta + \delta_{(6o-1)} + \delta_{(6r-1)}] \\ &+ (6k-2)L_{(6k-2)}i_{(6o-5)}i_{(6r-5)} \sin[(6k+6o+6r-12)\theta + \delta_{(6o-5)} + \delta_{(6r-5)}] \\ &- (6k-4)L_{(6k-4)}i_{(6o-5)}i_{(6r-1)} \sin[(6k-6o+6r)\theta - \delta_{(6o-5)} + \delta_{(6r-1)}] \\ &- (6k-2)L_{(6k-2)}i_{(6o-1)}i_{(6r-5)} \sin[(6k-6o+6r-6)\theta - \delta_{(6o-1)} + \delta_{(6r-5)}] \\ &- (6k)L_{(6k)}i_{(6o)}i_{(6r)} \sin[(6k+6o+6r)\theta + \delta_{(6o)} + \delta_{(6r)}] \\ &- (6k-4)L_{(6k-4)}i_{(6o-1)}i_{(6r-5)} \sin[(6k+6o-6r)\theta + \delta_{(6o-1)} - \delta_{(6r-5)}] \\ &- (6k-2)L_{(6k-2)}i_{(6o-5)}i_{(6r-1)} \sin[(6k+6o-6r-6)\theta + \delta_{(6o-5)} - \delta_{(6r-1)}] \\ &+ (6k-4)L_{(6k-4)}i_{(6o-5)}i_{(6r-5)} \sin[(6k-6o-6r+6)\theta - \delta_{(6o-5)} - \delta_{(6r-5)}] \\ &+ (6k-2)L_{(6k-2)}i_{(6o-1)}i_{(6r-1)} \sin[(6k-6o-6r)\theta - \delta_{(6o-1)} - \delta_{(6r-1)}] \end{aligned} \right] \quad (\text{A.23})$$

Nous pouvons encore constater que, dans le cas où le courant n'a que des harmoniques impairs, le couple n'a que des harmoniques multiples de 6.

A.3 Harmoniques de couple avec un courant avec des fréquences subharmoniques.

Dans ce cas, nous considérons un courant dont les harmoniques ne sont pas uniquement des multiples de la fréquence fondamentale des courants. Ainsi, il existe des harmoniques à d'autres fréquences non multiples de la fréquence fondamentale (une source possible de ce type d'harmonique sont les harmoniques de la tension de bus de l'onduleur) :

$$\mathbf{i}_s = \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} = \sum_{z=1} i_z \begin{bmatrix} \sin[r_z \theta + \delta_z] \\ \sin[r_z \theta + \delta_z - 2\pi/3] \\ \sin[r_z \theta + \delta_z - 4\pi/3] \end{bmatrix} = \sum_{z=1} \frac{i_z}{2j} \begin{bmatrix} e^{j[r_z \theta + \delta_z]} - e^{-j[r_z \theta + \delta_z]} \\ e^{j[r_z \theta + \delta_z]} a^{-1} - e^{-j[r_z \theta + \delta_z]} a^1 \\ e^{j[r_z \theta + \delta_z]} a^{-2} - e^{-j[r_z \theta + \delta_z]} a^2 \end{bmatrix} \quad (\text{A.24})$$

où r_z est un nombre réel représentant l'harmonique et δ_z est le déphasage de cette harmonique.

En utilisant ce courant pour calculer les harmoniques de couple :

$$\Gamma_{flux} = \mathbf{i}_s^T \frac{d\mathbf{k}(\theta)}{d\theta} p = -p \mathbf{i}_s^T \sum_{\substack{k=1 \\ n=2k-1}} n \frac{k_n}{j2} \begin{bmatrix} e^{jn\theta} - e^{-jn\theta} \\ e^{jn\theta} a^{-n} - e^{-jn\theta} a^n \\ e^{jn\theta} a^{-2n} - e^{-jn\theta} a^{2n} \end{bmatrix}$$

$$\Gamma_{flux} = p \sum_{k=1} \sum_{n=2k-1} n \frac{i_z k_n}{4} \begin{bmatrix} e^{j[(r_z+n)\theta + \delta_z]} + e^{-j[(r_z+n)\theta + \delta_z]} - e^{j[(r_z-n)\theta + \delta_z]} - e^{-j[(r_z-n)\theta + \delta_z]} \\ + e^{j[(r_z+n)\theta + \delta_z]} a^{-(1+n)} + e^{-j[(r_z+n)\theta + \delta_z]} a^{(1+n)} \\ - e^{j[(r_z-n)\theta + \delta_z]} a^{-(1-n)} - e^{-j[(r_z-n)\theta + \delta_z]} a^{(1-n)} \\ + e^{j[(r_z+n)\theta + \delta_z]} a^{-2(1+n)} + e^{-j[(r_z+n)\theta + \delta_z]} a^{2(1+n)} \\ - e^{j[(r_z-n)\theta + \delta_z]} a^{-2(1-n)} - e^{-j[(r_z-n)\theta + \delta_z]} a^{2(1-n)} \end{bmatrix} = \quad (\text{A.25})$$

$$\Gamma_{flux} = p \sum_{k=1} \sum_{n=2k-1} n \frac{i_z k_n}{4} \left[\left(e^{j[(r_z+n)\theta + \delta_z]} + e^{-j[(r_z+n)\theta + \delta_z]} \right) \left(1 + a^{(1+n)} + a^{2(1+n)} \right) - \left(e^{j[(r_z-n)\theta + \delta_z]} + e^{-j[(r_z-n)\theta + \delta_z]} \right) \left(1 + a^{(1-n)} + a^{2(1-n)} \right) \right] =$$

$$\Gamma_{flux} = p \sum_{k=1} \sum_{n=2k-1} n \frac{i_z k_n}{2} \left[\left(1 + a^{(1+n)} + a^{2(1+n)} \right) \cos[(r_z + n)\theta + \delta_z] - \left(1 + a^{(1-n)} + a^{2(1-n)} \right) \cos[(r_z - n)\theta + \delta_z] \right] =$$

$$\Gamma_{flux} = \frac{3}{2} p \sum_{k=1} \sum_{z=1} i_z \left[\begin{matrix} (6k-1)k_{(6k-1)} \cos[(r_z + 6k-1)\theta + \delta_z] - \\ (6k-5)k_{(6k-5)} \cos[(r_z - 6k+5)\theta + \delta_z] \end{matrix} \right] \quad (\text{A.26})$$

En suivant la même démarche pour le couple réductant :

$$\Gamma_{rél} = p \frac{1}{2} \mathbf{i}_s^T \frac{d\mathbf{L}_s(\theta)}{d\theta} \mathbf{i}_s \quad (\text{A.27})$$

$$\Gamma_{rél} = p \frac{1}{8} \mathbf{i}_s^T \sum_{k=1} \sum_{\substack{z=1 \\ n=2k}} n L_n \mathbf{i}_z \left[\begin{array}{l} e^{j[(n+r_z)\theta+\delta_z]} (1 + a^{-(1-n)} + a^{(1-n)}) + e^{-j[(n+r_z)\theta+\delta_z]} (1 + a^{(1-n)} + a^{-(1-n)}) - \\ e^{j[(n-r_z)\theta-\delta_z]} (1 + a^{(1+n)} + a^{-(1+n)}) - e^{-j[(n-r_z)\theta-\delta_z]} (1 + a^{(1+n)} + a^{-(1+n)}) \\ e^{j[(n+r_z)\theta+\delta_z]} (a^n + a^{-(1+n)} + a^1) + e^{-j[(n+r_z)\theta+\delta_z]} (a^{-n} + a^{(1+n)} + a^{-1}) - \\ e^{j[(n-r_z)\theta-\delta_z]} (a^n + a^{(1-n)} + a^{-1}) - e^{-j[(n-r_z)\theta-\delta_z]} (a^{-n} + a^{-(1-n)} + a^1) \\ e^{j[(n+r_z)\theta+\delta_z]} (a^{-n} + a^{-1} + a^{(1+n)}) + e^{-j[(n+r_z)\theta+\delta_z]} (a^n + a^1 + a^{-(1+n)}) - \\ e^{j[(n-r_z)\theta-\delta_z]} (a^{-n} + a^1 + a^{-(1-n)}) - e^{-j[(n-r_z)\theta-\delta_z]} (a^n + a^{-1} + a^{(1-n)}) \end{array} \right] \quad (\text{A.28})$$

$$\Gamma_{rél} = p \frac{1}{8} \sum_{m=1} \sum_{k=1} \sum_{z=1} n L_n \frac{i_m i_z}{2j} \left[\begin{array}{l} + e^{j[(n+r_m+r_z)\theta+\delta_m+\delta_z]} [1 + a^{-(1-n)} + a^{(1-n)} + (a^n + a^{-(1+n)} + a)a^{-1} + (a^{-n} + a^{-1} + a^{(1+n)})a] \\ - e^{-j[(n+r_m+r_z)\theta+\delta_m+\delta_z]} [1 + a^{(1-n)} + a^{-(1-n)} + (a^{-n} + a^{(1+n)} + a^{-1})a + (a^n + a + a^{-(1+n)})a^{-1}] \\ - e^{j[(n-r_m+r_z)\theta-\delta_m+\delta_z]} [1 + a^{-(1-n)} + a^{(1-n)} + (a^n + a^{-(1+n)} + a)a + (a^{-n} + a^{-1} + a^{(1+n)})a^{-1}] \\ + e^{-j[(n-r_m+r_z)\theta-\delta_m+\delta_z]} [1 + a^{(1-n)} + a^{-(1-n)} + (a^{-n} + a^{(1+n)} + a^{-1})a^{-1} + (a^n + a + a^{-(1+n)})a] \\ - e^{j[(n+r_m-r_z)\theta+\delta_m-\delta_z]} [1 + a^{(1+n)} + a^{-(1+n)} + (a^n + a^{(1-n)} + a^{-1})a^{-1} + (a^{-n} + a^{-(1-n)} + a)a] \\ + e^{-j[(n+r_m-r_z)\theta+\delta_m-\delta_z]} [1 + a^{-(1+n)} + a^{(1+n)} + (a^{-n} + a^{-(1-n)} + a)a + (a^n + a^{(1-n)} + a^{-1})a^{-1}] \\ + e^{j[(n-r_m-r_z)\theta-\delta_m-\delta_z]} [1 + a^{(1+n)} + a^{-(1+n)} + (a^n + a^{(1-n)} + a^{-1})a + (a^{-n} + a^{-(1-n)} + a)a^{-1}] \\ - e^{-j[(n-r_m-r_z)\theta-\delta_m-\delta_z]} [1 + a^{-(1+n)} + a^{(1+n)} + (a^{-n} + a^{-(1-n)} + a)a^{-1} + (a^n + a^{(1-n)} + a^{-1})a] \end{array} \right] \quad (\text{A.29})$$

$$\Gamma_{rél} = p \frac{3}{8} \sum_{m=1} \sum_{k=1} \sum_{z=1} n L_n \frac{i_m i_z}{2j} \left[\begin{array}{l} + (e^{j[(n+r_m+r_z)\theta+\delta_m+\delta_z]} - e^{-j[(n+r_m+r_z)\theta+\delta_m+\delta_z]})(1 + a^{-(1-n)} + a^{(1-n)}) \\ + (e^{j[(n-r_m-r_z)\theta-\delta_m-\delta_z]} - e^{-j[(n-r_m-r_z)\theta-\delta_m-\delta_z]})(1 + a^{(1+n)} + a^{-(1+n)}) \end{array} \right]$$

$$\Gamma_{rél} = p \frac{3}{8} \sum_{m=1} \sum_{k=1} \sum_{z=1} n L_n i_m i_z \left[\begin{array}{l} + \sin[(n+r_m+r_z)\theta+\delta_m+\delta_z](1 + a^{-(1-n)} + a^{(1-n)}) \\ + \sin[(n-r_m-r_z)\theta-\delta_m-\delta_z](1 + a^{(1+n)} + a^{-(1+n)}) \end{array} \right]$$

$$\Gamma_{rél} = p \frac{9}{8} \sum_{m=1} \sum_{k=1} \sum_{z=1} i_m i_z \left[\begin{array}{l} + (6k-5)L_{(6k-5)} \sin[(6k-5+r_m+r_z)\theta+\delta_m+\delta_z] \\ + (6k-1)L_{(6k-1)} \sin[(6k-1-r_m-r_z)\theta-\delta_m-\delta_z] \end{array} \right] \quad (\text{A.30})$$

Nous pouvons constater que les équations (A.26) et (A.30) sont difficilement exploitables sous cette forme. Nous ne pouvons rien conclure sans avoir plus de renseignements sur les harmoniques du courant.

Une autre approche consiste en transformer le courant sous la forme suivante :

$$\mathbf{i}_s = \left[1 + \sum_k A_k \cos(2\pi k f_1 t + \gamma_k) \right] \sum_{n=1} i'_n \begin{bmatrix} \sin[n(\theta) + \delta_n] \\ \sin[n(\theta - 2\pi/3) + \delta_n] \\ \sin[n(\theta - 4\pi/3) + \delta_n] \end{bmatrix} \quad (\text{A.31})$$

$$i_{3o} = 0, \quad \forall o \in N^+$$

Où A_k et γ_k sont l'amplitude et le déphasage des harmoniques générés par une source « externe » d'harmoniques et f_1 est la fréquence fondamentale de ce source (par exemple, la tension de bus). i_n et δ_n sont l'amplitude et le déphasage du courant sans la perturbation introduite par la source « externe ».

Avec le courant de l'équation (A.31), les équations de couple (A.13) et (A.22) sont modifiées comme suit:

$$\Gamma'_{flux} = \left[1 + \sum_k A_k \cos(2\pi k f_1 t + \gamma_k) \right] \Gamma_{flux} \quad (\text{A.32})$$

$$\Gamma'_{rel} = \left[1 + \sum_k A_k \cos(2\pi k f_1 t + \gamma_k) \right]^2 \Gamma_{rel} \quad (\text{A.33})$$

Nous pouvons constater que le couple aura les mêmes harmoniques que précédemment, mais avec des harmoniques supplémentaires provenant de la fréquence de la source externe. Le couple aura donc une combinaison des harmoniques originaires de la source externe et des harmoniques de la référence de tension fournie par la commande.

L'obtention d'un courant sous la forme (A.31) est très difficile de manière générale. Cependant, si nous connaissons les sources d'harmoniques externes, nous pouvons séparément mesurer les harmoniques de ces sources et avoir une idée des harmoniques les plus importants. Nous pouvons aussi utiliser la mesure de courant pour calculer ses harmoniques et par conséquent, les harmoniques de couple en utilisant les formules (A.26) et (A.30).

A.4 Harmoniques de couple en présence d'un décalage de mesure des courants.

Supposons que nous ayons une mesure des courants avec des décalages comme suit:

$$\begin{aligned} i_{am} &= i_a + \Delta i_a \\ i_{bm} &= i_b + \Delta i_b \\ i_{cm} &= -i_{am} - i_{bm} \end{aligned} \quad (\text{A.34})$$

A basse vitesse, le correcteur de courant arrive à contrôler les courants i_{am} et i_{am} pour qu'ils suivent les références sinusoïdales et, par conséquent, les courants i_a , i_b et i_c sont sinusoïdaux mais ils ont une composante continue. Ainsi, les courants dans la machine s'écrivent :

$$\mathbf{i}'_s = \mathbf{i}_s + \Delta\mathbf{i}_s \quad (\text{A.35})$$

où \mathbf{i}_s correspond à l'équation (A.6) et $\Delta\mathbf{i}_s$ est :

$$\Delta\mathbf{i}_s = \begin{bmatrix} \Delta i_a \\ \Delta i_b \\ -\Delta i_a - \Delta i_b \end{bmatrix} \quad (\text{A.36})$$

Ce décalage étant un terme additif, nous pouvons développer les équations de couple comme suit à partir des équations (A.7) et (A.15) :

$$\begin{aligned} \Gamma''_{flux} &= \Gamma_{flux} + \begin{bmatrix} \Delta i_a \\ \Delta i_b \\ -\Delta i_a - \Delta i_b \end{bmatrix}^T \frac{d\mathbf{k}(\theta)}{d\theta} p = \\ \Gamma''_{flux} &= \Gamma_{flux} - p \sum_{\substack{k=1 \\ n=2k-1}} n \frac{k_n}{j2} \left[\begin{array}{l} \Delta i_a \left[e^{jn\theta} (1 - a^{-2n}) - e^{-jn\theta} (1 - a^{2n}) \right] + \\ \Delta i_b \left[e^{jn\theta} (a^{-n} - a^{-2n}) - e^{-jn\theta} (a^n - a^{2n}) \right] \end{array} \right] = \end{aligned} \quad (\text{A.37})$$

$$\begin{aligned} \Gamma''_{flux} &= \Gamma_{flux} - p \sum_{k=1} \frac{1}{j2} \left[\begin{array}{l} (6k-5)k_{(6k-5)} \Delta i_a \left[e^{j(6k-5)\theta} (1-a) - e^{-j(6k-5)\theta} (1-a^{-1}) \right] + \\ (6k-5)k_{(6k-5)} \Delta i_b \left[e^{j(6k-5)\theta} (a^{-1}-a) - e^{-j(6k-5)\theta} (a-a^{-1}) \right] + \\ (6k-1)k_{(6k-1)} \Delta i_a \left[e^{j(6k-1)\theta} (1-a^{-1}) - e^{-j(6k-1)\theta} (1-a) \right] + \\ (6k-1)k_{(6k-1)} \Delta i_b \left[e^{j(6k-1)\theta} (a-a^{-1}) - e^{-j(6k-1)\theta} (a^{-1}-a) \right] \end{array} \right] = \\ \Gamma''_{flux} &= \Gamma_{flux} - p \sum_{k=1} \left[\begin{array}{l} (6k-5)k_{(6k-5)} \sqrt{3} [\Delta i_a \sin[(6k-5)\theta - \pi/6] - \Delta i_b \cos[(6k-5)\theta]] + \\ (6k-1)k_{(6k-1)} \sqrt{3} [\Delta i_a \sin[(6k-1)\theta + \pi/6] + \Delta i_b \cos[(6k-1)\theta]] \end{array} \right] \end{aligned} \quad (\text{A.38})$$

Nous pouvons constater qu'un décalage des courants crée des harmoniques à la fréquence fondamentale des courants dont l'amplitude est proportionnelle aux décalages. Plus généralement, nous pouvons dire que le décalage crée des harmoniques impairs de couple excepté les multiples de 3.

La même démarche est suivie pour le couple réductant en partant des équations (A.7) et (A.22) :

$$\Gamma_{rel} = p \frac{1}{2} \mathbf{i}_s^T \frac{d\mathbf{L}_s(\theta)}{d\theta} \mathbf{i}_s \quad (\text{A.39})$$

$$\Gamma_{\text{réel}}'' = \Gamma_{\text{réel}} + p \Delta \mathbf{i}_s^T \frac{d\mathbf{L}_s(\theta)}{d\theta} \left(\mathbf{i}_s + \frac{1}{2} \Delta \mathbf{i}_s \right) =$$

$$\Gamma_{\text{réel}}'' = \Gamma_{\text{réel}} + p - \sum_{\substack{k=1 \\ n=2k}} n \frac{L_n}{j2} \quad (\text{A.40})$$

$$\begin{bmatrix} \Delta i_a \left[e^{jn\theta} (1 - a^{-n}) - e^{-jn\theta} (1 - a^n) \right] + \Delta i_b \left[e^{jn\theta} (a^n - a^{-n}) - e^{-jn\theta} (a^{-n} - a^n) \right] \\ \Delta i_a \left[e^{jn\theta} (a^n - 1) - e^{-jn\theta} (a^{-n} - 1) \right] + \Delta i_b \left[e^{jn\theta} (a^{-n} - 1) - e^{-jn\theta} (a^n - 1) \right] \\ \Delta i_a \left[e^{jn\theta} (a^{-n} - a^n) - e^{-jn\theta} (a^n - a^{-n}) \right] + \Delta i_b \left[e^{jn\theta} (1 - a^n) - e^{-jn\theta} (1 - a^{-n}) \right] \end{bmatrix}^T \left(\mathbf{i}_s + \frac{1}{2} \Delta \mathbf{i}_s \right) =$$

$$\Gamma_{\text{réel}}'' = \Gamma_{\text{réel}} + p \sum_{z=1} \sum_{\substack{k=1 \\ n=2k}} n \frac{L_n i_z}{4}$$

$$\begin{bmatrix} \Delta i_a \left[e^{jn\theta} (1 - a^{-n}) - e^{-jn\theta} (1 - a^n) \right] + \Delta i_b \left[e^{jn\theta} (a^n - a^{-n}) - e^{-jn\theta} (a^{-n} - a^n) \right] \\ \Delta i_a \left[e^{jn\theta} (a^n - 1) - e^{-jn\theta} (a^{-n} - 1) \right] + \Delta i_b \left[e^{jn\theta} (a^{-n} - 1) - e^{-jn\theta} (a^n - 1) \right] \\ \Delta i_a \left[e^{jn\theta} (a^{-n} - a^n) - e^{-jn\theta} (a^n - a^{-n}) \right] + \Delta i_b \left[e^{jn\theta} (1 - a^n) - e^{-jn\theta} (1 - a^{-n}) \right] \end{bmatrix}^T \quad (\text{A.41})$$

$$\begin{bmatrix} e^{j(z\theta + \delta_z)} - e^{-j(z\theta + \delta_z)} \\ e^{j(z\theta + \delta_z)} a^{-z} - e^{-j(z\theta + \delta_z)} a^z \\ e^{j(z\theta + \delta_z)} a^{-2z} - e^{-j(z\theta + \delta_z)} a^{2z} \end{bmatrix} + p \sum_{\substack{k=1 \\ n=2k}} n \frac{-L_n}{j2}$$

$$\begin{bmatrix} \Delta i_a^2 \left[e^{jn\theta} (1 - a^{-n}) - e^{-jn\theta} (1 - a^n) \right] + \Delta i_a \Delta i_b \left[e^{jn\theta} (a^n - a^{-n}) - e^{-jn\theta} (a^{-n} - a^n) \right] + \\ \Delta i_a \Delta i_b \left[e^{jn\theta} (a^n - 1) - e^{-jn\theta} (a^{-n} - 1) \right] + \Delta i_b^2 \left[e^{jn\theta} (a^{-n} - 1) - e^{-jn\theta} (a^n - 1) \right] - \\ (\Delta i_a + \Delta i_b) \left[\Delta i_a \left[e^{jn\theta} (a^{-n} - a^n) - e^{-jn\theta} (a^n - a^{-n}) \right] + \Delta i_b \left[e^{jn\theta} (1 - a^n) - e^{-jn\theta} (1 - a^{-n}) \right] \right] \end{bmatrix}$$

$$\Gamma_{\text{réel}}'' = \Gamma_{\text{réel}} + p \sum_{z=1} \sum_{\substack{k=1 \\ n=2k}} n \frac{L_n i_z}{4} \begin{bmatrix} \Delta i_a \begin{bmatrix} + e^{j[(n+z)\theta + \delta_z]} (1 - a^{-n} + a^{(n-z)} - a^{-z} + a^{-(n-z)} - a^{(n+z)}) \\ + e^{-j[(n+z)\theta + \delta_z]} (1 - a^n + a^{-(n-z)} - a^z + a^{(n-z)} - a^{-(n+z)}) \\ - e^{j[(n-z)\theta - \delta_z]} (1 - a^{-n} + a^{(n+z)} - a^z + a^{-(n+z)} - a^{(n-z)}) \\ - e^{-j[(n-z)\theta - \delta_z]} (1 - a^n + a^{-(n+z)} - a^{-z} + a^{(n+z)} - a^{-(n-z)}) \end{bmatrix} + \\ \Delta i_b \begin{bmatrix} e^{j[(n+z)\theta + \delta_z]} (a^n - a^{-n} + a^{-(n+z)} - a^{-z} + a^z - a^{(n+z)}) \\ + e^{-j[(n+z)\theta + \delta_z]} (a^{-n} - a^n + a^{(n+z)} - a^z + a^{-z} - a^{-(n+z)}) \\ - e^{j[(n-z)\theta - \delta_z]} (a^n - a^{-n} + a^{-(n-z)} - a^z + a^{-z} - a^{(n-z)}) \\ - e^{-j[(n-z)\theta - \delta_z]} (a^{-n} - a^n + a^{(n-z)} - a^{-z} + a^z - a^{-(n-z)}) \end{bmatrix} \end{bmatrix} \quad (\text{A.42})$$

$$+ p \sum_{\substack{k=1 \\ n=2k}} n \frac{-L_n}{j2} \begin{bmatrix} \Delta i_a^2 \left[e^{jn\theta} (1 - a^{-n} - a^{-n} + a^n) - e^{-jn\theta} (1 - a^n - a^n + a^{-n}) \right] + \\ \Delta i_a \Delta i_b \left[e^{jn\theta} (a^n - a^{-n} + a^n - 1 - a^{-n} + a^n - 1 + a^n) - \\ e^{-jn\theta} (a^{-n} - a^n + a^{-n} - 1 - a^n + a^{-n} - 1 + a^{-n}) \right] + \\ \Delta i_b^2 \left[e^{jn\theta} (a^{-n} - 1 - 1 + a^n) - e^{-jn\theta} (a^n - 1 - 1 + a^{-n}) \right] - \end{bmatrix}$$

$$\begin{aligned}
\Gamma_{réel}'' = \Gamma_{réel} + p \frac{3\sqrt{3}}{2} \sum_{z=1} \sum_{k=1} & \left[\begin{array}{l} \Delta i_a \left[\begin{array}{l} + (6k-2)L_{(6k-2)}i_{(3z-2)} \cos[(6k+3z-4)\theta + \delta_{(3z-2)} + \pi/6] \\ + (6k-4)L_{(6k-4)}i_{(3z-1)} \cos[(6k+3z-5)\theta + \delta_{(3z-1)} - \pi/6] \\ - (6k-4)L_{(6k-4)}i_{(3z-2)} \cos[(6k-3z-2)\theta + \delta_{(3z-2)} - \pi/6] \\ - (6k-2)L_{(6k-2)}i_{(3z-1)} \cos[(6k-3z-1)\theta + \delta_{(3z-1)} + \pi/6] \end{array} \right] + \\ \Delta i_b \left[\begin{array}{l} - (6k-2)L_{(6k-2)}i_{(3z-2)} \sin[(6k+3z-4)\theta + \delta_{(3z-2)}] \\ + (6k-4)L_{(6k-4)}i_{(3z-1)} \sin[(6k+3z-5)\theta + \delta_{(3z-1)}] \\ - (6k-4)L_{(6k-4)}i_{(3z-2)} \sin[(6k-3z-2)\theta + \delta_{(3z-2)}] \\ + (6k-2)L_{(6k-2)}i_{(3z-1)} \sin[(6k-3z-1)\theta + \delta_{(3z-1)}] \end{array} \right] \end{array} \right] + \\ + p \sum_{k=1} & \left[\begin{array}{l} \Delta i_a^2 [(6k-4)L_{(6k-4)} \sin[(6k-4)\theta - 2\pi/3] + (6k-2)L_{(6k-2)} \sin[(6k-2)\theta + 2\pi/3]] - \\ 2\Delta i_a \Delta i_b \left[\begin{array}{l} (6k-4)L_{(6k-4)} \sin[(6k-4)\theta + 2\pi/3] + \\ (6k-2)L_{(6k-2)} \sin[(6k-2)\theta - 2\pi/3] \end{array} \right] + \\ \Delta i_b^2 [(6k-4)L_{(6k-4)} \sin[(6k-4)\theta] + (6k-2)L_{(6k-2)} \sin[(6k-2)\theta]] \end{array} \right]
\end{aligned} \quad (A.43)$$

Dans le cas des harmoniques du couple réductant, nous pouvons constater que les décalages introduisent des harmoniques pairs, exceptés les harmoniques multiples de 3.

A.5 Harmoniques de couple en présence d'une erreur du gain de mesure des courants.

Supposons que nous ayons une mesure des courants avec des erreur de gain comme suit:

$$\begin{aligned}
i_{am} = G_a i_a \quad \text{ou} \quad i_a = G_1 i_{am} \\
i_{bm} = G_b i_b \quad \quad i_b = G_2 i_{bm}
\end{aligned} \quad (A.44)$$

Ainsi le courant statorique peut s'écrire :

$$\begin{aligned}
\mathbf{i}_s = \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} = \sum_{z=1} i_z \begin{bmatrix} G_1 \sin[z\theta + \delta_z] \\ G_2 \sin[z(\theta - 2\pi/3) + \delta_z] \\ -G_1 \sin[z\theta + \delta_z] - G_2 \sin[z(\theta - 2\pi/3) + \delta_z] \end{bmatrix} \\
\mathbf{i}_s = \sum_{z=1} \frac{i_z}{j2} \begin{bmatrix} e^{j(z\theta + \delta_z)} G_1 - e^{-j(z\theta + \delta_z)} G_1 \\ e^{j(z\theta + \delta_z)} G_2 a^{-z} - e^{-j(z\theta + \delta_z)} G_2 a^z \\ e^{j(z\theta + \delta_z)} [-G_1 - G_2 a^{-z}] - e^{-j(z\theta + \delta_z)} [-G_1 - G_2 a^z] \end{bmatrix}
\end{aligned} \quad (A.45)$$

$$i_{3o} = 0, \quad \forall o \in N^+$$

Nous appliquons ces courants aux termes de l'équation du couple (A.7). Ainsi, le couple créé par le flux rotorique devient :

$$\Gamma_{flux} = \mathbf{i}_s^T \frac{d\mathbf{k}(\theta)}{d\theta} p = -p \mathbf{i}_s^T \sum_{\substack{k=1 \\ n=2k-1}} n \frac{k_n}{j2} \begin{bmatrix} e^{jn\theta} - e^{-jn\theta} \\ e^{jn\theta} a^{-n} - e^{-jn\theta} a^n \\ e^{jn\theta} a^{-2n} - e^{-jn\theta} a^{2n} \end{bmatrix}$$

$$\Gamma_{flux} = p \sum_{\substack{k=1 \\ n=2k-1}} \sum_{z=1} n \frac{i_z k_n}{4} \begin{bmatrix} + e^{j[(n+z)\theta+\delta_z]} [G_1(1-a^n) + G_2 a^{-z}(a^{-n} - a^n)] \\ + e^{-j[(n+z)\theta+\delta_z]} [G_1(1-a^{-n}) + G_2 a^z(a^n - a^{-n})] \\ - e^{j[(n-z)\theta-\delta_z]} [G_1(1-a^n) + G_2 a^z(a^{-n} - a^n)] \\ - e^{-j[(n-z)\theta-\delta_z]} [G_1(1-a^{-n}) + G_2 a^{-z}(a^n - a^{-n})] \end{bmatrix} \quad (\text{A.46})$$

$$\Gamma_{flux} = p \sum_{r=1} \sum_{z=1} \frac{i_z}{4} \begin{bmatrix} + (6r-5)k_{(6r-5)} e^{j[(6r-5+z)\theta+\delta_z]} [G_1(1-a) + G_2 a^{-z}(a^{-1} - a)] \\ + (6r-5)k_{(6r-5)} e^{-j[(6r-5+z)\theta+\delta_z]} [G_1(1-a^{-1}) + G_2 a^z(a - a^{-1})] \\ - (6r-5)k_{(6r-5)} e^{j[(6r-5-z)\theta-\delta_z]} [G_1(1-a) + G_2 a^z(a^{-1} - a)] \\ - (6r-5)k_{(6r-5)} e^{-j[(6r-5-z)\theta-\delta_z]} [G_1(1-a^{-1}) + G_2 a^{-z}(a - a^{-1})] \\ + (6r-1)k_{(6r-1)} e^{j[(6r-1+z)\theta+\delta_z]} [G_1(1-a^{-1}) + G_2 a^{-z}(a - a^{-1})] \\ + (6r-1)k_{(6r-1)} e^{-j[(6r-1+z)\theta+\delta_z]} [G_1(1-a) + G_2 a^z(a^{-1} - a)] \\ - (6r-1)k_{(6r-1)} e^{j[(6r-1-z)\theta-\delta_z]} [G_1(1-a^{-1}) + G_2 a^z(a - a^{-1})] \\ - (6r-1)k_{(6r-1)} e^{-j[(6r-1-z)\theta-\delta_z]} [G_1(1-a) + G_2 a^{-z}(a^{-1} - a)] \end{bmatrix} \quad (\text{A.47})$$

$$\Gamma_{flux} = p\sqrt{3} \sum_{r=1} \sum_{z=1} \frac{i_z}{4} \begin{bmatrix} + (6r-5)k_{(6r-5)} e^{j[(6r-5+z)\theta+\delta_z]} \begin{bmatrix} G_1 e^{-j\frac{\pi}{6}} + G_2 e^{-j\left(\frac{2\pi}{3}z+\frac{\pi}{2}\right)} \\ G_1 e^{j\frac{\pi}{6}} + G_2 e^{j\left(\frac{2\pi}{3}z+\frac{\pi}{2}\right)} \end{bmatrix} \\ + (6r-5)k_{(6r-5)} e^{-j[(6r-5+z)\theta+\delta_z]} \begin{bmatrix} G_1 e^{j\frac{\pi}{6}} + G_2 e^{j\left(\frac{2\pi}{3}z+\frac{\pi}{2}\right)} \\ G_1 e^{-j\frac{\pi}{6}} + G_2 e^{j\left(\frac{2\pi}{3}z-\frac{\pi}{2}\right)} \end{bmatrix} \\ - (6r-5)k_{(6r-5)} e^{j[(6r-5-z)\theta-\delta_z]} \begin{bmatrix} G_1 e^{-j\frac{\pi}{6}} + G_2 e^{j\left(\frac{2\pi}{3}z-\frac{\pi}{2}\right)} \\ G_1 e^{j\frac{\pi}{6}} + G_2 e^{-j\left(\frac{2\pi}{3}z-\frac{\pi}{2}\right)} \end{bmatrix} \\ - (6r-5)k_{(6r-5)} e^{-j[(6r-5-z)\theta-\delta_z]} \begin{bmatrix} G_1 e^{j\frac{\pi}{6}} + G_2 e^{-j\left(\frac{2\pi}{3}z-\frac{\pi}{2}\right)} \\ G_1 e^{-j\frac{\pi}{6}} + G_2 e^{j\left(\frac{2\pi}{3}z-\frac{\pi}{2}\right)} \end{bmatrix} \\ + (6r-1)k_{(6r-1)} e^{j[(6r-1+z)\theta+\delta_z]} \begin{bmatrix} G_1 e^{j\frac{\pi}{6}} + G_2 e^{-j\left(\frac{2\pi}{3}z-\frac{\pi}{2}\right)} \\ G_1 e^{-j\frac{\pi}{6}} + G_2 e^{j\left(\frac{2\pi}{3}z-\frac{\pi}{2}\right)} \end{bmatrix} \\ + (6r-1)k_{(6r-1)} e^{-j[(6r-1+z)\theta+\delta_z]} \begin{bmatrix} G_1 e^{-j\frac{\pi}{6}} + G_2 e^{j\left(\frac{2\pi}{3}z-\frac{\pi}{2}\right)} \\ G_1 e^{j\frac{\pi}{6}} + G_2 e^{j\left(\frac{2\pi}{3}z+\frac{\pi}{2}\right)} \end{bmatrix} \\ - (6r-1)k_{(6r-1)} e^{j[(6r-1-z)\theta-\delta_z]} \begin{bmatrix} G_1 e^{j\frac{\pi}{6}} + G_2 e^{j\left(\frac{2\pi}{3}z+\frac{\pi}{2}\right)} \\ G_1 e^{-j\frac{\pi}{6}} + G_2 e^{-j\left(\frac{2\pi}{3}z+\frac{\pi}{2}\right)} \end{bmatrix} \\ - (6r-1)k_{(6r-1)} e^{-j[(6r-1-z)\theta-\delta_z]} \begin{bmatrix} G_1 e^{-j\frac{\pi}{6}} + G_2 e^{-j\left(\frac{2\pi}{3}z+\frac{\pi}{2}\right)} \\ G_1 e^{j\frac{\pi}{6}} + G_2 e^{j\left(\frac{2\pi}{3}z+\frac{\pi}{2}\right)} \end{bmatrix} \end{bmatrix} \quad (\text{A.48})$$

$$\Gamma_{flux} = p \frac{\sqrt{3}}{2} \sum_{r=1} \sum_{z=1} i_z \begin{bmatrix} + (6r-5)k_{(6r-5)}A(-1,-z)\cos[(6r-5+z)\theta + \delta_z + \phi(-1,-z)] \\ - (6r-5)k_{(6r-5)}A(-1,+z)\cos[(6r-5-z)\theta - \delta_z + \phi(-1,+z)] \\ + (6r-1)k_{(6r-1)}A(+1,-z)\cos[(6r-1+z)\theta + \delta_z + \phi(+1,-z)] \\ - (6r-1)k_{(6r-1)}A(+1,+z)\cos[(6r-1-z)\theta - \delta_z + \phi(+1,+z)] \end{bmatrix} \quad (A.49)$$

$$A(s, z) = \left\| G_1 e^{j\frac{\pi}{6}s} + G_2 e^{j\left(\frac{2\pi}{3}z + \frac{\pi}{2}s\right)} \right\| \quad \phi(s, z) = \text{angle} \left(G_1 e^{j\frac{\pi}{6}s} + G_2 e^{j\left(\frac{2\pi}{3}z + \frac{\pi}{2}s\right)} \right)$$

Nous pouvons constater que les erreurs de gain introduisent des harmoniques pairs.

En faisant la même démarche pour le couple réductant :

$$\Gamma_{rel} = p \mathbf{i}_s^T \frac{d\mathbf{L}_s(\theta)}{d\theta} \mathbf{i}_s = p \mathbf{i}_s^T \sum_{k=1} \sum_{z=1} n \frac{L_n}{4} i_z$$

$$\begin{bmatrix} + e^{j[(n+z)\theta + \delta_z]} [G_1(1-a^{-n}) + G_2 a^{-z}(a^n - a^{-n})] + e^{-j[(n+z)\theta + \delta_z]} [G_1(1-a^n) + G_2 a^z(a^{-n} - a^n)] \\ - e^{j[(n-z)\theta - \delta_z]} [G_1(1-a^{-n}) + G_2 a^z(a^n - a^{-n})] - e^{-j[(n-z)\theta - \delta_z]} [G_1(1-a^n) + G_2 a^{-z}(a^{-n} - a^n)] \\ + e^{j[(n+z)\theta + \delta_z]} [G_1(a^n - 1) + G_2 a^{-z}(a^{-n} - 1)] + e^{-j[(n+z)\theta + \delta_z]} [G_1(a^{-n} - 1) + G_2 a^z(a^n - 1)] \\ - e^{j[(n-z)\theta - \delta_z]} [G_1(a^n - 1) + G_2 a^z(a^{-n} - 1)] - e^{-j[(n-z)\theta - \delta_z]} [G_1(a^{-n} - 1) + G_2 a^{-z}(a^n - 1)] \\ + e^{j[(n+z)\theta + \delta_z]} [G_1(a^{-n} - a^n) + G_2 a^{-z}(1 - a^n)] + e^{-j[(n+z)\theta + \delta_z]} [G_1(a^n - a^{-n}) + G_2 a^z(1 - a^{-n})] \\ - e^{j[(n-z)\theta - \delta_z]} [G_1(a^{-n} - a^n) + G_2 a^z(1 - a^n)] - e^{-j[(n-z)\theta - \delta_z]} [G_1(a^n - a^{-n}) + G_2 a^{-z}(1 - a^{-n})] \end{bmatrix} \quad (A.50)$$

$$\Gamma_{rel} = p \frac{1}{4} \sum_{m=1} \sum_{k=1} \sum_{z=1} \frac{1}{j2} n L_n i_m i_z$$

$$\begin{bmatrix} + e^{j[(n+m+z)\theta + \delta_m + \delta_z]} G_1 [G_1(1-a^{-n}) + G_2 a^{-z}(a^n - a^{-n})] - e^{-j[(n+m+z)\theta + \delta_m + \delta_z]} G_1 [G_1(1-a^n) + G_2 a^z(a^{-n} - a^n)] \\ - e^{j[(n+m-z)\theta + \delta_m - \delta_z]} G_1 [G_1(1-a^{-n}) + G_2 a^z(a^n - a^{-n})] + e^{-j[(n+m-z)\theta + \delta_m - \delta_z]} G_1 [G_1(1-a^n) + G_2 a^{-z}(a^{-n} - a^n)] \\ - e^{j[(n-m+z)\theta - \delta_m + \delta_z]} G_1 [G_1(1-a^{-n}) + G_2 a^{-z}(a^n - a^{-n})] + e^{-j[(n-m+z)\theta - \delta_m + \delta_z]} G_1 [G_1(1-a^n) + G_2 a^z(a^{-n} - a^n)] \\ + e^{j[(n-m-z)\theta - \delta_m - \delta_z]} G_1 [G_1(1-a^{-n}) + G_2 a^z(a^n - a^{-n})] - e^{-j[(n-m-z)\theta - \delta_m - \delta_z]} G_1 [G_1(1-a^n) + G_2 a^{-z}(a^{-n} - a^n)] \\ + e^{j[(n+m+z)\theta + \delta_m + \delta_z]} G_2 a^{-m} [G_1(a^n - 1) + G_2 a^{-z}(a^{-n} - 1)] - e^{-j[(n+m+z)\theta + \delta_m + \delta_z]} G_2 a^m [G_1(a^{-n} - 1) + G_2 a^z(a^n - 1)] \\ - e^{j[(n+m-z)\theta + \delta_m - \delta_z]} G_2 a^{-m} [G_1(a^n - 1) + G_2 a^z(a^{-n} - 1)] + e^{-j[(n+m-z)\theta + \delta_m - \delta_z]} G_2 a^m [G_1(a^{-n} - 1) + G_2 a^{-z}(a^n - 1)] \\ - e^{j[(n-m+z)\theta - \delta_m + \delta_z]} G_2 a^m [G_1(a^n - 1) + G_2 a^{-z}(a^{-n} - 1)] + e^{-j[(n-m+z)\theta - \delta_m + \delta_z]} G_2 a^{-m} [G_1(a^{-n} - 1) + G_2 a^z(a^n - 1)] \\ + e^{j[(n-m-z)\theta - \delta_m - \delta_z]} G_2 a^m [G_1(a^n - 1) + G_2 a^z(a^{-n} - 1)] - e^{-j[(n-m-z)\theta - \delta_m - \delta_z]} G_2 a^{-m} [G_1(a^{-n} - 1) + G_2 a^{-z}(a^n - 1)] \\ + e^{j[(n+m+z)\theta + \delta_m + \delta_z]} [-G_1 - G_2 a^{-m}] [G_1(a^{-n} - a^n) + G_2 a^{-z}(1 - a^n)] - e^{-j[(n+m+z)\theta + \delta_m + \delta_z]} [-G_1 - G_2 a^m] [G_1(a^n - a^{-n}) + G_2 a^z(1 - a^{-n})] \\ - e^{j[(n+m-z)\theta + \delta_m - \delta_z]} [-G_1 - G_2 a^{-m}] [G_1(a^{-n} - a^n) + G_2 a^z(1 - a^n)] + e^{-j[(n+m-z)\theta + \delta_m - \delta_z]} [-G_1 - G_2 a^m] [G_1(a^n - a^{-n}) + G_2 a^{-z}(1 - a^{-n})] \\ - e^{j[(n-m+z)\theta - \delta_m + \delta_z]} [-G_1 - G_2 a^m] [G_1(a^{-n} - a^n) + G_2 a^{-z}(1 - a^n)] + e^{-j[(n-m+z)\theta - \delta_m + \delta_z]} [-G_1 - G_2 a^{-m}] [G_1(a^n - a^{-n}) + G_2 a^z(1 - a^{-n})] \\ + e^{j[(n-m-z)\theta - \delta_m - \delta_z]} [-G_1 - G_2 a^m] [G_1(a^{-n} - a^n) + G_2 a^z(1 - a^n)] - e^{-j[(n-m-z)\theta - \delta_m - \delta_z]} [-G_1 - G_2 a^{-m}] [G_1(a^n - a^{-n}) + G_2 a^{-z}(1 - a^{-n})] \end{bmatrix} \quad (A.51)$$

$$\Gamma_{rel} = p \frac{1}{4} \sum_{m=1} \sum_{k=1} \sum_{z=1} \frac{1}{j2} n L_n i_m i_z$$

$$\left[\begin{aligned} &+ e^{j[(n+m+z)\theta + \delta_m + \delta_z]} \left[G_1^2 (1 - 2a^{-n} + a^n) + G_1 G_2 (a^{-z} + a^{-m}) (2a^n - a^{-n} - 1) + G_2^2 a^{-z} a^{-m} (a^{-n} - 2 + a^n) \right] \\ &- e^{-j[(n+m+z)\theta + \delta_m + \delta_z]} \left[G_1^2 (1 - 2a^n + a^{-n}) + G_1 G_2 (a^z + a^m) (2a^{-n} - a^n - 1) + G_2^2 a^z a^m (a^n - 2 + a^{-n}) \right] \\ &- e^{j[(n+m-z)\theta + \delta_m - \delta_z]} \left[G_1^2 (1 - 2a^{-n} + a^n) + G_1 G_2 (a^z + a^{-m}) (2a^n - a^{-n} - 1) + G_2^2 a^z a^{-m} (a^{-n} - 2 + a^n) \right] \\ &+ e^{-j[(n+m-z)\theta + \delta_m - \delta_z]} \left[G_1^2 (1 - 2a^n + a^{-n}) + G_1 G_2 (a^{-z} + a^m) (2a^{-n} - a^n - 1) + G_2^2 a^{-z} a^m (a^n - 2 + a^{-n}) \right] \\ &- e^{j[(n-m+z)\theta - \delta_m + \delta_z]} \left[G_1^2 (1 - 2a^{-n} + a^n) + G_1 G_2 (a^{-z} + a^m) (2a^n - a^{-n} - 1) + G_2^2 a^{-z} a^m (a^{-n} - 2 + a^n) \right] \\ &+ e^{-j[(n-m+z)\theta - \delta_m + \delta_z]} \left[G_1^2 (1 - 2a^n + a^{-n}) + G_1 G_2 (a^z + a^{-m}) (2a^{-n} - a^n - 1) + G_2^2 a^z a^{-m} (a^n - 2 + a^{-n}) \right] \\ &+ e^{j[(n-m-z)\theta - \delta_m - \delta_z]} \left[G_1^2 (1 - 2a^{-n} + a^n) + G_1 G_2 (a^z + a^m) (2a^n - a^{-n} - 1) + G_2^2 a^z a^m (a^{-n} - 2 + a^n) \right] \\ &- e^{-j[(n-m-z)\theta - \delta_m - \delta_z]} \left[G_1^2 (1 - 2a^n + a^{-n}) + G_1 G_2 (a^{-z} + a^{-m}) (2a^{-n} - a^n - 1) + G_2^2 a^{-z} a^{-m} (a^n - 2 + a^{-n}) \right] \end{aligned} \right] \quad (A.52)$$

$$\Gamma_{rel} = p \frac{3}{4} \sum_{m=1} \sum_{r=1} \sum_{z=1} \frac{1}{j2} i_m i_z$$

$$\left[\begin{aligned} &+ (6r-4)L_{(6r-4)} e^{j[(6r-4+m+z)\theta + \delta_m + \delta_z]} \left[-G_1^2 a^1 + G_1 G_2 (a^{-z} + a^{-m}) a^{-1} - G_2^2 a^{-z} a^{-m} \right] \\ &- (6r-4)L_{(6r-4)} e^{-j[(6r-2+m+z)\theta + \delta_m + \delta_z]} \left[-G_1^2 a^{-1} + G_1 G_2 (a^z + a^m) a^1 - G_2^2 a^z a^m \right] \\ &- (6r-4)L_{(6r-4)} e^{j[(6r-4+m-z)\theta + \delta_m - \delta_z]} \left[-G_1^2 a^1 + G_1 G_2 (a^z + a^{-m}) a^{-1} - G_2^2 a^z a^{-m} \right] \\ &+ (6r-4)L_{(6r-4)} e^{-j[(6r-4+m-z)\theta + \delta_m - \delta_z]} \left[-G_1^2 a^{-1} + G_1 G_2 (a^{-z} + a^m) a^1 - G_2^2 a^{-z} a^m \right] \\ &- (6r-4)L_{(6r-4)} e^{j[(6r-4-m+z)\theta - \delta_m + \delta_z]} \left[-G_1^2 a^1 + G_1 G_2 (a^{-z} + a^m) a^{-1} - G_2^2 a^{-z} a^m \right] \\ &+ (6r-4)L_{(6r-4)} e^{-j[(6r-4-m+z)\theta - \delta_m + \delta_z]} \left[-G_1^2 a^{-1} + G_1 G_2 (a^z + a^{-m}) a^1 - G_2^2 a^z a^{-m} \right] \\ &+ (6r-4)L_{(6r-4)} e^{j[(6r-4-m-z)\theta - \delta_m - \delta_z]} \left[-G_1^2 a^1 + G_1 G_2 (a^z + a^m) a^{-1} - G_2^2 a^z a^m \right] \\ &- (6r-4)L_{(6r-4)} e^{-j[(6r-4-m-z)\theta - \delta_m - \delta_z]} \left[-G_1^2 a^{-1} + G_1 G_2 (a^{-z} + a^{-m}) a^1 - G_2^2 a^{-z} a^{-m} \right] \\ &+ (6r-2)L_{(6r-2)} e^{j[(6r-2+m+z)\theta + \delta_m + \delta_z]} \left[-G_1^2 a^{-1} + G_1 G_2 (a^{-z} + a^{-m}) a^1 - G_2^2 a^{-z} a^{-m} \right] \\ &- (6r-2)L_{(6r-2)} e^{-j[(6r-2+m+z)\theta + \delta_m + \delta_z]} \left[-G_1^2 a^1 + G_1 G_2 (a^z + a^m) a^{-1} - G_2^2 a^z a^m \right] \\ &- (6r-2)L_{(6r-2)} e^{j[(6r-2+m-z)\theta + \delta_m - \delta_z]} \left[-G_1^2 a^{-1} + G_1 G_2 (a^z + a^{-m}) a^1 - G_2^2 a^z a^{-m} \right] \\ &+ (6r-2)L_{(6r-2)} e^{-j[(6r-2+m-z)\theta + \delta_m - \delta_z]} \left[-G_1^2 a^1 + G_1 G_2 (a^{-z} + a^m) a^{-1} - G_2^2 a^{-z} a^m \right] \\ &- (6r-2)L_{(6r-2)} e^{j[(6r-2-m+z)\theta - \delta_m + \delta_z]} \left[-G_1^2 a^{-1} + G_1 G_2 (a^{-z} + a^m) a^1 - G_2^2 a^{-z} a^m \right] \\ &+ (6r-2)L_{(6r-2)} e^{-j[(6r-2-m+z)\theta - \delta_m + \delta_z]} \left[-G_1^2 a^1 + G_1 G_2 (a^z + a^{-m}) a^{-1} - G_2^2 a^z a^{-m} \right] \\ &+ (6r-2)L_{(6r-2)} e^{j[(6r-2-m-z)\theta - \delta_m - \delta_z]} \left[-G_1^2 a^{-1} + G_1 G_2 (a^z + a^m) a^1 - G_2^2 a^z a^m \right] \\ &- (6r-2)L_{(6r-2)} e^{-j[(6r-2-m-z)\theta - \delta_m - \delta_z]} \left[-G_1^2 a^1 + G_1 G_2 (a^{-z} + a^{-m}) a^{-1} - G_2^2 a^{-z} a^{-m} \right] \end{aligned} \right] \quad (A.53)$$

$$\left[G_1 G_2 (a^z + a^m) a^{-s} - G_1^2 a^s - G_2^2 a^z a^m \right] = - (G_1 a^{-s} - G_2 a^z) (G_1 a^{-s} - G_2 a^m)$$

$$\begin{aligned}
\Gamma_{rel} &= p \frac{3}{4} \sum_{m=1} \sum_{r=1} \sum_{z=1} i_m i_z \\
&\left[\begin{aligned}
&+ (6r-4)L_{(6p-4)}A(1,-m,-z)\sin[(6r-4+m+z)\theta + \delta_m + \delta_z + \phi(1,-m,-z)] \\
&- (6r-4)L_{(6p-4)}A(1,-m,z)\sin[(6r-4+m-z)\theta + \delta_m - \delta_z + \phi(1,-m,z)] \\
&- (6r-4)L_{(6p-4)}A(1,m,-z)\sin[(6r-4-m+z)\theta - \delta_m + \delta_z + \phi(1,m,-z)] \\
&+ (6r-4)L_{(6p-4)}A(1,m,z)\sin[(6r-4-m-z)\theta - \delta_m - \delta_z + \phi(1,m,z)] \\
&+ (6r-2)L_{(6r-2)}A(-1,-m,-z)\sin[(6r-2+m+z)\theta + \delta_m + \delta_z + \phi(1,-m,-z)] \\
&- (6r-2)L_{(6r-2)}A(-1,-m,z)\sin[(6r-2+m-z)\theta + \delta_m - \delta_z + \phi(1,-m,z)] \\
&- (6r-2)L_{(6r-2)}A(-1,m,-z)\sin[(6r-2-m+z)\theta - \delta_m + \delta_z + \phi(1,m,-z)] \\
&+ (6r-2)L_{(6r-2)}A(-1,m,z)\sin[(6r-2-m-z)\theta - \delta_m - \delta_z + \phi(1,m,z)]
\end{aligned} \right] \quad (A.54)
\end{aligned}$$

$$A(s, m, z) = \left\| G_1 e^{-j\frac{2\pi}{3}s} - G_2 e^{j\frac{2\pi}{3}z} \right\| \cdot \left\| G_1 e^{-j\frac{2\pi}{3}s} - G_2 e^{j\frac{2\pi}{3}m} \right\|$$

$$\phi(s, m, z) = \text{angle} \left(G_1 e^{-j\frac{2\pi}{3}s} - G_2 e^{j\frac{2\pi}{3}z} \right) + \text{angle} \left(G_1 e^{-j\frac{2\pi}{3}s} - G_2 e^{j\frac{2\pi}{3}m} \right) + \pi$$

Nous pouvons observer que l'erreur de gain introduit des harmoniques à toutes les fréquences multiples de la fréquence fondamentale.

Annexe B: La rétropropagation.

Le noyau de toute méthode de rétropropagation est l'application de la règle d'enchaînement des dérivées partielles ordonnées pour calculer la sensibilité d'une fonction de coût par rapport aux paramètres du réseau. En d'autres termes, la rétropropagation est employée pour impliquer un passage en arrière de l'erreur à chaque nœud interne dans le réseau, qui est alors utilisé pour calculer le gradient des coefficients de pondération pour chaque nœud. L'apprentissage progresse en propageant alternativement en avant les signaux d'entrée du réseau et en propageant en arrière les erreurs instantanées.

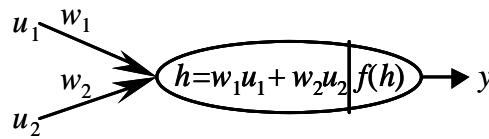


Figure B.1 : Sortie d'un neurone.

La figure B.1 montre la sortie d'un neurone. La relation entre la valeur de sortie y et la valeur de l'entrée totale h étant la fonction $f()$, la sensibilité de la sortie par rapport à l'entrée totale du neurone h est obtenue en linéarisant cette fonction :

$$\Delta y = \frac{\partial f}{\partial h} \Delta h \quad (\text{B.1})$$

De même, nous pouvons dire que la sensibilité de l'entrée totale est dépendant des variations des coefficients de pondération w_1 et w_2 :

$$\Delta h = \frac{\partial h}{\partial w_1} \Delta w_1 + \frac{\partial h}{\partial w_2} \Delta w_2 = u_1 \Delta w_1 + u_2 \Delta w_2 \quad (\text{B.2})$$

Ainsi, la variation de la sortie par rapport aux coefficients de pondération est :

$$\Delta y = \frac{\partial f}{\partial h} u_1 \Delta w_1 + \frac{\partial f}{\partial h} u_2 \Delta w_2 \quad (\text{B.3})$$

$$\frac{\Delta y}{\Delta w_1} = \frac{\partial f}{\partial h} u_1 \text{ et } \frac{\Delta y}{\Delta w_2} = \frac{\partial f}{\partial h} u_2 \quad (\text{B.4})$$

Avec ces sensibilités nous pouvons observer comment nous devons changer les coefficients de pondération pour réduire ou augmenter la valeur de sortie.

Nous pouvons obtenir ce même résultat en calculant le différentiel total de la sortie :

$$y = f(w_1 u_1 + w_2 u_2) \rightarrow dy = \frac{\partial f}{\partial w_1} dw_1 + \frac{\partial f}{\partial w_2} dw_2 \quad (\text{B.5})$$

$$\frac{\partial f}{\partial w_1} = \frac{\partial f}{\partial h} \frac{\partial h}{\partial w_1} = \frac{\partial f}{\partial h} u_1 \text{ et } \frac{\partial f}{\partial w_2} = \frac{\partial f}{\partial h} \frac{\partial h}{\partial w_2} = \frac{\partial f}{\partial h} u_2 \quad (\text{B.6})$$

Dans les cas où nous avons plusieurs couches, les entrées du neurone u_1 et u_2 , sont des sortie d'un neurone d'autre couche et suivant le raisonnement donné par l'équation (B.5), nous pouvons obtenir la sensibilité de la sortie par rapport aux l'entrées u_1 et u_2 :

$$y = f(w_1 u_1 + w_2 u_2) \rightarrow dy = \frac{\partial f}{\partial w_1} dw_1 + \frac{\partial f}{\partial w_2} dw_2 + \frac{\partial f}{\partial u_1} du_1 + \frac{\partial f}{\partial u_2} du_2 \quad (\text{B.7})$$

$$\frac{\partial f}{\partial u_1} = \frac{\partial f}{\partial h} \frac{\partial h}{\partial u_1} = \frac{\partial f}{\partial h} w_1 \text{ et } \frac{\partial f}{\partial u_2} = \frac{\partial f}{\partial h} \frac{\partial h}{\partial u_2} = \frac{\partial f}{\partial h} w_2 \quad (\text{B.8})$$

Annexe C: Table de figures.

Figure I.1 : Erreur de poursuite dans le mouvement en spirale carrée sur les axes X et Y.	9
Figure I.2 : Détail de la trajectoire de la figure I.1.	10
Figure I.3 : Représentation d'un neurone réel.	12
Figure I.4 : Modèle général d'un neurone.	13
Figure I.5 : Exemple d'un réseau de neurones avec une structure en couches.	14
Figure I.6 : Exemple d'un réseau de neurones récurrent.	14
Figure I.7 : Exemple d'un réseau de neurones en couches avec une récurrence entre couches.	14
Figure 1.1 : Modèle d'un axe de machine-outil.	21
Figure 1.2 : Modèle mécanique simplifié d'un axe de machine-outil avec une résonance.	22
Figure 1.3 : Modèle mécanique simplifié d'un axe de machine-outil avec deux résonances.	22
Figure 1.4 : Diagramme de Bode du modèle complet et simplifié montrant l'approche de la réduction de l'ordre du modèle mécanique.	23
Figure 1.5 : Amplitude et fréquence de résonance en fonction de la position de l'ensemble vis/écrou.	24
Figure 1.6 : Représentation de l'effet global des frottements.	24
Figure 1.7 : Représentation de l'effet des jeux.	25
Figure 1.8 : Amplitude et fréquence de résonance en fonction de la position de l'ensemble vis/écrou.	25
Figure 1.9 : Evolution de la force de coupe sur un axe dans l'usinage.	26
Figure 1.10 : Allure de l'inductance statorique pour une machine synchrone à pôles enterrés.	28
Figure 1.11 : Harmoniques de couple en présence des harmoniques de l'inductance statorique ($F_s=1,41\text{Hz}$).	29
Figure 1.12 : Harmoniques de couple en présence des harmoniques de FEM ($F_s=1,41\text{Hz}$).	30
Figure 1.13 : Harmoniques de couple causés par le couple de crantage ($F_s=1,41\text{Hz}$, $N_e/p=10$).	31
Figure 1.14 : Filtre de bus et résistance de freinage.	32
Figure 1.15 : Structure de l'onduleur de tension.	32
Figure 1.16 : Modulation des références de tension avec la porteuse de MLI.	33
Figure 1.17 : Structure de la commande vectorielle de la machine synchrone.	33
Figure 1.18 : Compensation des variations de la tension de Bus.	34
Figure 1.19 : Allure de la tension perturbatrice moyenne introduite par les temps de garde de la MLI.	36
Figure 1.20 : Harmoniques de courant causés par les temps de garde en boucle ouverte ($F_s=88,07\text{Hz}$, $F_{MLI}=8\text{kHz}$).	36
Figure 1.21 : Harmoniques de courant causés par les temps de garde en boucle fermée ($F_s=88,07\text{Hz}$, $F_{MLI}=8\text{kHz}$).	36
Figure 1.22 : Harmoniques de courant causés par la MLI en boucle ouverte ($F_s=88,07\text{Hz}$, $F_{MLI}=8000\text{Hz}$).	37
Figure 1.23 : Harmoniques de courant causés par la MLI en boucle fermée ($F_s=88,07\text{Hz}$, $F_{MLI}=8000\text{Hz}$).	37
Figure 1.24 : Harmoniques de couple causés par la tension de bus ($F_s=26,96\text{Hz}$, $F_{UBUS}=300\text{Hz}$).	39
Figure 1.25 : Modèles des capteurs de mesure du courant, de la position et la vitesse.	40
Figure 1.26 : Effet du synchronisme dans la mesure des courants.	40
Figure 1.27 : Harmoniques du couple en mesure synchrone et non synchrone à vitesse constant ($I_1=5\text{A}$, $F_s=88,07\text{Hz}$).	41

Figure 1.28: Représentation vectorielle des courants de la machine et les courants de la commande en présence d'un décalage sur le courant i_a	42
Figure 1.29: Harmoniques du couple avec un décalage de $\Delta i_a=0,1A$ ($I_1=5A$, $F_s=88,07Hz$).....	43
Figure 1.30: Représentation vectorielle des courants de la machine et de la commande en présence des différences de gain.....	43
Figure 1.31: Harmoniques du couple avec un gain de $G_a=1,1$ ($I_1=5A$, $F_s=88,07Hz$).	44
Figure 1.32: Erreur de discrétisation en fonction du signal original.	45
Figure 1.33: Harmoniques du couple sans discrétisation et avec discrétisation à vitesse constant ($I_1=5A$, $F_s=88,07Hz$).	46
Figure 1.34: Schéma de correction des références de tension.....	47
Figure 1.35: Harmonique H_3 et référence de tension modifiée β_{al} pour la tension maximale de sortie.....	47
Figure 1.36: Harmoniques de couple en la limitation de tension ($F_s=133,07Hz$).	47
Figure 2.1: Représentation d'un neurone réel.	52
Figure 2.2: Modèle général d'un neurone.	53
Figure 2.3: Fonctions de sortie des neurones.	54
Figure 2.4: Exemple d'un réseau de neurones avec une structure en couches.....	55
Figure 2.5: Exemple d'un réseau de neurones récurrent.	55
Figure 2.6: Exemple d'un réseau de neurones en couches avec une récurrence entre couches.	56
Figure 2.7: Sortie d'un réseau de neurones à base radiale avec deux entrées (X,Y), 4 neurones à base radiale et une sortie (Z).	58
Figure 2.8: Représentation des paramètres d'une couche du réseau des neurones.	60
Figure 2.9: Résultats d'un apprentissage d'un sinus.	66
Figure 2.10: Sur-apprentissage dû au nombre trop élevé de paramètres.....	67
Figure 2.11: Algorithme d'optimisation de Levenberg-Marquardt.....	73
Figure 2.12: Diagramme de temps de l'apprentissage en ligne et l'apprentissage différé.	75
Figure 2.13: Principe de l'estimation des paramètres d'un modèle.	78
Figure 2.14: Structure équivalente à un système d'état discret.	79
Figure 2.15: Structure équivalente à une fonction de transfert discrète.	79
Figure 2.16: Modèle série-parallèle d'identification.	80
Figure 2.17: Modèle parallèle d'identification.....	80
Figure 2.18: Signaux d'excitation pour l'identification.	83
Figure 2.19: Validation de l'identification avec un modèle linéaire récurrent.....	89
Figure 2.20: Données d'apprentissage et modèle résultant.	91
Figure 2.21: Validation du modèle trouvé de la figure 2.20 avec une entrée en forme des dents de scie.....	91
Figure 2.22: Validation de l'identification du système linéaire avec un modèle linéaire récurrent.	93
Figure 2.23: Tests de corrélation de l'erreur de prédiction de la figure 2.22b.	94
Figure 2.24: Modèle des bruits de mesure.	94
Figure 2.25: Modèle des bruit colorés et des perturbations.	94
Figure 2.26: Tests de corrélation de l'erreur de prédiction d'une structure OE quand le bruit affectant le système est coloré.....	95
Figure 2.27: Validation du modèle linéaire identifié avec les données d'apprentissage (1 ^{er} étape).	96

Figure 2.28: Validation du modèle linéaire identifié avec un autre jeu de données sans bruit (2 ^{ème} étape)	96
Figure 3.1: Modèle de simulation du servomécanisme entier	102
Figure 3.2: Modèle de simulation de l'axe et contrôle des courants	103
Figure 3.3: Fenêtre de dialogue pour l'entrée des paramètres du modèle d'un axe mécanique rigide.....	105
Figure 3.4: Banc d'essais des algorithmes	107
Figure 3.5: Structure interne de la carte DSP DS1102 de dSPACE.....	108
Figure 3.6: Diagramme d'interconnexion du banc d'essais des moteurs	109
Figure 3.7: Spectre de la tension de phase	110
Figure 3.8: Couple mesuré et estimé et les harmoniques de couple mesuré	112
Figure 3.9: Comparaison de l'allure de la vitesse et du couple.....	113
Figure 3.10: Réponses fréquentielles du boucle de courant.....	114
Figure 3.11: Différence entre les réponses fréquentielles montrés dans la figure 3.9.....	115
Figure 3.12: Machine-outil ANAYAK utilisé dans les essais expérimentaux	116
Figure 3.13: Diagramme d'interconnexion du banc d'essais de la machine-outil	117
Figure 3.14: Réponses fréquentielles de l'axe X à partir des essais et modèle identifié	119
Figure 3.15: Réponse fréquentielle estimée de la fonction de transfert entre le couple de référence et le couple mesuré	121
Figure 3.16: Réponse fréquentielle estimée à partir des données de simulation de la fonction de transfert entre le couple de référence et le couple estimé	121
Figure 3.17: Réponses fréquentielles estimés à partir des données expérimentales et de simulation	122
Figure 3.18: Harmoniques de couple avant la précompensation de couple (a) et après (b)	124
Figure 3.19: Validation du modèle identifié en prédiction: réponses temporelles	125
Figure 3.20: Validation du modèle identifié en prédiction: corrélations et marges de confiance	125
Figure B.1 : Sortie d'un neurone.....	153

