

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



Université Dr. Tahar Moulay de Saïda  
Faculté de la Technologie  
Département d'Electrotechnique



## Mémoire de Fin d'Etudes

En vue de l'obtention du diplôme de

Master (LMD)

Spécialité : ELECTROTECHNIQUE INDUSTRIELLE

Filière : ELECTROTECHNIQUE

Intitulé :

### **Modélisation d'un contrôleur PI-PSO pour la régulation de la vitesse d'un moteur asynchrone**

Présenté par :

BELOUFA NAIMA

ARDJANI IKRAM

Devant le jury composé de :

Pr. MILOUD Yahia  
Pr. MILOUDI Abdallah  
Dr. CHERIFI Djamila

Président  
Encadreur  
Examineur

Soutenu le ...../...../2020  
Promotion 2019-2020

## Résumé

---

# Modélisation d'un contrôleur PI-PSO pour la régulation de la vitesse d'un moteur asynchrone

## Résumé

Depuis son invention, le moteur asynchrone est devenu grâce à son faible coût, sa robustesse et sa fiabilité le moteur électrique le plus utilisé dans l'industrie. Cependant son utilisation à des vitesses variables a nécessité le développement de stratégies de commandes permettant une alimentation triphasée à fréquence et amplitude variable. Les stratégies de commande vectorielle et directe du couple qui effectuent un découplage entre le flux et le couple permettent une commande performante de la machine asynchrone.

La régulation de la vitesse qui offre l'adaptation de l'opération de découplage en cas de variation de charge ou des paramètres de la machine est une partie importante dans la stratégie de commande vectorielle et nécessite l'utilisation de régulateurs robustes. Le régulateur PI classique nécessite la connaissance de valeurs optimales des gains des actions proportionnel et intégrateur et trouve des difficultés à réagir efficacement aux variations des paramètres de la machine.

La méthode d'optimisation par essaim de particule PSO offre une possibilité d'optimiser les paramètres du régulateur PI classique afin d'améliorer les performances du régulateur et de réagir efficacement aux variations des paramètres de la machine.

Dans ce mémoire nous présentons une étude et une conception de régulateurs PI-PSO pour la régulation de la vitesse d'un moteur asynchrone utilisant la stratégie de commande vectorielle. Ce régulateur est obtenu à partir du régulateur PI classique dont les paramètres sont optimisés par l'algorithme PSO afin de minimiser une fonction objective ayant une relation directe avec l'erreur de vitesse. Deux régulateurs PI-PSO ont été conçus, le premier a été obtenu off – line et le second a été obtenu on – line.

Des travaux de simulations ont montrés que le régulateur PI-PSO peut éliminer le dépassement sans dégrader la réaction face aux perturbations de charge et donne ainsi de meilleures performances que le régulateur PI par placement des pôles.

# *REMERCIEMENTS*

Je remercie Dieu le tout puissant de m'avoir donné courage et patience pour mener à bien ce travail, qu'il soit béni et glorifié.

Je remercie chaleureusement Pr. Miloudi Abdallah, professeur à la Faculté de technologie de l'Université Dr Moulay Tahar de Saïda pour avoir accepté de diriger ce mémoire de fin d'étude. Ses contributions et observations m'ont permis d'achever au mieux ce travail.

Je remercie mes formidables parents qui m'ont toujours soutenue,

J'adresse mes remerciements les plus sincères au Pr. Miloud Yahia, Professeur à l'Université de Saïda, pour avoir accepté de présider le jury.

Je voudrais remercier vivement le membre du jury: Dr. Cherifi Djamila, Maître de conférences à l'université de Saïda, pour l'honneur qu'elle m'a fait par sa participation à l'évaluation scientifique de ce travail en tant qu'examineur.

Un grand merci également à tous mes amis, ainsi qu'à tous mes professeurs, chercheurs et tout le personnel de la Faculté des sciences et technologies de l'Université Dr Moulay Tahar, heureux de leur sympathie, de leur bon traitement et de leur coopération avec nous dans ce travail.

Enfin, il me serait impossible de terminer ces remerciements sans adresser une pensée chaleureuse à toutes les personnes qui m'ont aidé de près ou de loin pour terminer ce travail.

## *Dédicaces*

Toutes les lettres ne trouvent pas les bons mots ...

Tous les mots ne peuvent pas exprimer de gratitude,

Amour, respect et reconnaissance ...

De plus, ce n'est que:

Je dédie ce rapport:

À qui m'a fait le plus beau cadeau de ma vie, ma grand-mère

À ceux qui ont bu une tasse vide pour me donner des gouttes d'amour

A qui je complète ses doigts pour nous donner un moment de bonheur

À celui qui a récolté les épines de mon chemin pour ouvrir la voie pour que je sache, c'est mon grand père de cœur, Mohammad

A ceux qui m'ont fait l'amour et la tendresse

Au symbole de l'amour et du baume guérisseur

Au cœur blanc pur, ma mère bien-aimée, N,Bouazza

Aux cœurs purs et aux âmes innocentes aux tissus fins de la vie de mes frères.

À l'âme qui habitait mon âme Naima

Maintenant, les voiles s'ouvrent et l'ancre est levée pour lancer le navire à travers la vaste mer sombre, qui est la mer de la vie, et dans cette obscurité que seules les méduses des souvenirs brillent, les souvenirs des frères lointains à ceux que j'aimais et aimais mes amis et tous les membres de la famille Ardjani.

*Ardjani Ikram*

## ***Dédicaces***

*Je dédie ce modeste travail à mon père qui ma tellement  
Soutenu durant ma vie et ses sacrifices il est l'espoir de mon  
Existence pour réussir dans mes études.*

*A ma très chère mère pour le soutien moral et affectif*

*A*

*Mes frères Khalfalah et Imed*

*Ma sœur Maroua.*

*A*

*Toute ma famille(Beloufa)*

*A*

*Tous mes amis.*

*A tous mes enseignants qui ont fait leurs possibles pour me donner le maximum  
d'information concernant mes études.*

*A tous mes collègues de l'université surtout la promotion de l'informatique.*

*A toutes les personnes qui nous ont aidés de près ou de loin pour la réalisation  
de ce mémoire.*

***Beloufa Naima***

## Liste des symboles

Symbole	Signification	Unité
$N_p$	Nombre de particules de l'essaim.	
$N_d$	Nombre de variables du problème	
$K_{max}$	Nombre maximal d'itération.	
$v_{ij}(k)$	Vitesse de la $j^{ème}$ composante de la $i^{ème}$ particule de l'essaim, à la $k^{ème}$ Itération.	$m/s$
$p_{ij}$	$j^{ème}$ Composante de la meilleure position occupée par la $i^{ème}$ particule de l'essaim enregistrée dans les itérations précédentes (local best).	
$g_{ij}$	$j^{ème}$ composante de la meilleure position occupée par la $i^{ème}$ particule globale de l'essaim (global best).	
$x_{ij}(k)$	$j^{ème}$ Coordonnée de la position actuelle de particule $i$ a la $k^{ème}$ .	
$w$	Coefficient d'inertie.	
$c_1, c_2$	Coefficients d'accélération.	
$r_1, r_2$	Nombres aléatoires entre [0 ,1]	
$v_{as}, v_{bs}, v_{cs}$	Tensions statoriques triphasées	V
$v_{ds}, v_{qs}$	Tensions statoriques diphasées dans le repère	V
$i_{as}, i_{bs}, i_{cs}$	Courant statoriques triphasées	A
$L_s$	Inductance cyclique statorique	H
$L_r$	Inductance cyclique rotorique	H
$L_m$	Inductance mutuelle cyclique entre stator et rotor	H
$\Omega$	La vitesse mécanique	$rad/s$
$d, q$	Axes correspondants au référentiel lié au champ tournant	

## Liste des symboles

Symbole	Signification	Unité
$K_p, K_i$	Gains des régulateurs PI	
$R_s, R_r$	Résistances d'enroulements statorique et rotorique par phase	$\Omega$
$\sigma$	Coefficient de dispersion	
$\varphi_r$	Flux rotorique	$Wb$
$\varphi_s$	Flux statorique	$Wb$
$J$	Moment d'inertie de la partie tournant	$kg.m^2$
$f_v$	Coefficient de frottement visqueux.	$kg.m^2/s$
$C_r$	Couple résistant.	$N.m$
$C_e$	Couple électromagnétique.	$N.m$
$f_s$	Fréquence statorique	$Hz$
$p$	Nombre de paire de pôles	
$N_s$	Vitesse de rotation en trs/s du champ tournant statorique	trs/s
$\Omega_m$	La vitesse de rotation de la machine. .	$rad/s$
$\Omega_s$	La vitesse de statorique de la machine.	$rad/s$
$g$	Glissement du moteur asynchrone	
$\theta_s, \theta_{sl}, \theta$	Angles électriques statorique, rotorique, et du glissement	$rad$
$\omega, \omega_s, \omega_r$	Pulsations statorique, rotorique, et de glissement	$rad/s$
MLI	Modulation de largeur d'impulsion	
PWM	Pulse Width Modulation	
FOC	Commande Vectorielle à Flux Orienté	
IRFOC	Commande Vectorielle Indirecte à Flux Rotorique Orienté	

## Liste des symboles

---

Symbole	Signification	Unité
MAS	Machine Asynchrone	
PI	Régulateur proportionnel intégrale	

# Liste des figures

---

## Chapitre 1

<b>Figure 1.1.</b> Présentation d'une machine asynchrone.....	4
<b>Figure 1.2.</b> Structure d'un rotor bobiné.....	6
<b>Figure 1.3.</b> Structure d'un Rotor en cage d'écureuil.....	6
<b>Figure 1.4.</b> Rotor à cage d'écureuil en aluminium injecté et moulé sous pression...6	
<b>Figure 1.5.</b> Aspect extérieur d'un rotor à cage.....	6
<b>Figure 1.6.</b> Principe de fonctionnement d'une machine asynchrone.....	7
<b>Figure 1.7.</b> Représentation des enroulements de la machine asynchrone triphasée...9	
<b>Figure 1.8.</b> Schéma physique de la transformée de Park.....	12
<b>Figure 1.9.</b> Repérage angulaire des systèmes d'axes dans l'espace électrique.....	12
<b>Figure 1.10.</b> Modèle Simulink de la machine asynchrone à vide.....	18
<b>Figure 1.11.</b> Simulation de la machine asynchrone à vide.....	19
<b>Figure 1.12.</b> Simulation de la machine asynchrone avec une charge de 10 N.m à vide.....	20

## Chapitre 2

<b>Figure 2.1.</b> Schéma explicatif d'un convertisseur DC/AC (Onduleur).....	22
<b>Figure 2.2.</b> Représentation schématique d'un onduleur de tension.....	23
<b>Figure 2.3.</b> Onduleur de tension en pont triphasé à thyristors.....	24
<b>Figure 2.4.</b> Structures d'un onduleur de tension monophasé.....	25
<b>Figure 2.5.</b> Schéma de l'onduleur triphasé à deux niveaux.....	26
<b>Figure 2.6.</b> Onduleur deux niveaux simplifiés.....	28
<b>Figure 2.7.</b> Modèle Simulink de l'onduleur à deux niveaux de tensions.....	28
<b>Figure 2.8.</b> Schéma synoptique de la M.L.I.....	29
<b>Figure 2.9.</b> Description de la commande M.L.I. sinus – triangle.....	31
<b>Figure 2.10.</b> Modèle Simulink de la commande MLI sinus – triangle.....	32
<b>Figure 2.11.</b> Simulation de la commande MLI sinus – triangle pour $r = 0.8$ et $m = 6$ et $18$ .....	33

## Liste des figures

---

### Chapitre 3

<b>Figure 3.1.</b> Projection du flux rotorique et du courant statorique dans le repère $d, q$ .....	36
<b>Figure 3.2.</b> Schéma bloc de la régulation de vitesse d'une machine asynchrone par commande vectorielle indirecte.....	41
<b>Figure 3.3.</b> Modèle Simulink de la commande vectorielle d'une machine asynchrone.....	42
<b>Figure 3.4.</b> Performance de la commande vectorielle d'une machine Asynchrone.....	43

### Chapitre 4

<b>Figure 4.1.</b> Déplacement d'une particule.....	46
<b>Figure 4.2.</b> Schéma de principe du déplacement d'une particule.....	47
<b>Figure 4.3.</b> Voisinage géographique à l' instant $t$ et $t+1$ .....	50
<b>Figure 4.4.</b> Organigramme générale du PSO.....	52
<b>Figure 4.5.</b> Caractéristique de convergence de l'algorithme PSO.....	55

### Chapitre 5

<b>Figure 5.1.</b> Correction d'un processus avec un régulateur PI.....	56
<b>Figure 5.2.</b> Boucle de régulation de vitesse.....	57
<b>Figure 5.3.</b> Modèle Simulink MCC_PI.m pour la régulation de vitesse d'un MCC.....	59
<b>Figure 5.4.</b> Surface erreur dans la régulation de vitesse d'un MCC.....	59
<b>Figure 5.5.</b> Modèle Simulink de la régulation simultanée de vitesse d'un MCC avec un PI trouvé par PSO off-line et un PI trouvé par placement des pôles.....	62
<b>Figure 5.6.</b> Performances d'un PI-PSO off line et d'un PI par placement des pôles lors d'un démarrage en charge avec perturbation de charge d'un MCC.....	63
<b>Figure 5.7.</b> Performances d'un PI-PSO off line et d'un PI par placement des pôles lors d'une poursuite de la vitesse d'un MCC.....	64
<b>Figure 5.8.</b> Contrôleur PI-PSO on-line pour la commande d'un moteur.....	65

## Liste des figures

---

<b>Figure 5.9.</b> Modèle Simulink du contrôleur PI-PSO on-line.....	65
<b>Figure 5.10.</b> Modèle Simulink de la régulation de vitesse d'un MCC par un PI-PSO.....	66
<b>Figure 5.11.</b> Performances d'un PI-PSO on-line et d'un PI par placement des pôles lors d'un démarrage en charge avec perturbation de charge d'un MCC. ....	68
<b>Figure 5.12.</b> Performances d'un PI-PSO on line lors d'une poursuite de la vitesse d'un MCC.....	69
<b>Figure 5.13.</b> Modèle Simulink MAS_MLIV_PI.m pour la régulation de vitesse D'un moteur asynchrone.....	69
<b>Figure 5.14.</b> Performances d'un PI-PSO off line et d'un PI par placement des pôles lors d'un démarrage en charge avec perturbation de charge d'un MAS.....	72
<b>Figure 5.15.</b> Performances d'un PI-PSO off line et d'un PI par placement des pôles lors d'une poursuite de la vitesse d'un asynchrone.....	73
<b>Figure 5.16.</b> Performances d'un PI-PSO amélioré et d'un PI par placement des pôles lors d'un démarrage en charge avec perturbation de charge d'un MAS.....	76
<b>Figure 5.17.</b> Performances d'un PI-PSO amélioré et d'un PI par placement des pôles lors d'une poursuite de la vitesse d'un moteur asynchrone.....	77
<b>Figure 5.18.</b> Modèle Simulink de la régulation de vitesse d'un MAS par un PI-PSO.....	78
<b>Figure 5.19.</b> Performances d'un PI-PSO on-line et d'un PI par placement des pôles Lors d'un démarrage en charge avec perturbation de charge d'un MAS.....	80
<b>Figure 5.20.</b> Performances d'un PI-PSO on line lors d'une poursuite de la vitesse d'un MAS.....	81

# Sommaire

---

<b>Introduction générale.....</b>	<b>1</b>
-----------------------------------	----------

## **Chapitre 1 Modélisation et simulation d'un moteur asynchrone(MAS)**

1.1 Introduction .....	3
1.2 Constitution, principe de fonctionnement et glissement d'un moteur asynchrone .....	3
1.2.1. Constitution.....	3
1.2.2. Principe de fonctionnement de machine asynchrone.....	4
1.2.3. Glissement d'un moteur asynchrone .....	8
1.3. Représentation vectorielle de la machine .....	9
1.4. Modélisation de la machine asynchrone .....	10
1.4.1. Modélisation de la machine asynchrone dans le repère biphasé de Park .....	11
1.4.1.1. Description de la transformation de Park .....	11
1.4.1.2. Modélisation de la machine asynchrone dans le repère de Park.....	14
1.4.1.2.1. Equations électriques et magnétiques dans le repère de Park .....	14
1.4.1.2.2. Choix de l'orientation du repère de Park .....	15
1.4.1.2.3. Equation d'état de la machine asynchrone dans le repère de Park ...	15
1.4.1.2.4. Expression du couple électromagnétique dans un référentiel lié au champ tournant .....	16
1.4.1.2.5. Equations mécaniques de la machine asynchrone .....	17
1.4.2. Simulation numérique du modèle dynamique de la machine asynchrone .....	17
1.4.2.1. Interprétations des résultats .....	17
1.5. Conclusion .....	21

## **Chapitre 2 Modélisation et simulation d'un onduleur de tension**

2.1. Introduction .....	22
2.2. Différents types de commande des interrupteurs .....	22

# Sommaire

---

2.3. L'onduleur de tension..	23
2.3.1. Circuit de puissance d'un onduleur de tension	23
2.4. La structure élémentaire de l'onduleur de tension	23
2.5. Onduleur de tension triphasé à deux niveaux de tension	24
2.5.1. Principe de l'onduleur de tension triphasé	25
2.6. Modélisation de l'onduleur	26
2.7. L'onduleur triphasé à commande MLI	28
2.7.1. Commande par modulation de largeur d'impulsion	28
2.7.2. Principe de base de MLI	29
2.7.3. Commande MLI sinus - triangle	29
2.7.3.1. Caractéristique de la MLI sinus triangle	29
2.7.3.2. Modélisation de la commande MLI sinus – triangle	31
2.7.3.3. Simulation de la commande MLI sinus – triangle	32
2.8. Conclusion	34

## **Chapitre3 Modélisation et simulation de la commande vectorielle Indirecte d'un moteur asynchrone**

3.1. Introduction	35
3.2. Principe de la commande vectorielle	35
3.3. Les différents types de commande vectorielle	37
3.3.1. Commande vectorielle directe	37
3.3.2. Commande vectorielle indirecte	38
3.4. Mise en équation de la commande vectorielle indirecte	38
3.5. Simulation de la Commande Vectorielle d'un moteur asynchrone	41
3.6. Conclusion	44

# Sommaire

---

## Chapitre 4 Etude, modélisation et simulation de l'algorithme PSO

4.1. Introduction .....	45
4.2. Mode de fonctionnement de l'algorithme PSO .....	45
4.3. Principe de fonctionnement du PSO .....	46
4.4. Mécanisme de confinement .....	48
4.5 Notion de voisinage .....	49
4.6. Conception et déroulement de l'algorithme .....	51
4.7. Avantages et inconvénients de PSO .....	52
4.8. Exemple d'Application de l'algorithme PSO .....	53
4.9. Conclusion .....	55

## Chapitre 5 Conception d'un contrôleur PI-PSO pour contrôler la vitesse d'un MAS

5.1. Introduction .....	56
5.2. Régulateur PI .....	56
5.2.1. Recherche des paramètres du régulateur PI par la méthode de Placement de pôles .....	57
5.3. Conception off - line d'un Contrôleur PI-PSO pour un moteur à courant continu .....	59
5.4. Conception on - line d'un Contrôleur PI-PSO pour un moteur à courant continu .....	64
5.5. Conception off - line d'un Contrôleur PI-PSO pour un moteur asynchrone .....	69
5.6. Conception on - line d'un Contrôleur PI-PSO pour un moteur asynchrone .....	77
5.7. Conclusion .....	81
<b>Conclusion générale</b> .....	82
<b>Bibliographie</b> .....	84
<b>Annexe</b> .....	86

# Introduction générale

---

## Introduction générale

---

Les techniques d'entraînement électrique à vitesse variable, à base de moteur asynchrone MAS, ont pris une importance considérable dans de nombreuses applications industrielles. Le MAS est robuste, simple, rentable et moins cher, il est sûr et son fonctionnement ne génère pas d'étincelles comme c'est le cas pour un moteur à courant continu MCC.

L'application de la commande vectorielle introduite par Kovacs en 1959 et reprise par Blasshke en 1972, constitue une révolution pour le contrôle de la MAS. Cette technique consiste à ramener le comportement de la MAS à celui du MCC, en effectuant une linéarisation asymptotique et un découplage entre le flux et le couple électromagnétique.

La régulation de la vitesse dans la commande vectorielle est nécessaire car elle offre l'adaptation de l'opération de découplage en cas de variation de charge ou des paramètres de la machine. L'utilisation des régulateurs PI pour la régulation de vitesse nécessite la connaissance de valeurs optimales des gains des actions proportionnel et intégrateur et trouve ainsi des difficultés à réagir efficacement aux variations des paramètres de la machine.

La méthode d'optimisation par essaim de particule PSO, proposé en 1995 par Russel Eberhart (ingénieur en électricité) et James Kennedy (socio-psychologue), offre une possibilité de trouver les paramètres optimaux d'un régulateur PI afin d'améliorer les performances du régulateur. Cet algorithme évolutionnaire qui utilise une population de solutions candidates pour optimiser une fonction objective donnée peut être utilisé pour trouver les valeurs optimales des gains du régulateur PI permettant de minimiser une fonction objective dépendante de l'erreur de vitesse.

L'objectif de ce travail est de développer un régulateur PI dont les paramètres sont optimisés par la méthode d'optimisation PSO, nommé PI-PSO, pour être utilisé dans la commande vectorielle d'une machine asynchrone.

La disponibilité et l'accessibilité directes des résultats, à partir de simulateurs sont nécessaires dans tous les travaux de recherche modernes. Ainsi, dans le cadre de cette étude nous avons choisis de travailler sous environnement **Matlab/Simulink**.

Notre travail est divisé en cinq chapitres :

- Le premier chapitre présente la modélisation et la simulation d'un moteur asynchrone dans le repère de Park avec un référentiel lié au champ tournant. Des simulations sous Matlab sont effectuées pour valider le modèle.

## Introduction générale

---

- Le deuxième chapitre présente la modulation d'un onduleur de tension et son utilisation dans l'alimentation de la machine asynchrone. Des simulations sont effectuées pour observer l'effet de l'utilisation d'un onduleur de tension sur les signaux de sortie de la machine.
- Le troisième chapitre présente la modélisation et la simulation de la commande vectorielle Indirecte d'un moteur asynchrone.
- Le quatrième chapitre présente une de l'algorithme PSO avec une modélisation mathématique permettant sa conception. Une simulation de l'algorithme est faite pour la recherche de la valeur minimale d'une fonction à trois variables.
- Le cinquième chapitre présente la conception du régulateur PI-PSO on-line et off-line pour la régulation de la vitesse dans la commande vectorielle d'une machine asynchrone. La fonction objective utilisée étant la surface comprise entre la consigne et la courbe de la vitesse.

Ce mémoire sera achevé par une **conclusion générale**.

# Chapitre 1

## Modélisation et Simulation d'un Moteur asynchrone (MAS)

## **1.1. Introduction**

La représentation des processus physiques par les modèles mathématiques est une étape très importante dans l'asservissement des systèmes. En effet, afin d'élaborer une structure de commande, il est important d'établir un modèle mathématique qui représente fidèlement les caractéristiques du processus. L'objectif de ce chapitre est de donner un aperçu sur la modélisation des machines asynchrones triphasées sous forme d'équations d'état en vue de leur commande, ainsi qu'une simulation de la machine alimentée en tension à l'aide de SIMULINK.

La machine asynchrone ou machine à induction est actuellement la machine électrique dont l'usage est le plus répandu dans l'industrie. Son principal avantage réside dans l'absence de contacts électriques glissants, ce qui conduit à une structure simple et robuste facile à construire. Seulement la machine asynchrone présente un système d'équations très complexe et une structure fortement non linéaire à étudier (MEROUFEL.A, 2004). Il est donc nécessaire de développer un modèle plus simple. Le modèle mathématique de la machine asynchrone est un système à six équations différentielles, la résolution d'un tel système est difficile même avec l'utilisation de l'outil informatique. L'utilisation de la transformation de PARK, sous certaines hypothèses simplificatrices, permet de contourner cette difficulté. Elle permet une représentation biphasée équivalente de la machine triphasée ce qui réduit considérablement la complexité du modèle en vue de la commande.

## **1.2. Constitution, principe de fonctionnement et glissement d'un moteur asynchrone [1]**

### **1.2.1. Constitution**

Une machine asynchrone est formé :

**D'un stator** : anneau de tôles encoché à l'intérieur et portant un enroulement triphasé semblable à celui d'un alternateur. Cet enroulement est presque toujours relié à la source et constitue le primaire.

**D'un rotor**: anneau de tôles rainuré à l'extérieur, concentrique au premier et séparé de lui par un entrefer étroit d'épaisseur constante. Le rotor porte un enroulement polyphasé mis en court-circuit constituant le secondaire. On distingue principalement deux types de structures de rotors:

- ✓ **Un rotor à cage (d'écureuil):** l'ensemble à l'aspect d'une cage cylindrique dont à chaque bout une couronne métallique est raccordée dans laquelle se trouve un empilement de tôles dont l'axe du moteur passe au centre.
- ✓ **Rotor bobiné :** comme pour le rotor à cage il est constitué de tôles empilées, mais des encoches sont pratiquées pour le passage du bobinage qui lui même est raccordé en une extrémité en bout d'arbre sur des bagues l'autre extrémité du bobinage est connecté ensemble (point milieu du couplage étoile). Sur les bagues viennent frotter les balais (aussi appelé les collecteurs) qui sont raccordés au dispositif de démarrage (résistance).

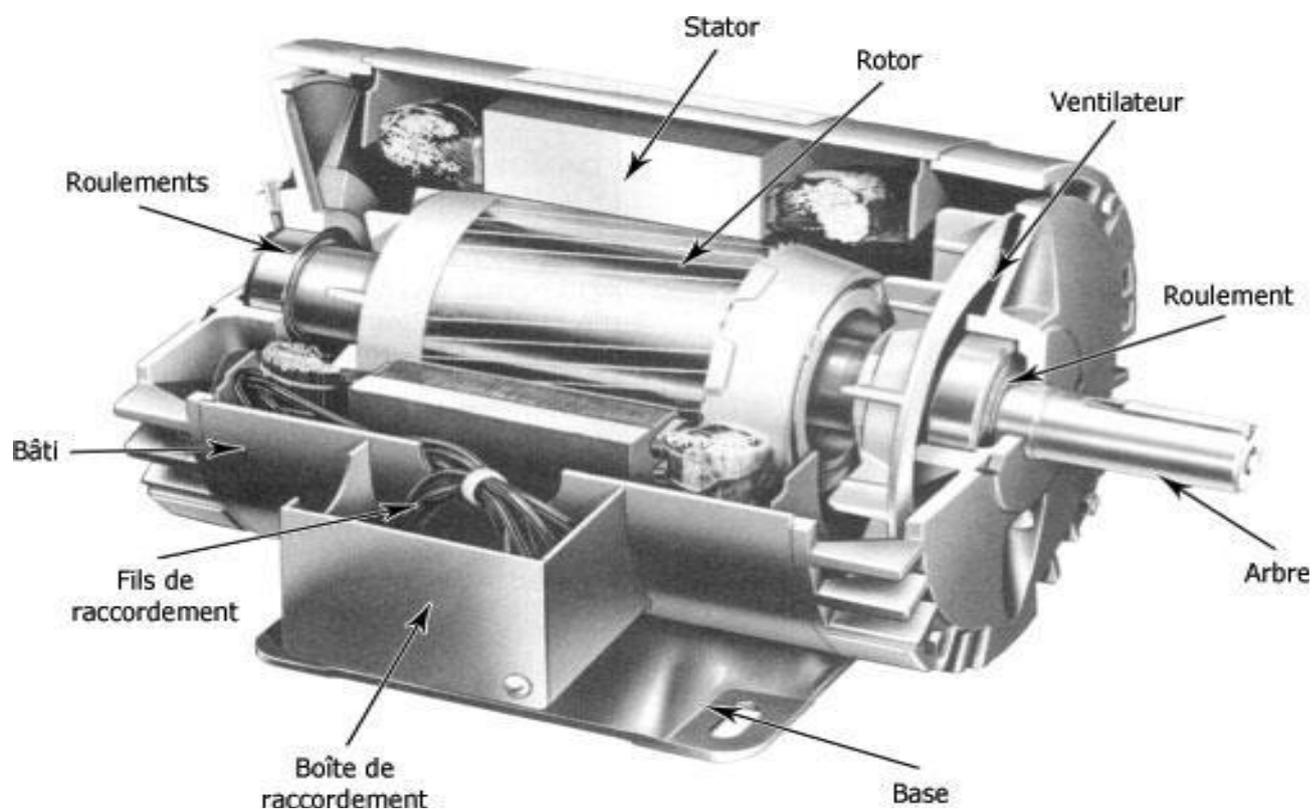


Figure 1.1. Présentation d'une machine asynchrone.

### 1.2.2. Principe de fonctionnement de machine asynchrone

Le stator (ayant  $p$  paires de pôles), alimenté par un réseau de fréquence  $f$ , crée une induction tournante  $B_s$  de vitesse  $N_s$ , telle que  $N_s = f/p$ .

Le rotor immobile est balayé par cette induction et des forces électromotrices sont engendrées dans les conducteurs (loi de Faraday).

Comme les circuits rotoriques sont fermés, des courants rotoriques prennent naissance. Il apparaît des forces électromotrices dues à l'action de l'induction statorique sur les courants

## Chapitre 1 Modélisation et Simulation d'un Moteur asynchrone (MAS)

---

rotoriques. En vertu de la loi de Lenz, ces forces tendent à entraîner le rotor dans le sens des inductions tournantes. Il existe un couple de démarrage, le rotor se met à tourner si le couple est suffisant.

Pour qu'il y ait couple, il faut donc :

- que les circuits rotoriques soient fermés, sinon les courants rotoriques sont nuls ;
- Que la vitesse  $N$  prise par le rotor soit différente de la vitesse  $N_s$  de l'induction.

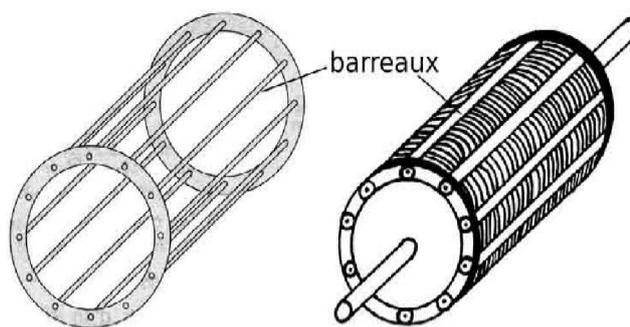
Si  $N = N_s$ , les conducteurs tournent à la vitesse de l'induction statorique, aucune f.é.m. n'est induite, et par conséquent aucun courant ne circule dans le rotor : il ne peut y avoir de couple.

On obtient donc un résultat très différent de celui de la machine synchrone pour laquelle il n'y avait de couple qu'au synchronisme. Pour la machine synchrone :

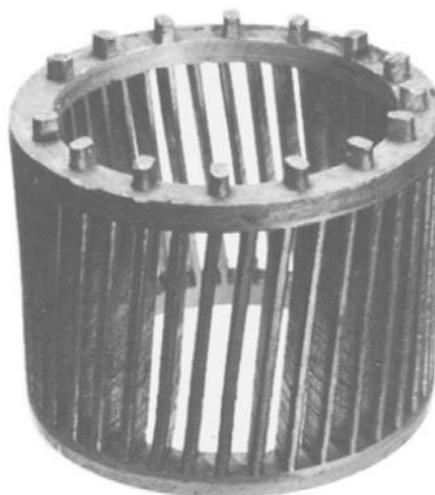
- ✓ si  $N < N_s$  couple moteur
- ✓ si  $N = N_s$  couple nul ;
- ✓ si  $N > N_s$  couple de freinage ;



**Figure 1.2.** Structure d'un rotor bobiné.



**Figure 1.3.** Structure d'un Rotor en cage d'écureuil.



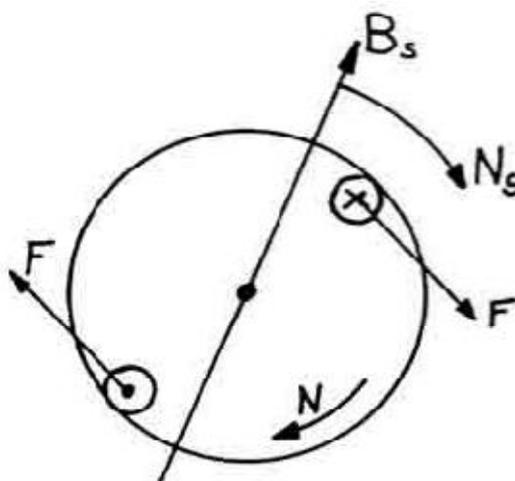
**Figure 1.4.** Rotor à cage d'écureuil en aluminium injecté et moulé sous pression.



**Figure 1.5.** Aspect extérieur d'un rotor à cage.

Démarrage en asynchrone d'un moteur synchrone : on place souvent une cage d'écuréuil sur l'inducteur d'un moteur synchrone. Cette cage s'oppose aux déplacements relatifs du rotor par rapport à l'induction tournante du stator et par suite amortit le mouvement de décalage du rotor lors des phénomènes transitoires dus aux variations brusques de couple (amortisseur Leblanc).

Comme pour le moteur asynchrone, le démarrage est alors possible en l'absence d'excitation continue. Si le moteur est à vide, il atteindra presque la vitesse de synchronisme et on pourra alors le synchroniser en l'excitant. Ce mode de démarrage, très simple, ne convient qu'aux moteurs synchrones de faible puissance en raison du courant et du couple élevé lors de la synchronisation.



**Figure 1.6.** Principe de fonctionnement d'une machine asynchrone.

Les courants statoriques créent un champ magnétique tournant dans le stator. La fréquence de rotation de ce champ est imposée par la fréquence des courants statoriques, c'est-à-dire que sa vitesse de rotation est proportionnelle à la fréquence de l'alimentation électrique. La vitesse de ce champ tournant est appelée vitesse de synchronisme.

L'enroulement au rotor est donc soumis à des variations de flux (du champ magnétique). Une force électromotrice induite apparaît qui crée des courants rotoriques. Ces courants sont responsables de l'apparition d'un couple qui tend à mettre le rotor en mouvement afin de

s'opposer à la variation de flux : loi de Lenz. Le rotor se met donc à tourner pour tenter de suivre le champ statorique.

La machine est dite asynchrone car elle est dans l'impossibilité, sans la présence d'un entraînement extérieur, d'atteindre la même vitesse que le champ statorique. En effet, dans ce cas, vu dans le référentiel du rotor, il n'y aurait pas de variation de champ magnétique ; les courants s'annuleraient, de même que le couple qu'ils produisent, et la machine ne serait plus entraînée. La différence de vitesse entre le rotor et le champ statorique est appelée vitesse de glissement.

### 1.2.3. Glissement d'un moteur asynchrone [2]

Considérons que le champ statorique tourne à la vitesse  $\Omega_s = \omega_s/p$ , où  $\omega_s$  est la pulsation du courant et de la tension statorique, et  $p$  le nombre de paire de pôles, et que  $\Omega_m$  est la vitesse de rotation de la machine.

Le glissement est une grandeur qui rend compte de l'écart de vitesse de rotation d'une machine asynchrone par rapport à une machine synchrone hypothétique construite avec le même stator, il s'agit de la variable  $g$  défini par :

$$g = \frac{N_s - N}{N_s} = \frac{\Omega_s - \Omega_m}{\Omega_s} = \frac{\omega_s - \omega_m}{\omega_s} \quad (1.1)$$

$N_s$  Désigne la vitesse de rotation en tour par seconde du champ tournant statorique :

$$N_s = \frac{\omega_s}{2\pi P} \quad (1.2)$$

$N$  Désigne la vitesse de rotation mécanique en tour par second du rotor :

$$N = \frac{\omega_m}{2\pi P} \quad (1.3)$$

La fréquence  $f_r$  de glissement est donnée par.

$$f_r = g f_s = f \quad (1.4)$$

Le glissement est compté positif pour un fonctionnement en moteur  $\Omega_m < \Omega_s$ , négatif pour un fonctionnement en génératrice  $\Omega_m > \Omega_s$ . La fréquence  $f$  est celle du réseau.

### 1.3. Représentation vectorielle de la machine

La machine asynchrone est représentée dans la figure 1.1 par six enroulements dans l'espace électrique. L'angle  $\theta$  repère l'axe de la phase rotorique de référence  $\vec{R}_a$  par rapport à l'axe fixe de la phase statorique  $\vec{S}_a$ .

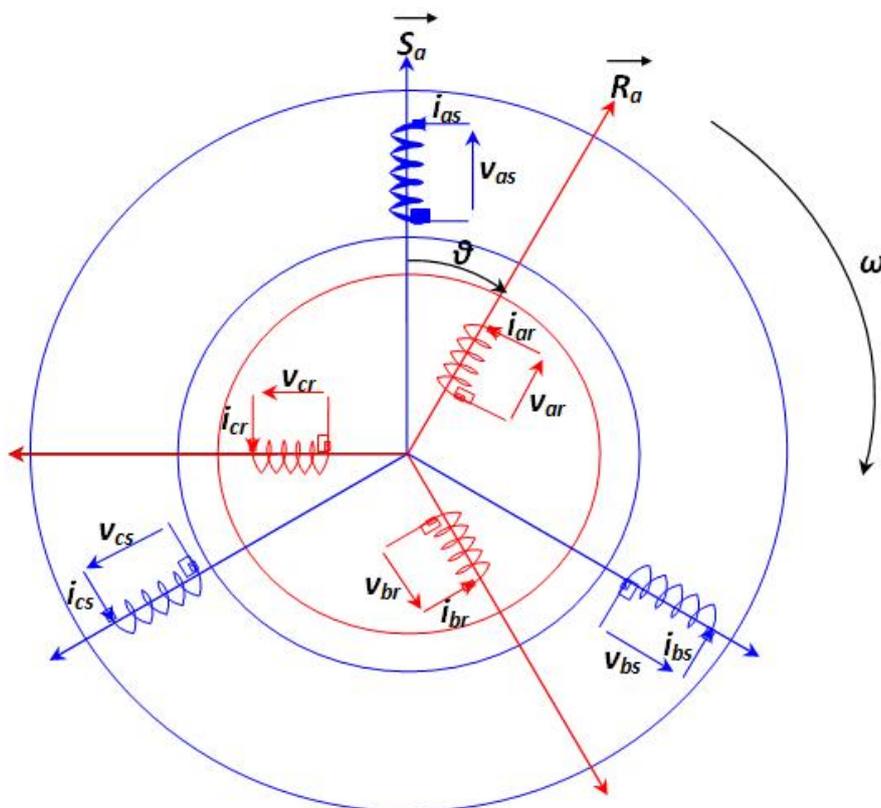


Figure 1.7. Représentation des enroulements de la machine asynchrone triphasée.

## 1.4. Modélisation de la machine asynchrone

Pour modéliser la machine asynchrone, on va supposer les hypothèses simplificatrices suivantes (NOUR.M, 2004) :

- La parfaite symétrie de la machine asynchrone ayant p paires de pôles ;
- Le rotor est bobiné, le bobinage est triphasé au stator et au rotor ;
- Le rotor est à cage, on peut considérer le bobinage triphasé équivalent ;
- Les axes de phase statorique et rotorique sont décalés identiquement d'un angle électrique de  $120^\circ$  ;
- La force magnétomotrice créée par chacune des phases du stator et du rotor est à répartition sinusoïdale ;
- Les résistances des enroulements sont constantes.

La loi de Faraday permet d'obtenir l'équation matricielle des tensions pour l'ensemble des phases statoriques et rotoriques :

$$\begin{cases} V_{sa} = R_s i_{sa} + \frac{d\varphi_{sa}}{dt} \\ V_{sb} = R_s i_{sb} + \frac{d\varphi_{sb}}{dt} \\ V_{sc} = R_s i_{sc} + \frac{d\varphi_{sc}}{dt} \end{cases} \quad (1.5)$$

$$\begin{cases} V_{ra} = R_r i_{ra} + \frac{d\varphi_{ra}}{dt} \\ V_{rb} = R_r i_{rb} + \frac{d\varphi_{rb}}{dt} \\ V_{rc} = R_r i_{rc} + \frac{d\varphi_{rc}}{dt} \end{cases} \quad (1.6)$$

Puisque le rotor est en court circuit on a  $V_{ra} = V_{rb} = V_{rc} = 0$ .

L'équation matricielle du flux est donnée par :

$$\begin{bmatrix} [\varphi_s] \\ [\varphi_r] \end{bmatrix} = \begin{bmatrix} [L_s] & [M_{sr}] \\ [M_{rs}] & [L_r] \end{bmatrix} \begin{bmatrix} [i_s] \\ [i_r] \end{bmatrix} \quad (1.7)$$

avec  $[L_s] = \begin{bmatrix} l_s & m_s & m_s \\ m_s & l_s & m_s \\ m_s & m_s & l_s \end{bmatrix}, \quad [L_r] = \begin{bmatrix} l_r & m_r & m_r \\ m_r & l_r & m_r \\ m_r & m_r & l_r \end{bmatrix}$

$$\text{Et : } [M_{sr}] = [M_{rs}]^T = m_{sr} \begin{bmatrix} \cos \theta & \cos \left( \theta - \frac{4\pi}{3} \right) & \cos \left( \theta - \frac{2\pi}{3} \right) \\ \cos \left( \theta - \frac{2\pi}{3} \right) & \cos \theta & \cos \left( \theta - \frac{4\pi}{3} \right) \\ \cos \left( \theta - \frac{4\pi}{3} \right) & \cos \left( \theta - \frac{2\pi}{3} \right) & \cos \theta \end{bmatrix}$$

Où  $l_s$  ( $l_r$ ) est l'inductance propre d'une phase statorique (rotorique),  $m_s$  ( $m_r$ ) est l'inductance mutuelle entre deux phases statoriques (rotoriques) et  $m_{sr}$  est le maximum de l'inductance mutuelle entre une phase statorique et une phase rotorique.

### 1.4.1. Modélisation de la machine asynchrone dans le repère biphasé de PARK [3]

Le modèle de la machine dans le repère triphasé étant fort complexe, on fait appel pour sa simplification à la transformation de Park. Physiquement elle peut être expliquée par une transformation de trois enroulements de la machine en seulement deux enroulements comme indiquée dans la figure 1.8.

#### 1.4.1.1. Description de la transformation de Park [4]

Dans la transformation de Park, les enroulements statoriques  $a_s, b_s, c_s$  et les enroulements rotoriques  $a_r, b_r, c_r$  sont respectivement transformés en deux enroulements statoriques  $d_s, q_s$  et deux enroulements rotoriques  $d_r, q_r$ . On désigne par  $d$  l'axe direct et  $q$  l'axe en quadrature. La figure 1.9 permet de définir les divers référentiels et les relations spatiales qui les lient.

Elle montre que  $\theta_s, \theta_r$  et  $\theta_m$  sont liés par la relation :

$$\theta_m = \theta_s - \theta_r \quad . \quad (1.8)$$

D'où

$$\frac{d\theta_m}{dt} = \frac{d\theta_s}{dt} - \frac{d\theta_r}{dt} \quad (1.9)$$

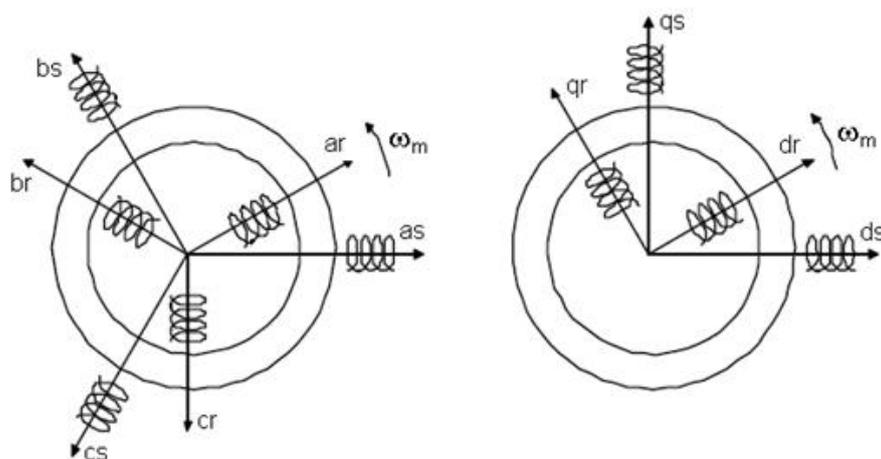


Figure 1.8. Schéma physique de la transformée de Park.

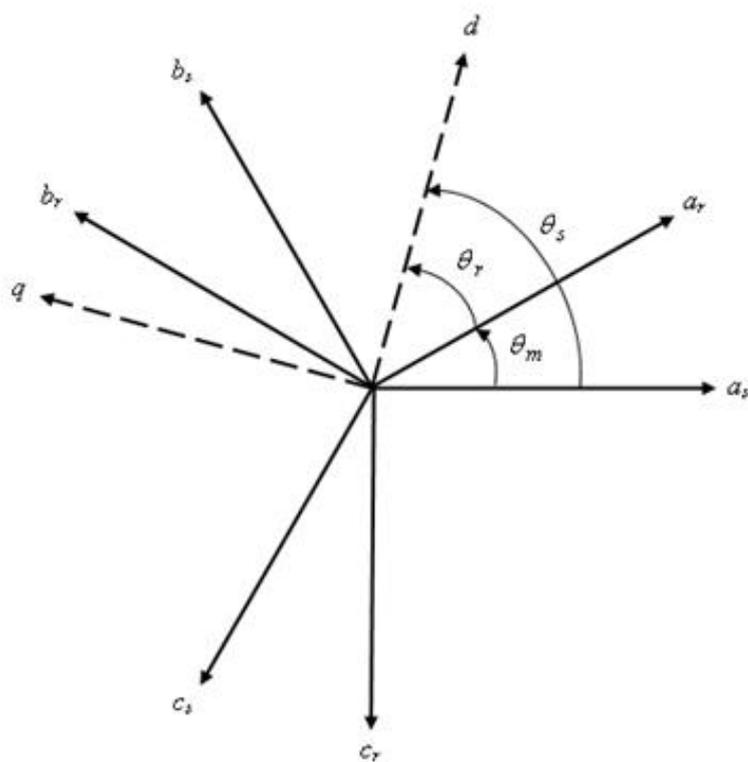


Figure 1.9. Repérage angulaire des systèmes d'axes dans l'espace électrique.

Dans la transformée de Park avec conservation de la puissance, on définit la matrice de transformation suivante [4] :

$$[P(\theta_s)] = \sqrt{\frac{2}{3}} \begin{bmatrix} \cos \theta_s & \cos\left(\theta_s - \frac{2\pi}{3}\right) & \cos\left(\theta_s - \frac{4\pi}{3}\right) \\ -\sin \theta_s & -\sin\left(\theta_s - \frac{2\pi}{3}\right) & -\sin\left(\theta_s - \frac{4\pi}{3}\right) \\ 1/\sqrt{2} & 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix} \quad (1.10)$$

Et son inverse est défini par :

$$[P(\theta_s)]^{-1} = \sqrt{\frac{2}{3}} \begin{bmatrix} \cos \theta_s & -\sin \theta_s & 1/\sqrt{2} \\ \cos\left(\theta_s - \frac{2\pi}{3}\right) & -\sin\left(\theta_s - \frac{2\pi}{3}\right) & 1/\sqrt{2} \\ \cos\left(\theta_s - \frac{4\pi}{3}\right) & -\sin\left(\theta_s - \frac{4\pi}{3}\right) & 1/\sqrt{2} \end{bmatrix} \quad (1.11)$$

La transformation de Park directe et inverse appliquée aux tensions par phase est donnée par les équations 1.12 et 1.13, où  $v_{ds}$  et  $v_{qs}$  sont les composantes direct et en quadrature de la tension statorique et  $v_{os}$  est la composante homopolaire qui est nulle.

$$\begin{bmatrix} v_{ds} \\ v_{qs} \\ v_{os} \end{bmatrix} = [P(\theta_s)] \begin{bmatrix} v_{as} \\ v_{bs} \\ v_{cs} \end{bmatrix} \quad (1.12)$$

$$\begin{bmatrix} v_{as} \\ v_{bs} \\ v_{cs} \end{bmatrix} = [P(\theta_s)]^{-1} \begin{bmatrix} v_{ds} \\ v_{qs} \\ v_{os} \end{bmatrix} \quad (1.13)$$

Les équations 1.12 et 1.13 peuvent aussi être appliquées aux autres grandeurs comme le courant ou le flux. La transformation des grandeurs statoriques est définie par :

$$\begin{bmatrix} i_{ds} \\ i_{qs} \\ i_{os} \end{bmatrix} = [P(\theta_s)] \begin{bmatrix} i_{as} \\ i_{bs} \\ i_{cs} \end{bmatrix} ; \quad \begin{bmatrix} \varphi_{ds} \\ \varphi_{qs} \\ \varphi_{os} \end{bmatrix} = [P(\theta_s)] \begin{bmatrix} \varphi_{as} \\ \varphi_{bs} \\ \varphi_{cs} \end{bmatrix} \quad (1.14)$$

$[P(\theta_s)]$  représente la matrice de transformation appliquée au stator. La matrice de transformation appliquée au rotor s'obtient en remplaçant  $\theta_s$  par  $\theta_r$ .

### 1.4.1.2. Modélisation de la machine asynchrone dans le repère de PARK

#### 1.4.1.2.1. Equations électriques et magnétiques dans le repère de Park [5]

Les équations des tensions et flux dans le repère de Park tournant à la vitesse angulaire  $\omega_s = d\theta_s/dt$  peuvent être obtenues en appliquant la transformation de Park aux équations 1.5, 1.6 et 1.7. Cette transformation nous donne :

$$\begin{cases} v_{ds} = R_s i_{ds} + \frac{d\varphi_{ds}}{dt} - \frac{d\theta_s}{dt} \varphi_{qs} \\ v_{qs} = R_s i_{qs} + \frac{d\varphi_{qs}}{dt} + \frac{d\theta_s}{dt} \varphi_{ds} \\ 0 = R_r i_{dr} + \frac{d\varphi_{dr}}{dt} - \frac{d\theta_r}{dt} \varphi_{qr} \\ 0 = R_r i_{qr} + \frac{d\varphi_{qr}}{dt} + \frac{d\theta_r}{dt} \varphi_{dr} \end{cases} \quad (1.15)$$

$$\begin{cases} \varphi_{ds} = L_s i_{ds} + M i_{dr} \\ \varphi_{qs} = L_s i_{qs} + M i_{qr} \\ \varphi_{dr} = L_r i_{dr} + M i_{ds} \\ \varphi_{qr} = L_r i_{qr} + M i_{qs} \end{cases} \quad (1.16)$$

Où  $L_s = l_s - m_s$ ,  $L_r = l_r - m_r$  et  $M = \frac{3}{2} m_{sr}$  sont les inductances cycliques statorique, rotorique et mutuelle.

L'équation 1.16 peut être transformée pour obtenir les courants statoriques en fonction des flux :

$$\begin{cases} i_{ds} = \frac{1}{\sigma L_s} \varphi_{ds} - \frac{M}{\sigma L_s L_r} \varphi_{dr} \\ i_{qs} = \frac{1}{\sigma L_s} \varphi_{qs} - \frac{M}{\sigma L_s L_r} \varphi_{qr} \\ i_{dr} = \frac{1}{\sigma L_r} \varphi_{dr} - \frac{M}{\sigma L_s L_r} \varphi_{ds} \\ i_{qr} = \frac{1}{\sigma L_r} \varphi_{qr} - \frac{M}{\sigma L_s L_r} \varphi_{qs} \end{cases} \quad (1.17)$$

Où  $\sigma = 1 - \frac{M^2}{L_s L_r}$  est le coefficient de dispersion.

### 1.4.1.2.2. Choix de l'orientation du repère de Park [6]

Il existe plusieurs choix de l'orientation du repère de Park qui dépendent des objectifs de l'application voulue:

- Axes tournants à la vitesse du rotor ( $\theta_r$  constant) : permet l'étude des grandeurs statoriques.
- Axes liés au stator ( $\theta_s = 0$ ) : c'est le repère naturel ou stationnaire de la machine asynchrone. Ce choix permet de simplifier la transformation de Park en celle de Clark dans le cas de non conservation de puissance ou celle de Concordia dans le cas contraire. C'est ce choix qui est utilisé pour la conception de la commande directe du couple.
- Axes solidaires du champs tournant : Ce choix permet d'avoir une pulsation de glissement et s'adapte parfaitement à la commande vectorielle par orientation du flux rotorique.

On désigne par  $\omega_s = d\theta_s/dt$ , la vitesse angulaire des axes  $d, q$  dans le repère statorique ( $a_s, b_s, c_s$ ) et par  $\omega_r = d\theta_r/dt$ , la vitesse angulaire des axes  $d, q$  dans le repère rotorique ( $a_r, b_r, c_r$ ), et par  $\omega_m = d\theta_m/dt$  la vitesse angulaire du repère rotorique dans le repère statorique de sorte qu'à partir de l'expression 1.8 il se déduit :

$$\frac{d\theta_r}{dt} = \omega_s - \omega_r = \omega_{gl} \quad (1.18)$$

Où  $\omega_{gl}$  est la pulsation de glissement.

### 1.4.1.2.3. Equation d'état de la machine asynchrone dans le repère de Park

Dans notre travail on a besoin de la représentation d'état de Park dans un référentiel lié au champ tournant qui est utilisée dans la conception de la commande vectorielle par orientation du flux rotorique. Ce référentiel est noté  $(d, q)$  et il se traduit par les conditions  $d\theta_s/dt = \omega_s$  et  $d\theta_r/dt = \omega_{gl}$ . D'où les équations électriques suivantes :

$$\begin{cases} v_{ds} = R_s i_{ds} + \frac{d\varphi_{ds}}{dt} - \omega_s \varphi_{qs} \\ v_{qs} = R_s i_{qs} + \frac{d\varphi_{qs}}{dt} + \omega_s \varphi_{ds} \\ 0 = R_r i_{dr} + \frac{d\varphi_{dr}}{dt} - \omega_{gl} \varphi_{qr} \\ 0 = R_r i_{qr} + \frac{d\varphi_{qr}}{dt} + \omega_{gl} \varphi_{dr} \end{cases} \quad (1.19)$$

En remplaçant les expressions des flux de l'équation 1.16 dans l'équation 1.19, on obtient les équations électriques de la machine asynchrone dans le référentiel lié au champ tournant :

$$\begin{bmatrix} v_{ds} \\ v_{qs} \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} R_s + L_s \frac{d}{dt} & -L_s \omega_s & M \frac{d}{dt} & -M \omega_s \\ L_s \omega_s & R_s + L_s \frac{d}{dt} & M \omega_s & M \frac{d}{dt} \\ M \frac{d}{dt} & -M \omega_{gl} & R_r + L_r \frac{d}{dt} & -L_r \omega_{gl} \\ M \omega_{gl} & M \frac{d}{dt} & L_r \omega_{gl} & R_r + L_r \frac{d}{dt} \end{bmatrix} \begin{bmatrix} i_{ds} \\ i_{qs} \\ i_{dr} \\ i_{qr} \end{bmatrix} \quad (1.20)$$

L'équation 1.20 est une équation différentielle matricielle dont la solution nous donne l'équation d'état de la machine asynchrone dans un référentiel lié au champ tournant suivante :

$$\frac{dx}{dt} = Ax + Bu \quad (1.21)$$

$$\text{Où } x = [i_{sd} \ i_{sq} \ i_{rd} \ i_{rq}]^T, \ u = [v_{sd} \ v_{sq}]^T, \ B = \frac{1}{\sigma L_s} \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -\beta_r & 0 \\ 0 & -\beta_r \end{bmatrix} \text{ et}$$

$$A = \begin{bmatrix} -\frac{1}{\sigma T_s} & \omega_s + \frac{1-\sigma}{\sigma} \omega_r & \frac{\beta_s}{\sigma T_r} & \frac{\beta_s}{\sigma} \omega_r \\ -\left(\omega_s + \frac{1-\sigma}{\sigma} \omega_r\right) & -\frac{1}{\sigma T_s} & -\frac{\beta_s}{\sigma} \omega_r & \frac{\beta_s}{\sigma T_r} \\ \frac{\beta_r}{\sigma T_s} & -\frac{\beta_r}{\sigma} \omega_r & -\frac{1}{\sigma T_r} & \omega_s - \frac{1}{\sigma} \omega_r \\ \frac{\beta_r}{\sigma} \omega_r & \frac{\beta_r}{\sigma T_s} & -\left(\omega_s - \frac{1}{\sigma} \omega_r\right) & -\frac{1}{\sigma T_r} \end{bmatrix}$$

$T_s = \frac{L_s}{r_s}$  Et  $T_r = \frac{L_r}{r_r}$  sont les constantes de temps statorique et rotorique du MAS.

Et  $\beta_s = L_m/L_s$  et  $\beta_r = L_m/L_r$

#### 1.4.1.2.4. Expression du couple électromagnétique dans un référentiel lié au champ tournant

Le couple électromagnétique est exprimé en fonction des courants rotoriques et statoriques.

$$C_e = pL_m(i_{sq}i_{rd} - i_{sd}i_{rq}) \quad (1.22)$$

### 1.4.1.2.5. Equations mécaniques de la machine asynchrone

La relation fondamentale de la dynamique permet d'écrire :

$$C_e - C_r = J \frac{d\Omega}{dt} + f\Omega \quad (1.23)$$

Où  $C_e$  représente le couple électromagnétique produit par le moteur,  $C_r$  le couple résistant,  $J$  le moment d'inertie de l'ensemble des parties tournantes et  $f$  le coefficient de frottement. Cette relation est une équation différentielle dont la variable est la vitesse angulaire du rotor.

### 1.4.2. Simulation numérique du modèle dynamique de la machine asynchrone

Après avoir réalisé le modèle dynamique de la machine asynchrone, une simulation de ce modèle est nécessaire afin d'étudier les performances dynamiques de cette machine.

La figure 1.10 montre le modèle de simulation de la machine asynchrone en utilisant le logiciel Simulink de Matlab. Dans cette figure les composantes de Park de la tension d'alimentation sont utilisées par l'équation d'état de la machine asynchrone afin d'estimer le courant. Ceux – ci vont permettre l'évaluation du couple électromagnétique et ainsi de la vitesse du rotor. Ce premier modèle est utilisé dans la commande vectorielle par orientation du flux rotorique. Les figures 1.11 et 1.12 montrent la simulation des performances dynamiques de ce modèle à vide et avec une charge de 10 N.m. Les paramètres de la machine utilisée sont donnés dans l'annexe.

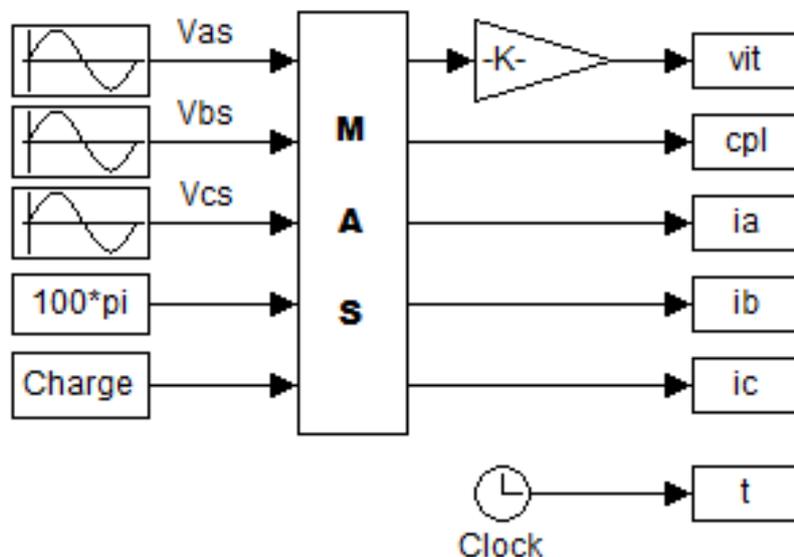
#### 1.4.2.1. Interprétations des résultats

Lors du démarrage à vide de la machine asynchrone, on constate un bref mais important appel de courant atteignant presque 7 fois le courant nominal. Ceci se répercute évidemment sur le couple électromagnétique produit par la machine qui se présente sous forme d'une courbe oscillatoire amorti prenant une valeur moyenne de 25 N.m et une valeur maximale de 45 N.m en régime transitoire, et une valeur constante presque nul en régime permanent. Les oscillations de couple se font évidemment ressentir sur l'évolution de la vitesse qui en régime permanent se stabilise à 1500 tr/mn.

L'application de la charge au démarrage se traduit par une dilatation dans le temps de l'appel de courant au démarrage et une augmentation de sa valeur en régime permanent. Ceci va évidemment se répercuter sur le couple électromagnétique dont la valeur en régime

## Chapitre 1 Modélisation et Simulation d'un Moteur asynchrone (MAS)

permanent dépasse de peu celle du couple résistant. Quant à la vitesse du rotor, on constate une augmentation de son temps de réponse et une chute d'environ 5.3% de sa valeur à vide.



Sous système MAS

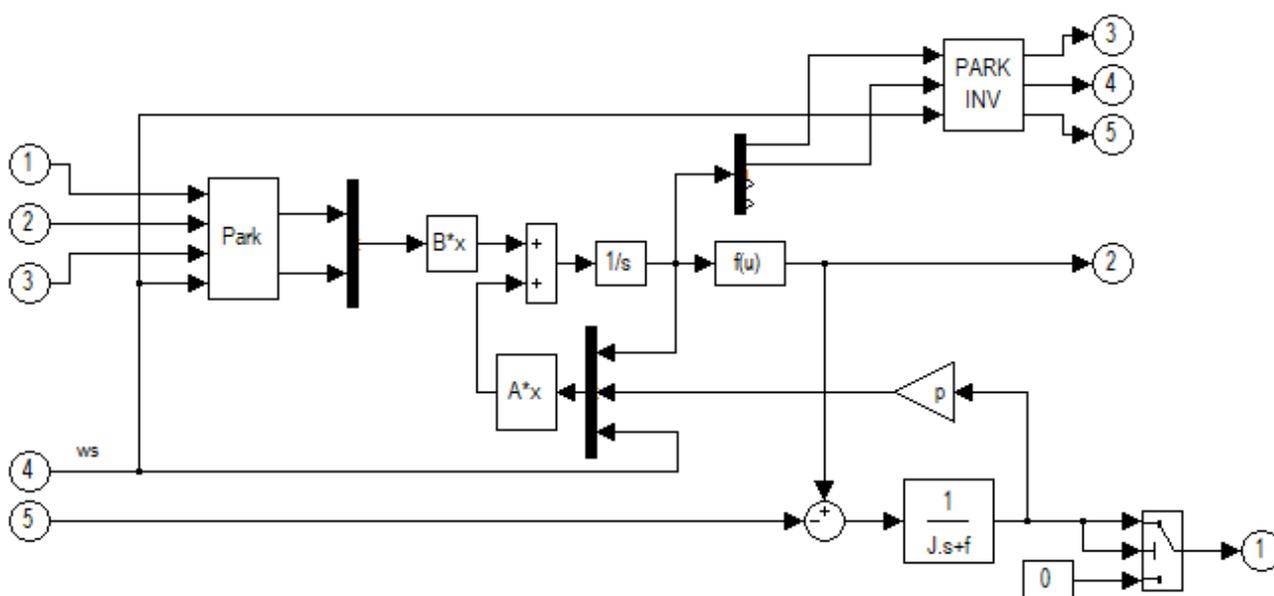


Figure 1.10. Modèle Simulink de la machine asynchrone à vide.

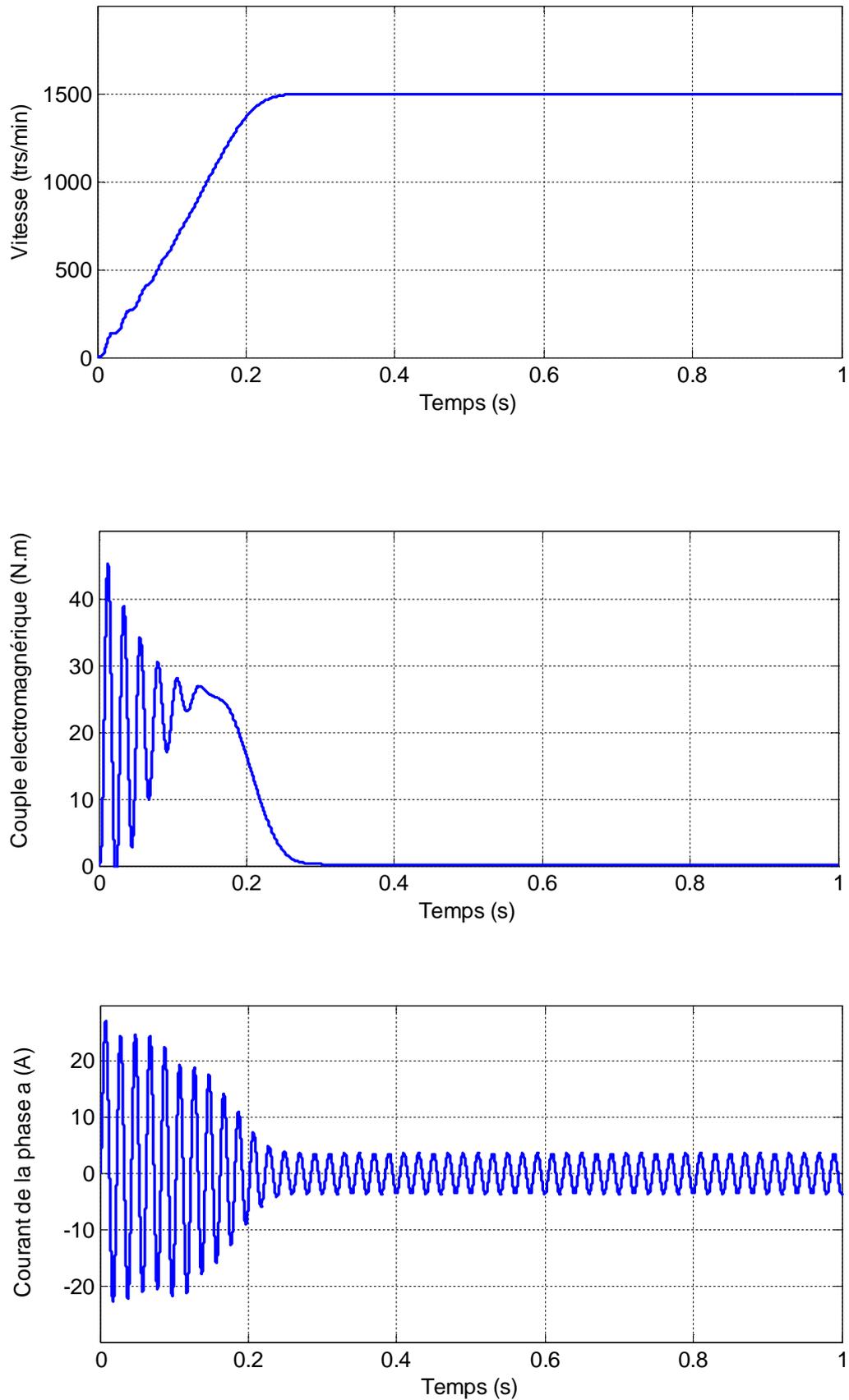


Figure 1.11. Simulation de la machine asynchrone à vide.

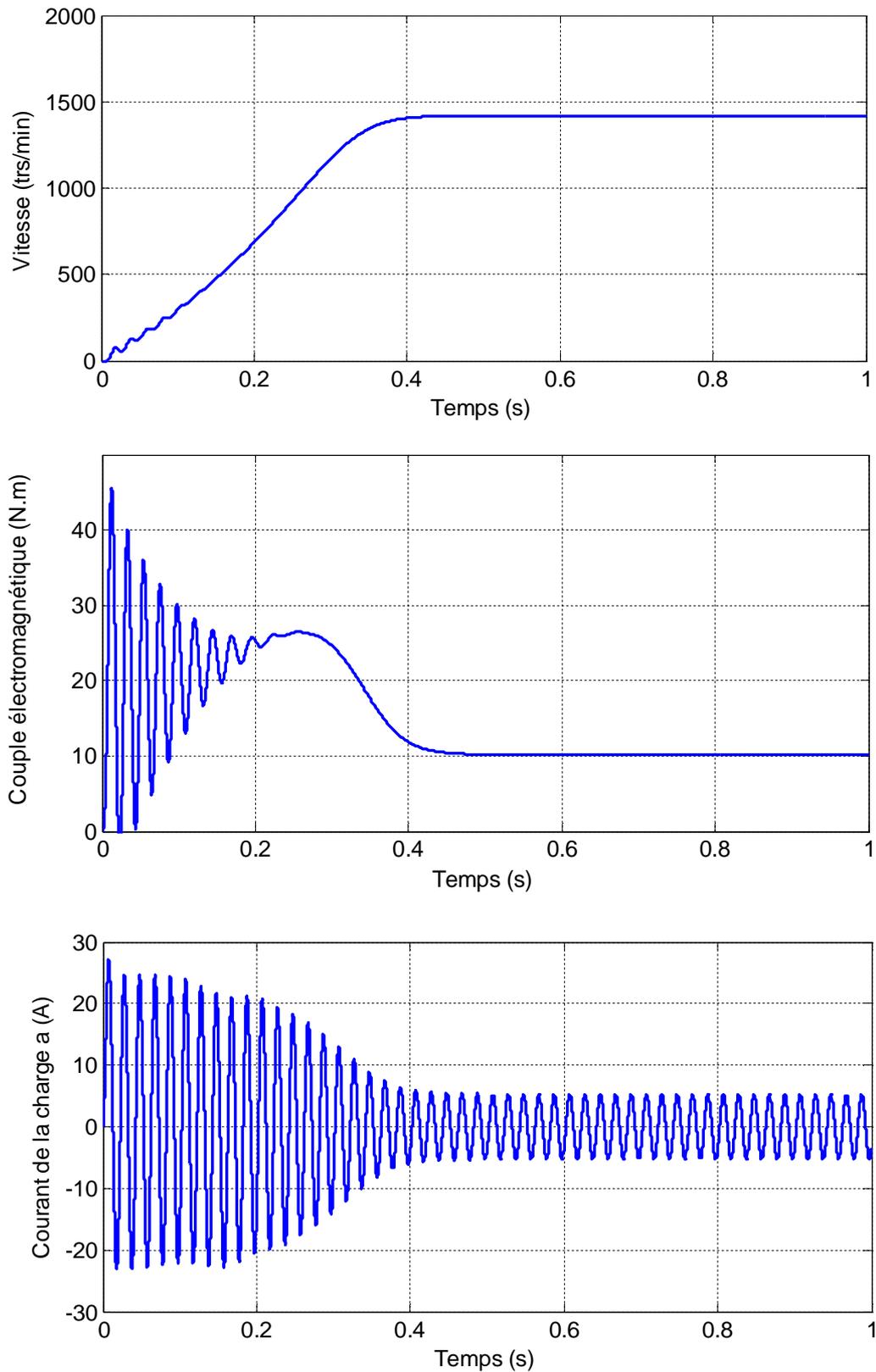


Figure 1.12. Simulation de la machine asynchrone avec une charge de 10 N.m à vide.

### **1.5. Conclusion**

Dans ce chapitre on a procédé à la modélisation de la machine asynchrone, puis on a fait une simulation des performances de cette machine. D'après les résultats de simulations obtenues, on peut conclure que le modèle mathématique proposé simule bien les performances réelles d'une machine asynchrone.

# Chapitre 2

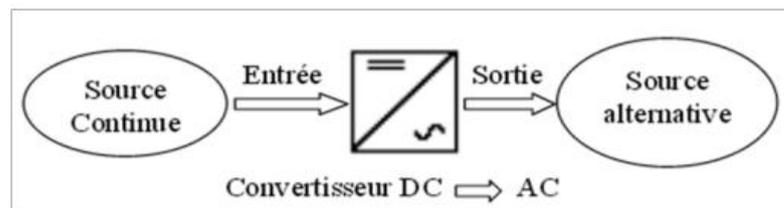
## Modélisation et Simulation d'un Onduleur de tension

## 2.1. Introduction [7]

Les onduleurs tiennent une place importante dans la conversion de l'énergie électrique. Ils sont utilisables pour générer des systèmes de tension alternatives et indispensables dans la constitution des systèmes d'entraînement de la machine alternative.

Les onduleurs sont des convertisseurs statique assurant la conversion continu-alternatif. Par une séquence adéquate de commande des semi-conducteurs de puissance (interrupteurs électroniques), il est possible d'obtenir à la sortie d'un convertisseur statique une tension alternative (ou courant alternatif) de valeur moyenne nulle.

Un onduleur est un dispositif d'électronique de puissance permettant de délivrer des tensions et des courants alternatifs à partir d'une source d'énergie électrique continue.



**Figure 2.1.** Schéma explicatif d'un convertisseur DC/AC (Onduleur).

Les structures des convertisseurs conduit à distinguer deux types d'onduleurs :

- Les onduleurs de tensions.
- Les onduleurs de courants.

Les premiers sont alimentés par une source de tension continue, les seconds par une source de courant continue.

Les onduleurs sont largement utilisés dans plusieurs applications industrielles :

- Variation de vitesse des moteurs asynchrones ou synchrones.
- Four à induction.
- Alimentation dans les avions.
- Alimentation sans interruption de secours.

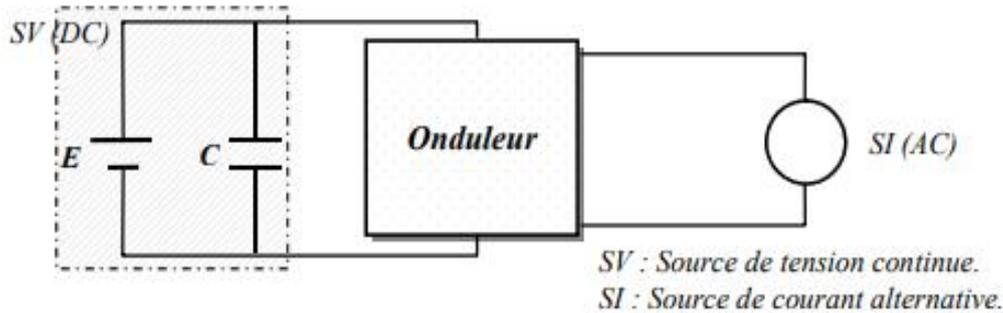
## 2.2. Différents types de commande des interrupteurs [7]

- La commande à onde rectangulaire.
- La commande à créneaux de largeur variable.
- La commande à paliers (à marche d'escalier).

- La commande à modulation de largeur d'impulsion.

### 2.3. L'onduleur de tension [8]

C'est un onduleur qui est alimenté par une source de tension continue, d'impédance interne négligeable. Sa tension n'est pas affectée par la variation du courant qui la traverse, la source continue impose la tension à l'entrée de l'onduleur et donc à sa sortie, figure 2.2 :



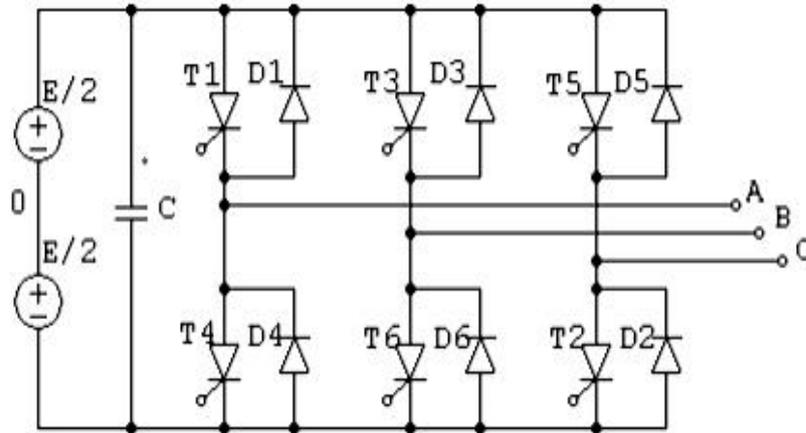
**Figure 2.2.** Représentation schématique d'un onduleur de tension.

L'onduleur de tension est autonome si la fréquence de la tension de sortie est indépendante du réseau alternatif, il est non autonome (assisté) si la fréquence est imposée par le réseau alternatif. Selon le choix des interrupteurs commandés, les onduleurs présentent des avantages les uns par rapport aux autres. C'est pourquoi un onduleur de tension à transistors présente des avantages par rapport à celui à thyristors :

- Facilité de commande (le transistor est commandé à l'ouverture et à la fermeture)
- Élimination du circuit de commutation forcée,
- Faibles pertes de commutation,
- Possibilité de fonctionnement à des fréquences plus élevées.

### 2.3.1. Circuit de puissance d'un onduleur de tension

Prenons comme exemple d'un onduleur de tension, le montage d'un pont triphasé à thyristors, figure 2.3 :



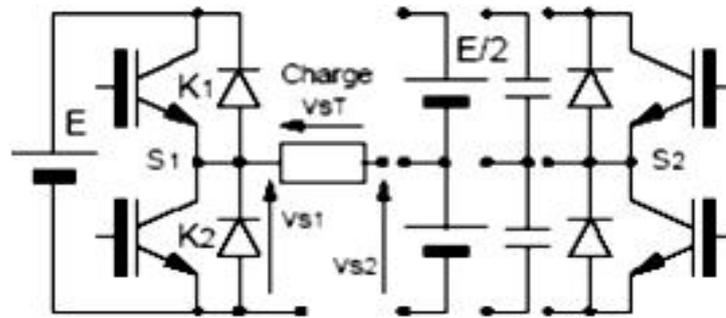
**Figure 2.3.** Onduleur de tension en pont triphasé à thyristors.

Les diodes en tête-bêche sur les interrupteurs commandés du circuit de puissance jouent un rôle fondamental dans le fonctionnement de l'onduleur en assurant essentiellement trois fonctions :

- Faciliter l'extinction des interrupteurs commandés en les polarisant en inverse lorsqu'il s'agit de thyristors,
- Autoriser la démagnétisation de la self de la charge inductive au moment de l'ouverture des interrupteurs,
- Assurer le redressement des courants débits par la charge (cas de la machine asynchrone) lorsqu'elle fonctionne en génératrice.

### 2.4. La structure élémentaire de l'onduleur de tension :

Pour obtenir une tension alternative, on a vu qu'il fallait utiliser une cellule de commutation et un pont capacitif à point milieu, la dynamique de la tension aux bornes de la charge est limitée, elle évolue entre  $\pm E/2$ . On utilise plutôt deux cellules de commutation en mode différentiel, comme représenté ci-dessous.



**Figure 2.4.** Structures d'un onduleur de tension monophasé.

On remarque que les interrupteurs sont bidirectionnels en courant, ce qui est nécessaire puisque le courant de sortie l'est systématiquement.

Cette propriété confère donc un comportement de réversibilité en puissance de la structure.

Dans tous les cas, les états des interrupteurs sont complémentaires avec parfois un temps mort destiné à éviter la conduction simultanée du bras.

## 2.5. Onduleur de tension triphasé à deux niveaux de tension [9]

### 2.5.1. Principe de l'onduleur de tension triphasé

L'onduleur triphasé en pont est constitué d'une source de tension continue et de six interrupteurs montés en pont. La tension continue est généralement obtenue par un redresseur triphasé à diode suivi d'un filtre.

Celui-ci est très utilisé en **MLI** pour l'alimentation des récepteurs triphasés équilibrés à tension et fréquence variable. Il peut être considéré comme étant la superposition de trois onduleurs en demi ponts monophasés (figure (2.5)).

Chacune des trois tensions de sortie est formée d'une onde bivalente les valeurs  $-E$  et  $+E$  mais décalées de  $2\pi/3$  l'une par rapport à l'autre.

De plus si le récepteur est couplé en étoile sans neutre ou en triangle, les harmoniques multiples de trois sont éliminées. Ainsi, le système triphasé obtenu à la sortie de l'onduleur est un système équilibré en tension ne contenant que les harmoniques impaires différents de trois.

Pour obtenir une tension alternative à partir d'une tension continue, il faut découper la tension d'entrée et l'appliquer une fois dans un sens, l'autre fois dans l'autre sens à la charge. L'onduleur de tension alimenté par une source de tension parfaite impose à sa sortie, grâce au jeu d'ouverture et de fermeture des interrupteurs, une tension alternative formée d'une

succession de créneaux rectangulaires à deux niveaux. La période de fonctionnement est fixée par la commande des interrupteurs.

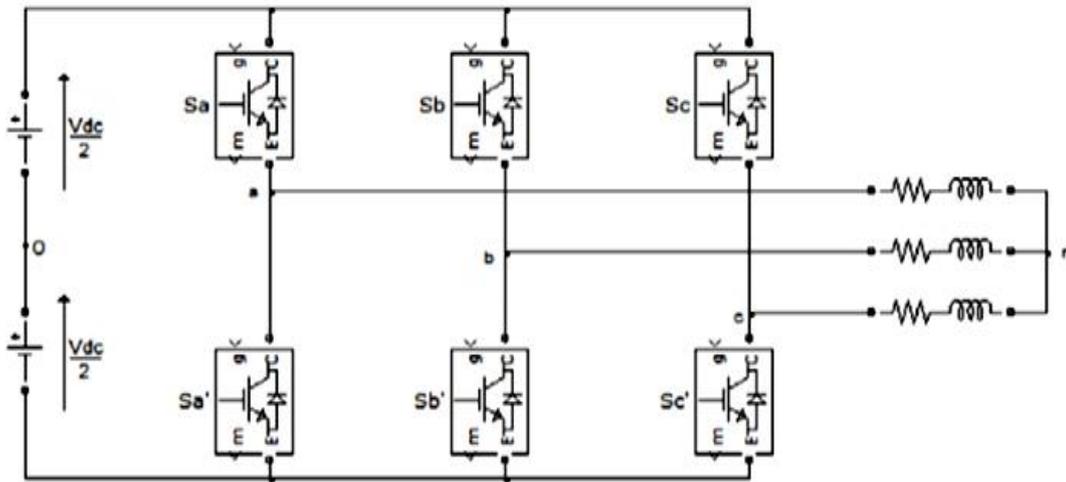


Figure 2.5. Schéma de l'onduleur triphasé à deux niveaux

Le schéma de principe de l'onduleur triphasé à deux niveaux de tension monté en pont alimentant une charge, est donné par la figure 2.5. La tension continue est généralement obtenue par un redresseur triphasé à diode suivi d'un filtre LC.

### 2.6. Modélisation de l'onduleur [7]

La machine est connectée à un onduleur triphasé à commutation commandée. Le point milieu de la source de tension est fictif. Les commandes des interrupteurs de même branche sont disjointes.

Les tensions générées par la MLI sont référencées par rapport au point fictif '0' de la source de l'onduleur (figure 2.5)

- Les tensions de référence par rapport au point neutre sont :

$$v_r = \begin{cases} v_{an} \\ v_{bn} \\ v_{cn} \end{cases} \tag{2.1}$$

- Les tensions aux bornes de l'onduleur

$$\begin{cases} v_{ab} = v_{ao} - v_{bo} \\ v_{bc} = v_{bo} - v_{co} \\ v_{ca} = v_{co} - v_{ao} \end{cases} \tag{2.2}$$

- Les tensions simples aux bornes de la charge montée en étoile sont données par :

$$\begin{cases} v_{an} = v_{ao} + v_{on} \\ v_{bn} = v_{bo} + v_{on} \\ v_{cn} = v_{co} + v_{on} \end{cases} \quad (2.3)$$

En faisant la somme des équations du système (2.3), on obtient :

$$v_{an} + v_{bn} + v_{cn} = v_{ao} + v_{bo} + v_{co} + 3v_{on} \quad (2.4)$$

Sachant que dans un régime équilibré ( $v_{an} + v_{bn} + v_{cn} = 0$ ), donc nous pouvons écrire :

$$v_{ao} + v_{bo} + v_{co} = -3v_{on} \quad (2.5)$$

Donc

$$v_{on} = -\frac{1}{3} (v_{ao} + v_{bo} + v_{co}) \quad (2.6)$$

En substituant l'équation 2.6 dans le système 2.3 il vient alors :

$$\begin{bmatrix} v_{an} \\ v_{bn} \\ v_{cn} \end{bmatrix} = \frac{1}{3} \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix} \begin{bmatrix} v_{ao} \\ v_{bo} \\ v_{co} \end{bmatrix} \quad (2.7)$$

Selon la condition des interrupteurs statiques ( $S_k$ ) de l'onduleur ( $S_k$ ) est égale à 1 si l'interrupteur est fermé et 0 sinon, avec  $k = a, b, c$ , les tensions de branches  $v_{ko}$  peuvent être exprimées en fonction des interrupteurs «  $S_k$  » par :

$$v_{ko} = (2S_k - 1) \cdot \frac{E}{2} \quad (2.8)$$

En utilisant l'expression 2.8 dans le système 2.7, on déduit les tensions de sortie de phase de l'onduleur comme suit :

$$\begin{bmatrix} v_{an} \\ v_{bn} \\ v_{cn} \end{bmatrix} = \frac{E}{6} \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix} \begin{bmatrix} 2S_a - 1 \\ 2S_b - 1 \\ 2S_c - 1 \end{bmatrix} \quad (2.9)$$

Après simplification, le modèle mathématique de l'onduleur à deux niveaux de tensions est donné par l'équation 2.10 où la condition des interrupteurs statiques  $S_k$  ( $k=a,b,c$ ) prennent la valeur 1 si l'interrupteur est fermé et la valeur 0 si l'interrupteur est ouvert.

$$\begin{bmatrix} v_{an} \\ v_{bn} \\ v_{cn} \end{bmatrix} = \frac{E}{3} \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix} \begin{bmatrix} S_a \\ S_b \\ S_c \end{bmatrix} \quad (2.10)$$

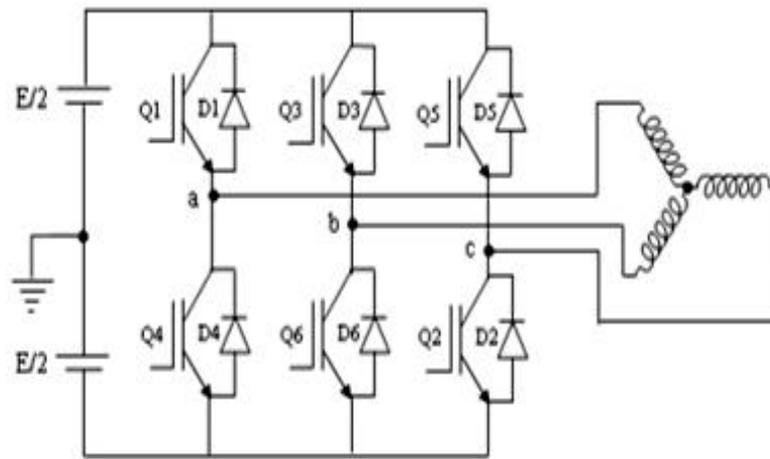


Figure 2.6. Onduleur deux niveaux simplifiés.

La figure 2.7.Représente le modèle Simulink de l'onduleur deux niveaux.

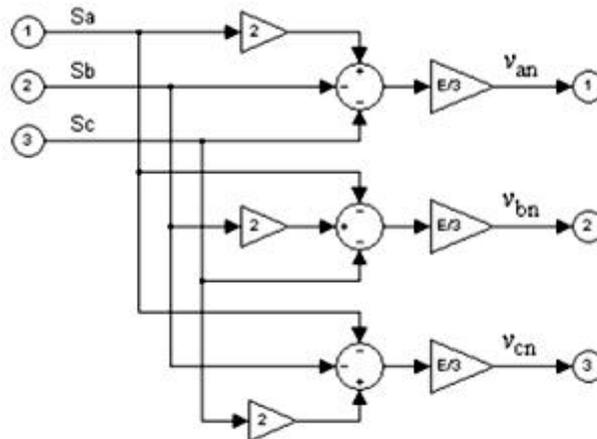


Figure 2.7. Modèle Simulink de l'onduleur à deux niveaux de tensions.

## 2.7. L'onduleur triphasé à commande MLI

### 2.7.1. Commande par modulation de largeur d'impulsion

La modulation de largeur d'impulsion « MLI » (en anglo-saxon Pulse Width Modulation « PWM ») consiste à adapter une fréquence de commutation supérieure à la fréquence des grandeurs de sortie et à former chaque alternance d'une tension de sortie d'une succession de créneaux de largeurs convenables. Cette technique peut être considérée comme une extension du principe de la commande par découpage ou la durée des impulsions n'est plus régulière mais choisie de façon à supprimer le maximum d'harmoniques de rangs faibles difficiles à filtrer.

### 2.7.2. Principe de base de MLI

Le principe de base de la modulation de largeur d'impulsion est sur le découpage d'une pleine onde rectangulaire. Ainsi la tension de sortie de l'onduleur est formée par une succession de créneaux d'amplitude égale à la tension d'alimentation (continue) et de largeur variable. La technique la plus répandue pour la reproduction d'un signal MLI est de comparer un signal triangulaire appelé porteuse de haute fréquence à un signal de référence appelé modulatrice et qui constitue l'énergie du signal recueil à la sortie de l'onduleur, figure 2.8.

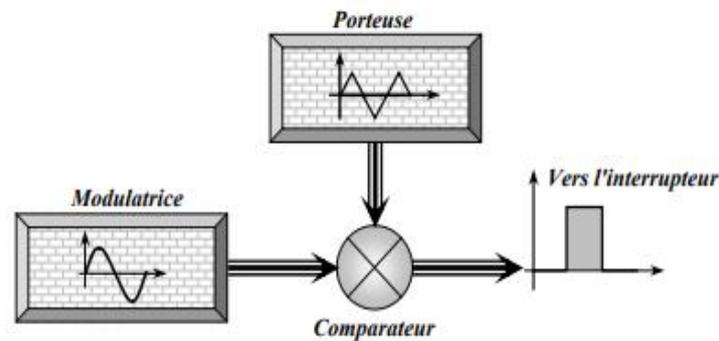


Figure 2.8. Schéma synoptique de la M.L.I.

### 2.7.3. Commande MLI sinus - triangle [10]

Les tensions obtenues aux bornes du récepteur pour les onduleurs triphasés conventionnels présentent plusieurs harmoniques. Il est donc nécessaire de chercher à se rapprocher d'une forme d'onde sinusoïdale. Pour cela on fait appel à la technique de modulation de largeur d'impulsion (MLI). Dans ce cas, la tension et la fréquence de sortie sont contrôlées par l'onduleur en utilisant la technique MLI. Cette méthode consiste à adopter une fréquence de commutation supérieure à la fréquence des grandeurs de sortie, et à former chaque alternance de la tension de sortie d'une succession de créneaux de largeurs convenables. Figure 2.9.

#### 2.7.3.1. Caractéristique de la MLI sinus triangle

Si la tension de référence est sinusoïdale, deux paramètres caractérisent la commande ; l'indice de modulation  $m$  et le taux de modulation  $r$ .

##### a) Indice de modulation $m$ :

L'indice de modulation  $m$  est égal au rapport de la fréquence de la porteuse sur celle de la référence. La valeur de cet indice est sujette à un compromis entre un bon rejet des

harmoniques et un bon rendement de l'onduleur qui se détériore à cause des pertes de commutations. [11]

$$m = \frac{f_p}{f_r} \quad (2.11)$$

Avec :  $m > 1$

On choisit généralement  $m$  supérieure à l'unité parce que l'augmentation mène au déplacement des harmoniques vers des fréquences élevés. Dans la simulation on constate que les valeurs très élevées de  $m$  provoquent une augmentation des déchets de tension, ce qui nous oblige à optimiser la valeur de  $m$ .

#### b) Coefficient de réglage de tension ( $r$ ) :

Le taux de modulation  $r$  est égal au rapport de l'amplitude de la tension de référence sur la valeur de crête de la porteuse. Dans la pratique, on s'arrange toujours à avoir un taux de modulation inférieur à l'unité, de façon à éviter les ratés de commutation qui pourront entraîner des discontinuités de fonctionnement, et en particulier dans les applications à vitesse variable où l'on fait varier l'amplitude de la tension de référence.

Le coefficient de réglage de la tension ou le taux de modulation est donnée par :

$$r = \frac{V_r}{V_p} \quad (2.12)$$

Où  $V_r$  est l'amplitude de la tension de référence et  $V_p$  est celle de la porteuse.

Si  $r = 0$ , la tension de sortie est formée d'onde rectangulaire, de même largeur est de fréquence  $m f_r$ , pour la fondamentale.

Il est préférable d'avoir un taux de modulation  $r$  inférieur à l'unité afin de prévenir les ratages de commutation qui pourraient entraîner des discontinuités.

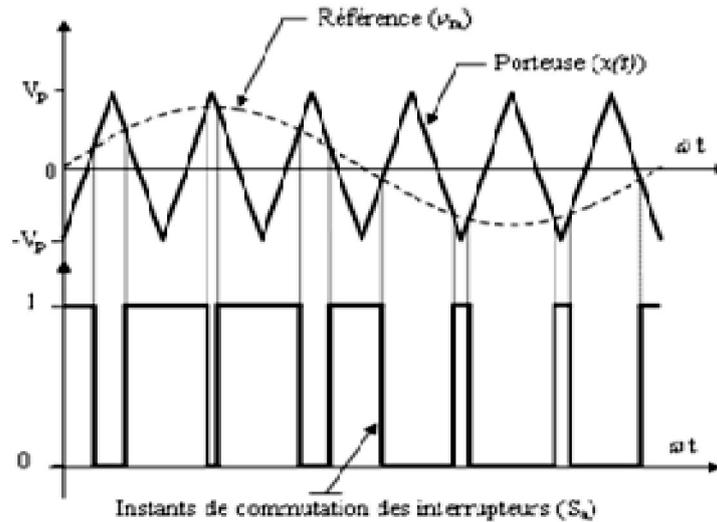


Figure 2.9. Description de la commande M.L.I. sinus – triangle.

### 2.7.3.2. Modélisation de la commande MLI sinus – triangle [7]

#### a) L'étude des structures symétrique de l'onduleur

L'étude se fait sur l'onduleur triphasé a deux niveaux a pleine onde et a modulation d'impulsion ;

##### a.1) La référence :

On représente les tensions de référence dans le cas de la MLI en triphasé par trois signaux sinusoïdaux déphasés les uns par rapport aux autres de  $120^\circ$ .

Les trois tensions sinusoïdales de référence sont données par :

$$\begin{cases} v_{ra}(t) = v_r \sin(2\pi f_r t) \\ v_{rb}(t) = v_r \sin(2\pi f_r t - 2\pi/3) \\ v_{rc}(t) = v_r \sin(2\pi f_r t - 4\pi/3) \end{cases} \quad (2.13)$$

##### a.2) La porteuse :

C'est un signal triangulaire qui peut être considéré comme une alternance de deux droites de pentes opposées. Elle est caractérisée par deux grandeurs :

- L'amplitude  $V_p$
- La fréquence  $f_p$

On définit l'équation de la porteuse dans sa période  $[0, T_p]$  par :

$$x(t) = \begin{cases} V_p (-1 + 4t/T_p) & \text{si } t \in [0, T_p/2] \\ V_p (3 - 4t/T_p) & \text{si } t \in [T_p/2, T_p] \end{cases} \quad (2.14)$$

Avec :  $T_p = 1/f_p$

### a.3) Equations des états des interrupteurs de l'onduleur :

La commande MLI sinus triangle utilise la comparaison avec la porteuse des trois composantes de la tension de référence afin de calculer les états  $S_a$ ,  $S_b$  et  $S_c$  des interrupteurs de l'onduleur. Ceux-ci sont donnés par l'équation 1.36 suivante :

$$S_{abc} = \begin{cases} 1 & \text{si } (v_{rabc} - x(t)) \geq 0 \\ 0 & \text{si } (v_{rabc} - x(t)) < 0 \end{cases} \quad (2.15)$$

#### 2.7.3.3. Simulation de la commande MLI sinus – triangle

La figure 2.10 représente le modèle Simulink de la commande MLI sinus triangle et la figure 2.11 montre la simulation des états  $S_a$ ,  $S_b$  et  $S_c$  des interrupteurs de l'onduleur ainsi que des tensions de sortie  $v_{an}$ ,  $v_{bn}$  et  $v_{cn}$  et leur spectre quand les tensions d'entrées sont triphasées sinusoïdales de fréquence 50 Hz et d'amplitude 220 V. Dans la simulation on a pris  $r = 0.8$ ,  $m = 6$  et  $m = 18$  et  $E = 500$  V

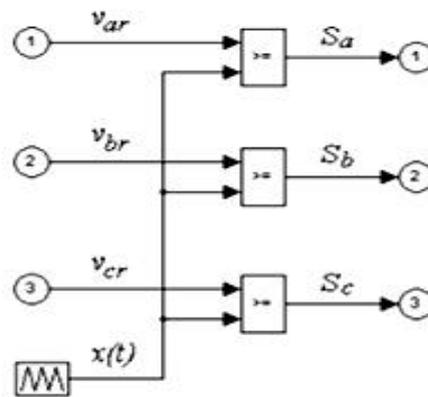


Figure 2.10. Modèle Simulink de la commande MLI sinus – triangle.

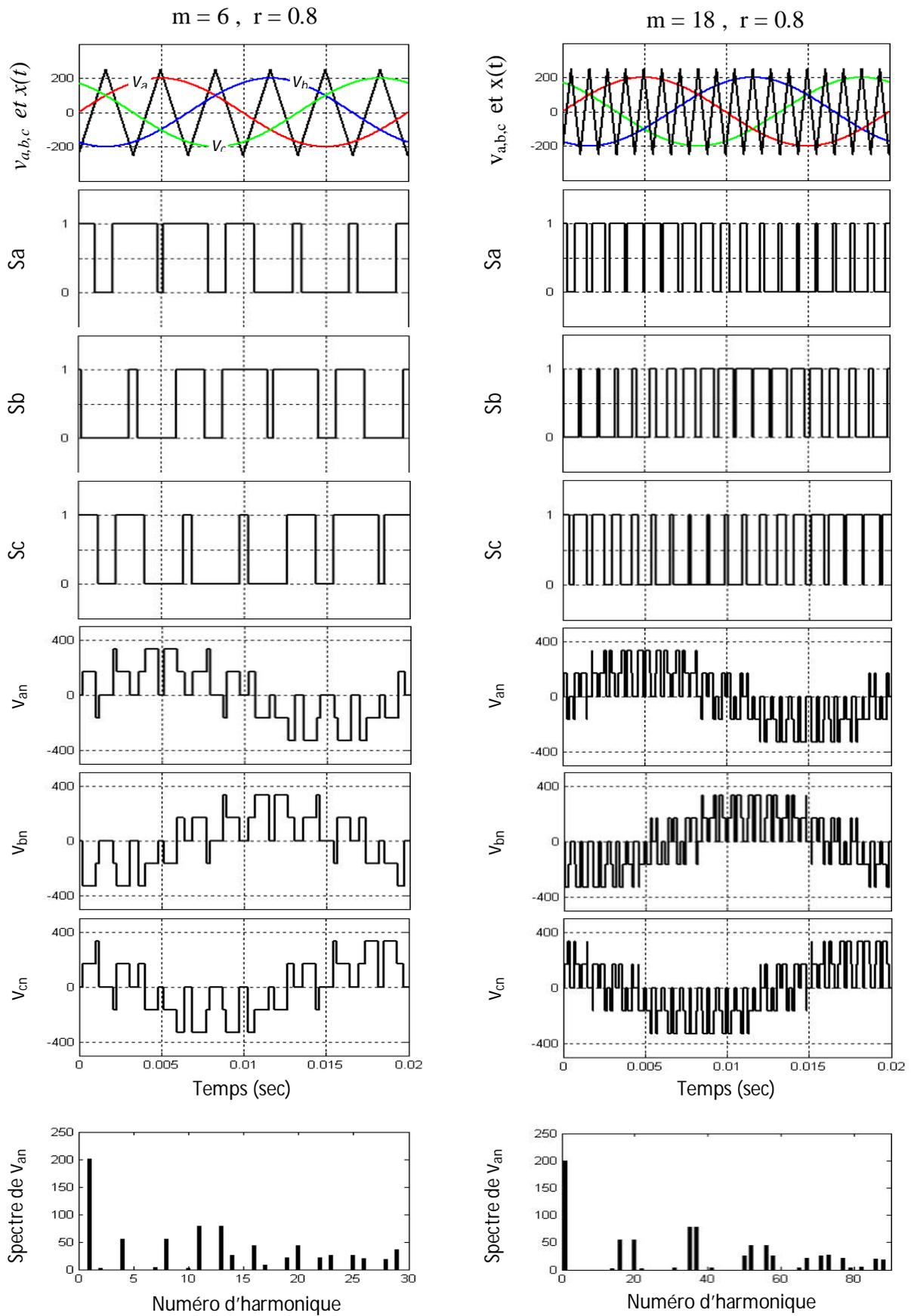


Figure 2.11. Simulation de la commande MLI sinus – triangle pour  $r = 0.8$  et  $m = 6$  et  $18.10$

La simulation montre que l'augmentation de l'indice de modulation rejette les harmoniques de la tension de sortie de l'onduleur vers des fréquences de rangs supérieurs diminuant ainsi l'effet de ces harmoniques sur les performances de la machine qu'il alimente et diminue d'autre part le cycle minimum d'ouverture – fermeture des interrupteurs, qui est de  $0.34\text{ ms}$  pour  $m = 6$  et  $0.1\text{ ms}$  pour  $m = 18$ , augmentant ainsi la perte de commutation par période. Le choix de l'indice de modulation utilisé dans la commande MLI sinus – triangle doit être choisi de manière à faire un compromis entre le rejet des harmoniques vers les hautes fréquences et les pertes de commutations. Un indice de modulation  $m = 21$  semble convenir pour les IGBT qu'on trouve sur le marché.

## **2.8. Conclusion**

Dans ce chapitre on a procédé à la modélisation de l'onduleur à deux niveaux de tension puis on a fait une simulation des performances de cet onduleur en utilisant les commandes MLI sinus triangle. D'après les résultats de simulation obtenue, on peut conclure que Le choix de l'indice de modulation utilisé dans la commande MLI sinus – triangle doit être choisi de manière à faire un compromis entre le rejet des harmoniques vers les hautes fréquences et les pertes de commutations.

# Chapitre 3

## Modélisation et simulation de la Commande Vectorielle

### 3.1. Introduction [12]

Le modèle dynamique de la machine à courant continu à excitation séparée comparé à celui de la machine induction, met en évidence le fait que cette dernière présente une commande plus complexe; ceci tient au fait que :

- Les grandeurs deviennent alternatives à fréquence variable.
- Le système est non linéaire, variant dans le temps et fortement couplé.
- Les grandeurs physiques rotoriques ne sont pas mesurables dans le cas du moteur à cage d'écureuil.

Les raisons précédentes ont été la cause principale de retard enregistrée pour l'émergence de solutions acceptables pour le contrôle de la machine à induction.

En effet, depuis les années 70, Blaschke et Hasse ont proposés une technique de la nouvelle théorie de la commande connue sous le nom du flux orienté ou en anglais (Field Oriented Control) qui est une technique de commande classique pour l'entraînement des machines Asynchrones.

L'idée fondamentale de cette méthode de commande est de ramener le comportement de la machine asynchrone à celui d'une machine à courant continu.

Dans ce qui suit on va développer en premier l'étude théorique de la commande vectorielle à flux rotorique orienté dans le but de l'implémenter sous Simulink.

### 3.2. Principe de la commande vectorielle [12]

Dans le premier chapitre, nous avons défini la transformée de Park nécessaire au changement de coordonnées utilisé pour la commande vectorielle. Une fois dans ce repère, le moteur asynchrone peut être considéré comme un système multi variable sur lequel le vecteur d'entrée est constitué des deux composantes de la tension  $v_{sd}$  et  $v_{sq}$  dans le repère  $d, q$  et des pulsations du champ tournant  $\omega_s$  et du rotor  $\omega_r$ .

La sortie est constituée de l'ensemble des flux et courants au stator et au rotor.

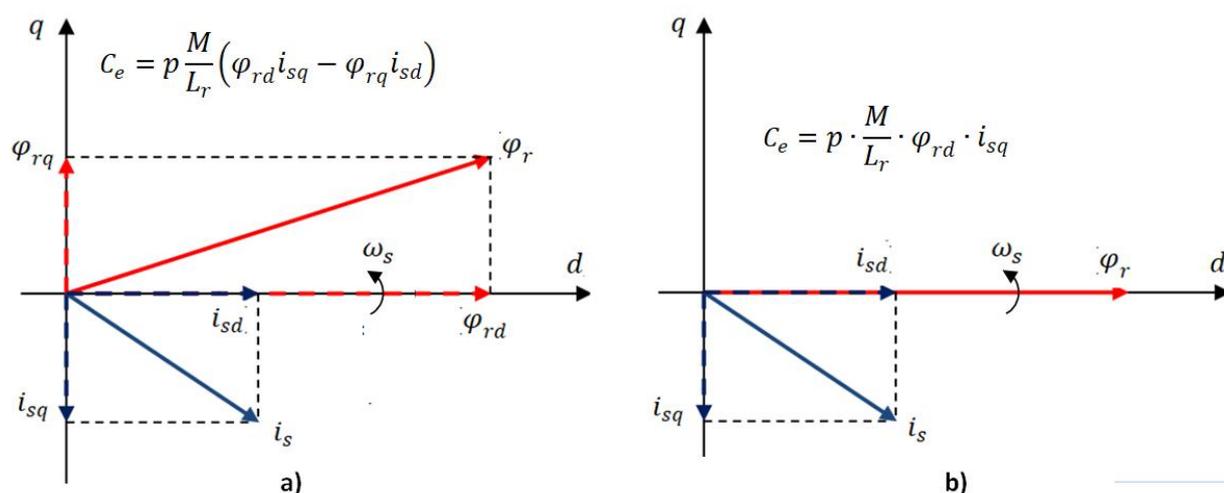
Lorsqu'un moteur électrique entraîne une charge mécanique il est indispensable, pour bien piloter la dynamique de celle-ci, de maîtriser le couple instantané de celui-ci.

L'idée directrice de la commande vectorielle est d'avoir pour la machine asynchrone un couple moteur proportionnel à un flux et un courant comme pour la machine à courant continu.

Ainsi, reprenons l'expression du couple électromagnétique de la machine asynchrone

$$C_e = pM(i_{sq}i_{rd} - i_{sd}i_{rq}) = p \frac{M}{L_r} (\varphi_{rd}i_{sq} - \varphi_{rq}i_{sd}) \quad (3.1)$$

Le repère  $d, q$  dans lequel sont projeté le flux rotorique et le courant statorique tourne à la vitesse du champ tournant  $\omega_s$ .



**Figure 3.1** Projection du flux rotorique et du courant statorique dans le repère  $d, q$ .

Il existe, pour la machine asynchrone, une infinité de positions du repère  $d, q$  tournant à la vitesse  $\theta_s = \omega_s \cdot t$ . Pour celles-ci le flux rotorique et le courant statorique se projettent conformément à la figure 3.1. La position du repère  $d, q$  étant arbitraire, afin d'avoir une expression du couple électromagnétique analogue à celle d'un moteur à courant continu nous orienterons l'axe  $d$  dans la direction du flux rotorique  $\varphi_r$  (figure 3.1.b).

L'expression du couple électromagnétique devient :

$$C_e = p \cdot \frac{M}{L_r} \cdot \varphi_r \cdot i_{sq} \quad (3.2)$$

Pour obtenir cette orientation il faut calculer la pulsation  $\omega_s$ , que l'on intégrera pour calculer l'angle  $\theta_s$ . nécessaire aux transformations de coordonnées. A partir du modèle mathématique de la machine asynchrone on peut exprimer la composante du flux rotorique sur l'axe  $q$  par :

$$\varphi_{rq} = 0 = \frac{MR_r}{L_r} i_{sq} - i_{sq} (\omega_s - \omega_r) \varphi_r \quad (3.3)$$

D'où

$$\omega_s = \omega_r + \frac{MR_r}{L_r} \frac{i_{sq}}{\varphi_r} \quad (3.4)$$

Ainsi si nous commandons correctement les courants statoriques  $i_{sd}$  et  $i_{sq}$  nous maîtriserons le couple de la machine asynchrone. Le courant  $i_{sd}$  permettra de fixer le flux rotorique  $\varphi_r$  et le courant pilotera le couple électromagnétique.

### 3.3. Les différents types de commande vectorielle

Il existe deux types de commande vectorielle, la commande vectorielle directe et la commande vectorielle indirecte. Dans la commande vectorielle directe on effectue une régulation du flux qui nécessite la connaissance de celui-ci, alors que dans la commande vectorielle indirecte, on s'affranchit de la connaissance de ce flux en faisant quelque approximation.

#### 3.3.1. Commande vectorielle directe

Cette méthode nécessite une bonne connaissance du module du flux et sa phase, celle-ci doit être vérifiée quelque soit le régime transitoire effectué. Il faut donc procéder à une série de mesures disponibles au sein du processus. Ce mode de contrôle est ainsi appelé parce qu'une régulation du flux est introduite par une boucle de contre réaction nécessitant la mesure ou l'estimation de ces variations. Dans son principe, la commande vectorielle directe doit être réalisée par la mesure du flux qui nécessite l'utilisation du capteur placé dans l'entrefer de la machine, opération généralement délicate à réaliser.

### 3.3.2. Commande vectorielle indirecte

Cette méthode est ainsi appelée car on ne fait pas appel à une boucle de régulation du flux, et par conséquent elle ne nécessite pas l'utilisation du capteur ou d'estimation du flux.

Dans cette commande le vecteur du flux rotorique est estimé à partir de la mesure de la vitesse du rotor, en utilisant un bloc de défluxage et le modèle mathématique de la commande vectorielle est obtenu en se basant sur le modèle mathématique du moteur asynchrone dans un système de référence tournant en synchronisme avec le vecteur du flux rotorique.

Dans notre travail on va utiliser la commande vectorielle indirecte.

### 3.4. Mise en équation de la commande vectorielle indirecte

Rappelons que dans un repère lié au champ tournant, les équations des tensions statoriques et rotoriques de la MAS sont les suivantes:

$$\begin{cases} v_{sd} = R_s i_{sd} + \frac{d\varphi_{sd}}{dt} - \omega_s \varphi_{sq} \\ v_{sq} = R_s i_{sq} + \frac{d\varphi_{sq}}{dt} + \omega_s \varphi_{sd} \end{cases} \quad (3.5)$$

$$\begin{cases} 0 = R_r i_{rd} + \frac{d\varphi_{rd}}{dt} - \omega_{gl} \varphi_{rq} \\ 0 = R_r i_{rq} + \frac{d\varphi_{rq}}{dt} + \omega_{gl} \varphi_{rd} \end{cases} \quad (3.6)$$

La mise en œuvre de la commande vectorielle à flux rotorique orienté est basée sur l'orientation du repère tournant d'axes  $d, q$ , tel que l'axe  $d$  soit confondu avec la direction de  $\varphi_r$  (figure 3.1.b).

Alors :

$$\begin{cases} \varphi_{rd} = \varphi_r \\ \varphi_{rq} = 0 \end{cases} \quad (3.7)$$

L'équation 1.17 du chapitre 01 donne les équations des flux statoriques :

$$\begin{cases} \varphi_{sd} = \sigma L_s i_{sd} + \frac{M}{L_r} \varphi_r \\ \varphi_{sq} = \sigma L_s i_{sq} \end{cases} \quad (3.8)$$

En remplaçant dans l'équation 3.5 on obtient les composantes des tensions statoriques :

$$\begin{cases} v_{sd} = R_s i_{sd} + \sigma L_s \frac{di_{sd}}{dt} + \frac{M}{L_r} \frac{d\varphi_r}{dt} - \omega_s \sigma L_s i_{sq} \\ v_{sq} = R_s i_{sq} + \sigma L_s \frac{di_{sq}}{dt} + \omega_s \frac{M}{L_r} \varphi_r + \omega_s \sigma L_s i_{sd} \end{cases} \quad (3.9)$$

Les composantes des tensions rotoriques sont données par :

$$\begin{cases} v_{rd} = 0 = R_r i_{rd} + \frac{d\varphi_r}{dt} \\ v_{rq} = 0 = R_r i_{rq} + \omega_{gl} \varphi_r \end{cases} \quad (3.10)$$

Des équations 1.16 du chapitre 01 on obtient :

$$\begin{cases} L_r i_{rd} + M i_{sd} = \varphi_r \\ L_r i_{rd} + M i_{sq} = 0 \end{cases} \quad (3.11)$$

Les équations 3.10 et 3.11 nous donnent :

$$\omega_{gl} = \frac{M}{T_r \varphi_r} i_{sq} \quad (3.12)$$

$$\frac{L_r}{R_r} \frac{d\varphi_r}{dt} + \varphi_r = M i_{sd} \quad (3.13)$$

$$i_{sd} = \frac{1}{M} \left( \frac{L_r}{R_r} \frac{d\varphi_r}{dt} + \varphi_r \right) \quad (3.14)$$

En utilisant l'expression du couple de l'équation 3.1 avec  $\varphi_{rd} = \varphi_r$  et  $\varphi_{rq} = 0$  on obtient :

$$C_e = p \cdot \frac{M}{L_r} \cdot \varphi_r \cdot i_{sq} \quad (3.15)$$

D'où l'expression de la composante en quadrature du courant statorique :

$$i_{sq} = \frac{L_r}{pM} \cdot \frac{C_e}{\varphi_r} \quad (3.16)$$

En régime permanent les dérivées de  $\varphi_r$ , de  $i_{sd}$  et de  $i_{sq}$  sont nulles et les équations (3.16) et 3.10 deviennent :

$$i_{sd} = \frac{\varphi_r}{M} \quad (3.17)$$

L'équation (3.11) devient :

$$\begin{cases} v_{sd} = R_s i_{sd} - \sigma L_s \omega_s i_{sq} \\ v_{sq} = R_s i_{sq} + L_s \omega_s i_{sd} \end{cases} \quad (3.18)$$

La figure (3.2) montre le schéma bloc de la commande vectorielle indirecte. Les équations (3.12) et (3.18) – (3.20) sont utilisées pour concevoir le bloc de la commande vectorielle indirecte.

La vitesse angulaire électrique  $\omega_s$  de l'axe direct (pulsation statorique) est calculée en utilisant la loi d'autopilotage  $\omega_s = \omega_m + \omega_{gl}$  où  $\omega_m = p\Omega$  est la vitesse angulaire électrique du rotor. La position  $\theta_s$  du flux rotorique utilisée dans la transformation de Park est obtenue par intégration de  $\omega_s$ . Le module du flux rotorique est obtenu en utilisant la relation de défluxage suivante :

$$\varphi_r = \begin{cases} \varphi_{rn} & \text{si } |\Omega| \leq \Omega_n \\ \varphi_{rn} \cdot \frac{\Omega_n}{|\Omega|} & \text{si } |\Omega| > \Omega_n \end{cases} \quad (3.19)$$

Où  $\varphi_{rn}$  est le flux de référence nominal,  $\Omega$  est la vitesse mécanique et  $\Omega_n$  est sa valeur nominale.

Le défluxage assure un flux rotorique constant pour un fonctionnement normal de la machine et une diminution hyperbolique pour un fonctionnement en survitesse.

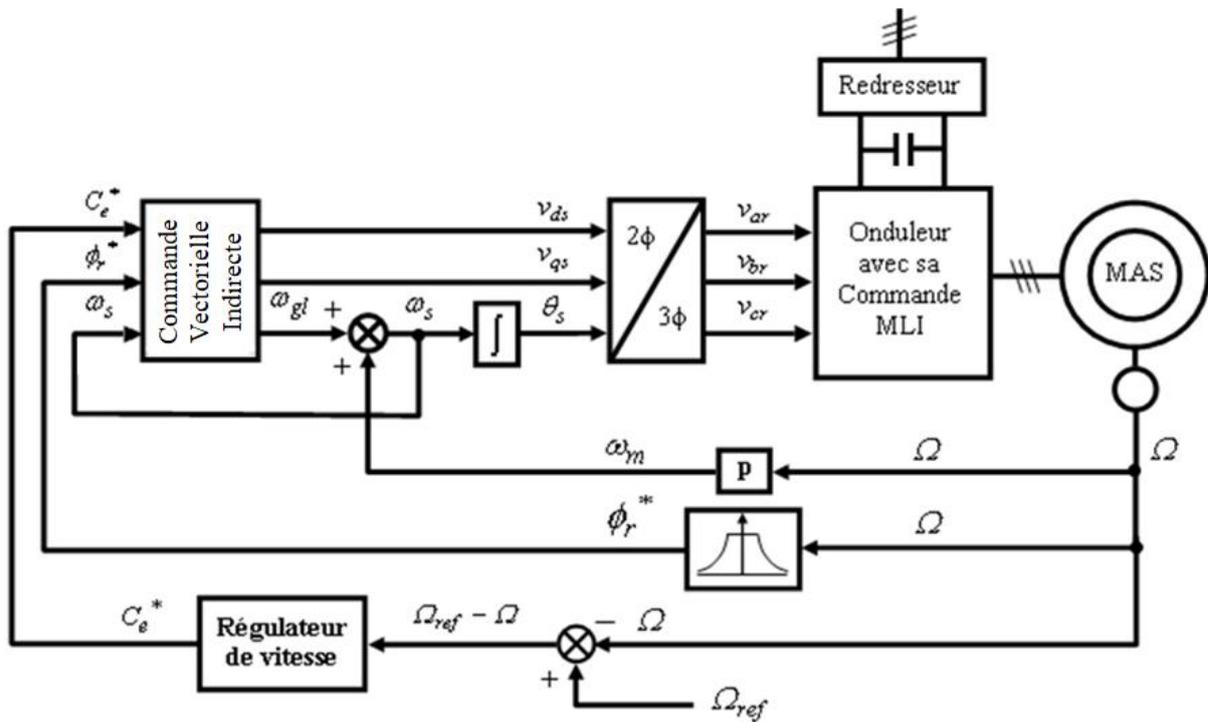
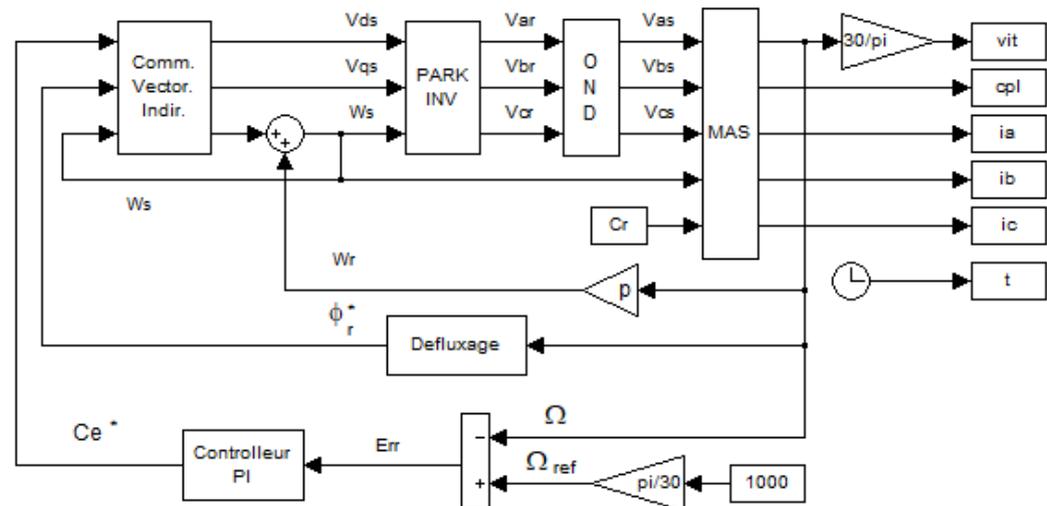


Figure 3.2. Schéma bloc de la régulation de vitesse d'une machine asynchrone par commande vectorielle indirecte.

### 3.5. Simulation de la Commande Vectorielle d'un moteur asynchrone

Après avoir réalisé les modèles mathématiques des différents éléments utilisés dans la régulation de vitesse d'une machine asynchrone par commande vectorielle indirecte, une simulation est nécessaire afin d'étudier les performances de cette commande.

La figure 3.3 montre le modèle Simulink de la régulation de vitesse d'une machine asynchrone par commande vectorielle indirecte en utilisant le logiciel Simulink de Matlab.



Sous système Comm. Vect. Indir.

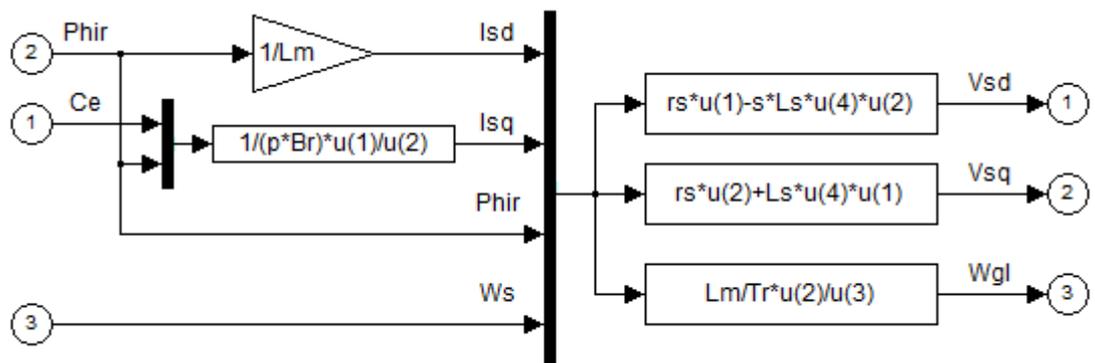
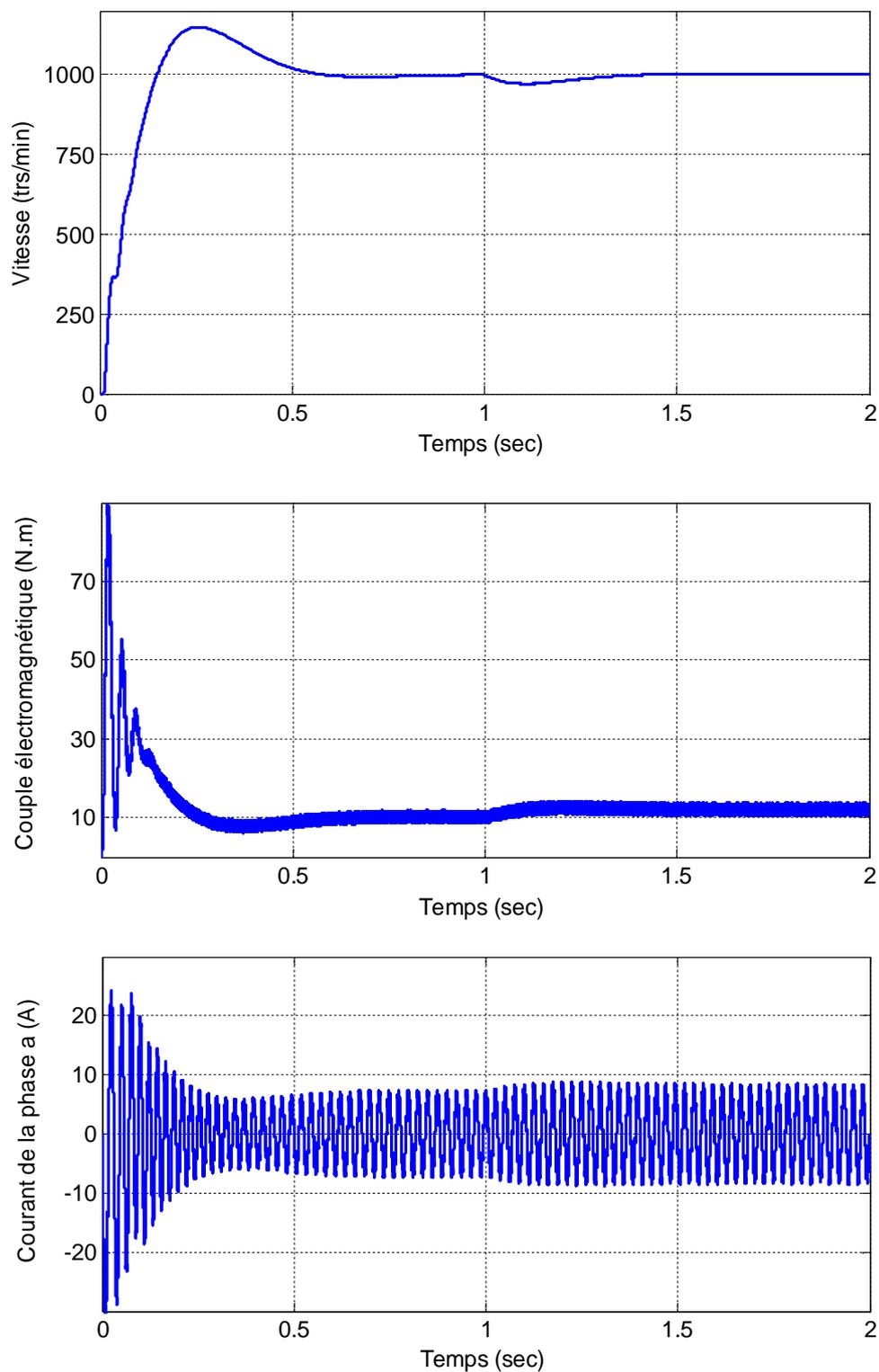


Figure 3.3. Modèle Simulink de la commande vectorielle d'une machine asynchrone.

La figure 3.4 montre la simulation des performances dynamiques de la commande vectorielle d'une machine asynchrone avec une charge de 10 N.m au démarrage et avec une perturbation de charge de +2 N.m au temps  $t = 1$  sec. Les paramètres de la machine utilisée sont donnés dans l'annexe. Les paramètres du régulateur PI utilisés ont été trouvés par la méthode des placements des pôles et sont données par  $K_p = 0.4329$  et  $K_i = 3.1$ . La vitesse de référence a été fixée à 1000 trs/min.

Les résultats de simulation montre que la vitesse de la machine asynchrone ainsi commandée fait un dépassement de 14.9% avec un temps de réponse 5% supérieur à 0.4 secs. La perturbation de charge est rejetée totalement en 0.7 secs avec une chute de vitesse de 29 trs/min.



**Figure 3.4.** Performance de la commande vectorielle d'une machine asynchrone.

On remarque que même avec la méthode des placements des pôles pour la recherche des paramètres du régulateur PI, le dépassement est assez important (presque 15%), Si on veut diminuer le dépassement, on doit diminuer l'action intégrale et dégrader la réaction du régulateur face aux perturbations de la charge.

### **3.6. Conclusion**

Dans ce chapitre on a procédé à la modélisation de la commande vectorielle indirecte d'une machine asynchrone. Un modèle Simulink a été réalisé et une simulation a été faite pour étudier les performances de cette commande. D'après les résultats de simulation obtenus, on peut conclure que la commande vectorielle de la MAS permet de maintenir la vitesse de la machine collé à sa référence même dans le cas des perturbations de charge. Le dépassement au démarrage est acceptable, la réponse au démarrage et le temps de rejet des perturbations sont assez rapide.

Dans le chapitre 04 on va introduire le contrôleur PI-PSO qu'on va implémenter dans la commande vectorielle d'une machine asynchrone dans le chapitre 05.

# Chapitre 4

Etude, modélisation et simulation de la  
technique PSO pour l'optimisation des  
paramètres d'un contrôleur PI

### 4.1. Introduction [13]

L'optimisation par essaim particulaire (OEP), ou Particle Swarm Optimzation (PSO) en anglais, est un algorithme évolutionnaire qui utilise une population de solutions candidates pour développer une solution optimale à un problème donné. Cet algorithme a été proposé par Russel Eberhart (ingénieur en électricité) et James Kennedy (socio-psychologue) en 1995. Il s'inspire à l'origine du monde du vivant, plus précisément du comportement social des animaux évoluant en essaim, tels que les bancs de poissons et les vols groupés d'oiseaux. En effet, on peut observer chez ces animaux des dynamiques de déplacement relativement complexes, alors qu'individuellement chaque individu a une « intelligence » limitée, et ne dispose que d'une connaissance locale de sa situation dans l'essaim.

L'essaim de particules correspond à une population d'agents simples, appelés particules. Chaque particule est considérée comme une solution du problème donné, où elle possède une position (Vecteur solution) et une vitesse.

Ce comportement social basé sur l'analyse de l'environnement et du voisinage constitue alors une méthode de recherche d'optimum par l'observation des tendances des individus voisins. Chaque individu cherche à optimiser ses chances en suivant une tendance qu'il modère par ses propres vécus.

### 4.2. Mode de fonctionnement de l'algorithme PSO

La population dans l'algorithme PSO est nommée essaim, chaque individu du groupe est dit particule. Le déplacement de toute particule (comme indiqué ci-dessus) est régi par des règles et conditions bien spécifiques, influencé par le mouvement des autres particules du voisinage. Dans un tel contexte ce déplacement à une signification et doit parallèlement répondre à une logique, fondement même du PSO. Il consiste à chercher un optimum dans un voisinage donné, ce déplacement est influé par les trois composantes suivantes :

- **Une composante d'inertie** : la particule s'efforce de suivre instinctivement son cap de déplacement.
- **Une composante cognitive** : la particule fait tout pour se diriger vers la meilleure position rencontrée jusqu'à présent.

- **Une composante sociale** : la particule s'inspire également de l'expérience, du parcours des autres particules, pour se diriger vers la meilleure position rencontrée par ses voisins.

Ce déplacement est illustré dans la Figure 4.1.

A terme, on se rend compte que toutes les particules, après un certain nombre d'itérations, convergent vers une même position qui est la meilleure rencontrée par l'ensemble des particules. Cette position n'est pas pour autant la meilleure position du voisinage.

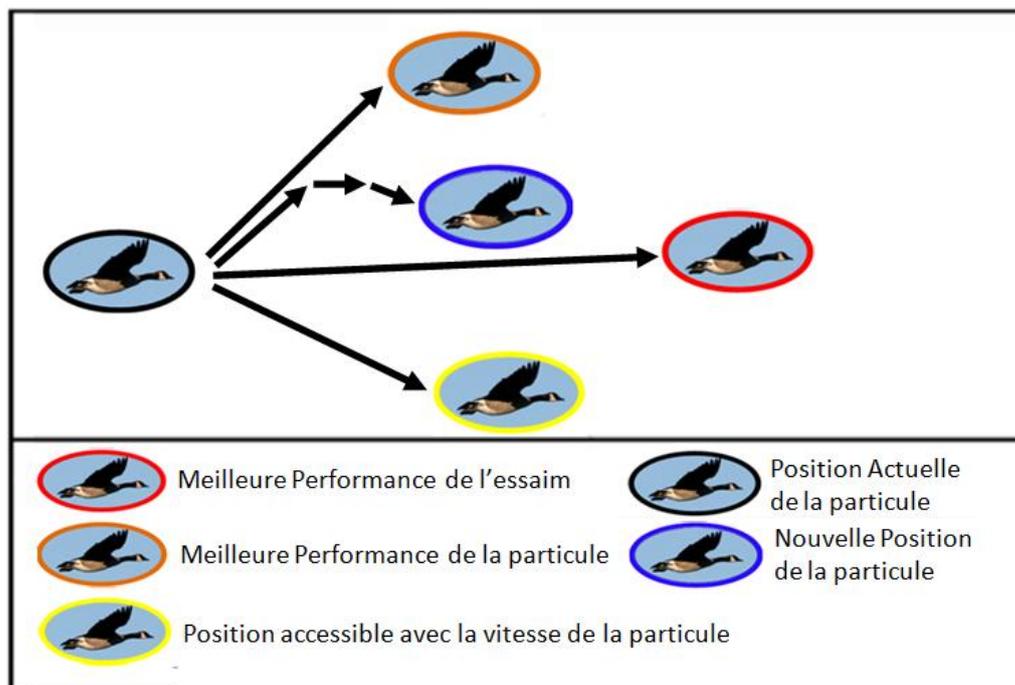


Figure 4.1. Déplacement d'une particule.

### 4.3. Principe de fonctionnement du PSO [13-19]

Un essaim est disposé de façon aléatoire et homogène dans l'espace de recherche et chaque particule possède la capacité de se déplacer avec une vitesse aléatoire. Ainsi, à chaque pas de temps, chaque particule :

- Évalue la qualité de sa position et garde en mémoire sa meilleure performance, c'est-à-dire la meilleure position atteinte jusqu'ici (elle peut être la position courante) et sa qualité (la valeur de la fonction à optimiser en cette position).
- Interroge un certain nombre de particules pour obtenir de chacune d'entre elles sa propre meilleure performance.

- Choisit la meilleure des meilleures performances dont elle a connaissance, puis adapte sa vitesse en fonction de cette information et de ses propres données et se déplace en conséquence.

Le principe de la méthode d'essaim de particule est résumé par la figure(4.1). Pour réaliser son prochain mouvement, chaque particule combine trois tendances : suivre sa vitesse propre, revenir vers sa meilleure performance, aller vers la meilleure performance de ses informatrices (l'essaim de particules).

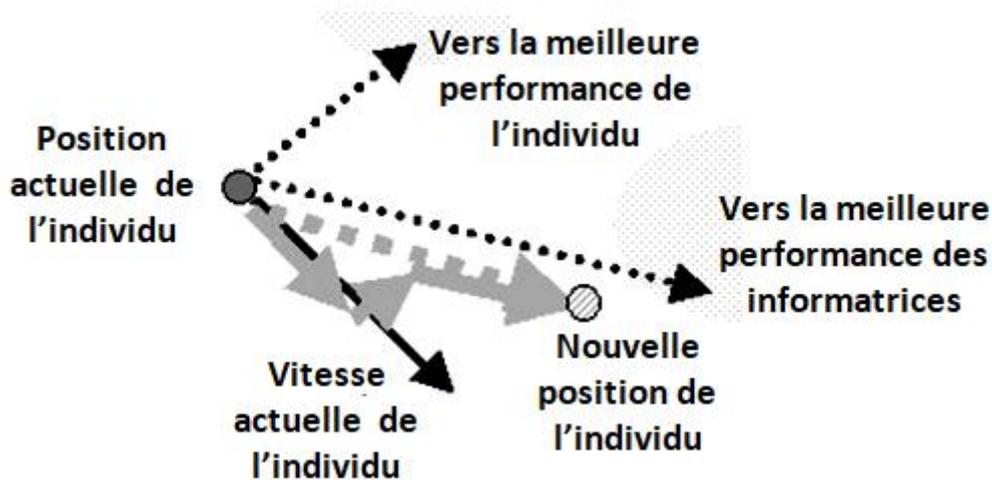


Figure 4.2. Schéma de principe du déplacement d'une particule.

Dans un espace de recherche de dimension D la particule i de l'essaim est modélisée par son vecteur position  $\vec{x}_{ij} = (x_{i1}x_{i2}.....x_{id})^T$ .et par son vecteur vitesse  $\vec{v}_{ij} = (v_{i1}v_{i2}.....v_{id})^T$ .

La qualité de sa position est déterminée par la valeur de la fonction objective (cout) en ce point. Cette particule garde en mémoire la meilleure position par laquelle elle est déjà passée, que l'on note  $\vec{p}_{ij} = (p_{i1}p_{i2}.....p_{id})^T$ .La meilleure position atteinte par ses particules voisines est notée  $\vec{g}_{ij} = (g_{i1}g_{i2}.....g_{id})^T$ .

Le vecteur vitesse est calculée à partir de l'équation (4.1) :

$$\vec{v}_{ij} = w. v_{ij}(k - 1) + c_1r_1 (p_{ij}(k - 1) - x_{ij}(k - 1)) + c_2r_2 (g_{ij}(k - 1) - x_{ij}(k - 1)) \tag{4.1}$$

La position à l'itération  $k$  de la particule  $i$  est alors définie par l'équation (4.2) :

$$x_{ij}(k) = x_{ij}(k-1) + v_{ij}(k) \quad (4.2)$$

Avec  $i=1,2,3,\dots,N_p$  ;  $j=1,2,3,\dots,N_d$  ;  $k=1, 2, 3,\dots,K_{max}$  Et sachant que :

$w \cdot v_{ij}(k-1)$  : correspond à la composante physique du déplacement. Le paramètre  $w$  contrôle l'influence de la direction de déplacement sur le déplacement futur.

$c_1 r_1 (p_{ij}best - x_{ij}(k-1))$  Correspond à la composante cognitive du déplacement ou  $c_1$  contrôle de comportement cognitif de la particule.

$c_2 r_2 (g_{ij}best - x_{ij}(k-1))$  Correspond à la composante sociale du déplacement ; ou contrôle l'aptitude sociale de la particule.

#### 4.4. Mécanisme de confinement [15-19]

Il arrive parfois que compte tenu de la position courante et de la vitesse courante d'une particule, cette dernière a tendance à sortir de l'espace de recherche lors de son déplacement. Lorsque c'est le cas, l'algorithme fait intervenir un mécanisme de confinement, dit confinement d'intervalle, afin de gérer le déplacement de la particule et ce, de manière à ce que celui-ci l'amène sur un nouveau point appartenant lui aussi à l'espace de recherche.

De manière générale, ce mécanisme consiste à ramener la particule dans le point admissible « le plus proche ».

$$\begin{cases} \text{if } x_{ij} > x_{jmax} \text{ alors } x_{ij} = x_{jmax} \\ \text{if } x_{ij} < x_{jmin} \text{ alors } x_{ij} = x_{jmin} \\ v_{ij} = 0 \end{cases} \quad (4.3)$$

Où :  $x_{jmax}$  et  $x_{jmin}$  sont les valeurs limites du paramètre  $x_{ij}$ .

Dans une autre part, le coefficient d'inertie joue un rôle important dans la procédure de recherche car elle garantit un équilibre entre la recherche locale et la recherche globale, en 2002 M. Clerc et J. Kennedy ont présenté une autre approche de calcul qui introduit un coefficient de construction qui permet de contrôler la vitesse au cours de l'itération et donc au

paramètre de convergence et de nouveaux paramètres relatifs aux composantes cognitive et sociale  $\phi_1$  et  $\phi_2$  ont ainsi été introduit selon l'équation 4.4 et 4.5

$$x = \frac{2 \cdot k}{|2 - \phi - \sqrt{(\phi^2 - 4\phi)}|} \quad (4.4)$$

$$\phi = \phi_1 + \phi_2 \geq 4; \text{ et } \phi_1 = \phi_2 = 2.05 \quad (4.5)$$

Et les paramètres de l'équation (4.1) deviennent :

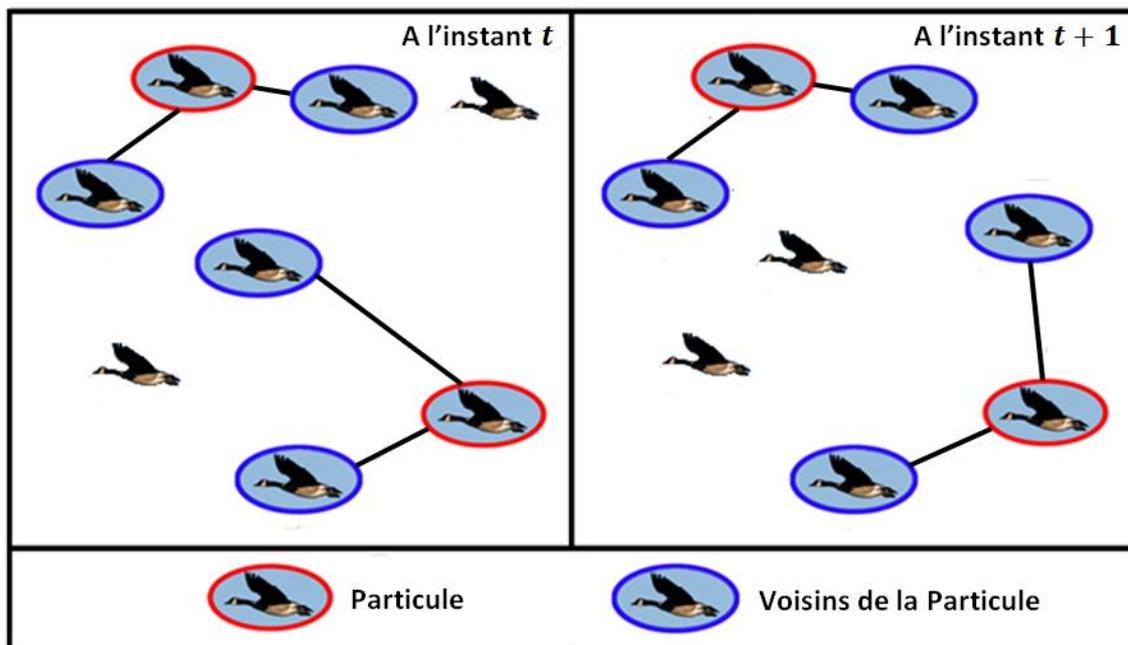
$$\begin{cases} w = x \\ c_1 = x \cdot \phi_1 \\ c_2 = x \cdot \phi_2 \end{cases} \quad (4.6)$$

A partir de l'équation (4.4) on peut éliminer la vitesse  $v_{max}$ .

#### 4.5 Notion de voisinage

Chaque particule dispose d'un sous-ensemble d'autres particules avec lequel elle est en interaction, c'est le voisinage de la particule. Cet entrelacement de rapports entre toutes les particules est assimilé à la sociométrie ou à la topologie de l'essaim. On dénombre deux types de voisinage :

- **Voisinage géographique** : c'est un voisinage dynamique où les voisins sont les particules les plus proches. A chaque itération, les nouveaux voisins ou groupes doivent être réajustés en se référant à une distance prédéfinie dans l'espace de recherche.



**Figure 4.3.** Voisinage géographique à l' instant  $t$  et  $t+1$ .

Ceci est illustré par la figure 4.3. où le voisinage d'une particule est un groupe formé des deux particules les plus proches. Dans cette figure la notion de voisinage dynamique est mise en évidence puisque pour un même essaim à l'instant «  $t$  » et à l'instant «  $t+1$  » le voisinage n'est plus le même.

- **Voisinages sociaux :**

Ce type de voisinage est considéré comme statique, les voisins restent figés, autrement-dit, ils demeurent inchangés. C'est le voisinage auquel on a le plus souvent recours, en raison :

- ✓ De sa simplicité de programmation.
- ✓ Parce qu'il offre un meilleur rapport temps/coût, en termes de calcul.
- ✓ Dans un scénario de convergence, un voisinage social s'oriente forcément vers un voisinage géographique.

#### 4.6. Conception et déroulement de l'algorithme [15-19]

Pour expliquer le principe de l'algorithme PSO appliqué pour résoudre un problème de minimisation ou de maximisation, considérons le problème d'optimisation:

$$\min\{f(x_j)\}, \quad j = 1, 2, 3, \dots, d \quad (4.7)$$

La fonction forme associée est :

$$\text{forme}(x_j) = f(x_j) \quad (4.8)$$

L'algorithme de PSO se déroule suivant les étapes suivantes:

**Etape 1:**

Initialiser une population de particules et de vitesses, uniformément réparties dans l'espace de recherche, et fixer le critère d'arrêt.

**Etape 2 :**

Évaluer la fonction objective, pour chaque particule.

**Etape 3 :**

Mettre à jour la meilleure position  $p_{best}$  pour chaque particule et la meilleure position globale  $g_{best}$  dans la population.

**Etape 4 :**

Mettre à jour la position et la vitesse utilisant les équations 4.1 et 4.2

**Etape 5 :**

Si une particule, déplacée par l'algorithme vers la limite ou à l'extérieur de l'espace de recherche, le mécanisme de confinement des particules exprimé par les équations (4.3) intervient et ramène la particule à l'intérieur de l'espace de recherche.

**Etape 6 :**

Vérifier le critère d'arrêt. Si le critère d'arrêt n'est pas satisfait, aller vers l'étape 2 ; sinon, le programme se termine, et la solution optimale est produite.

L'algorithme de la figure (4.4) résume ces étapes :

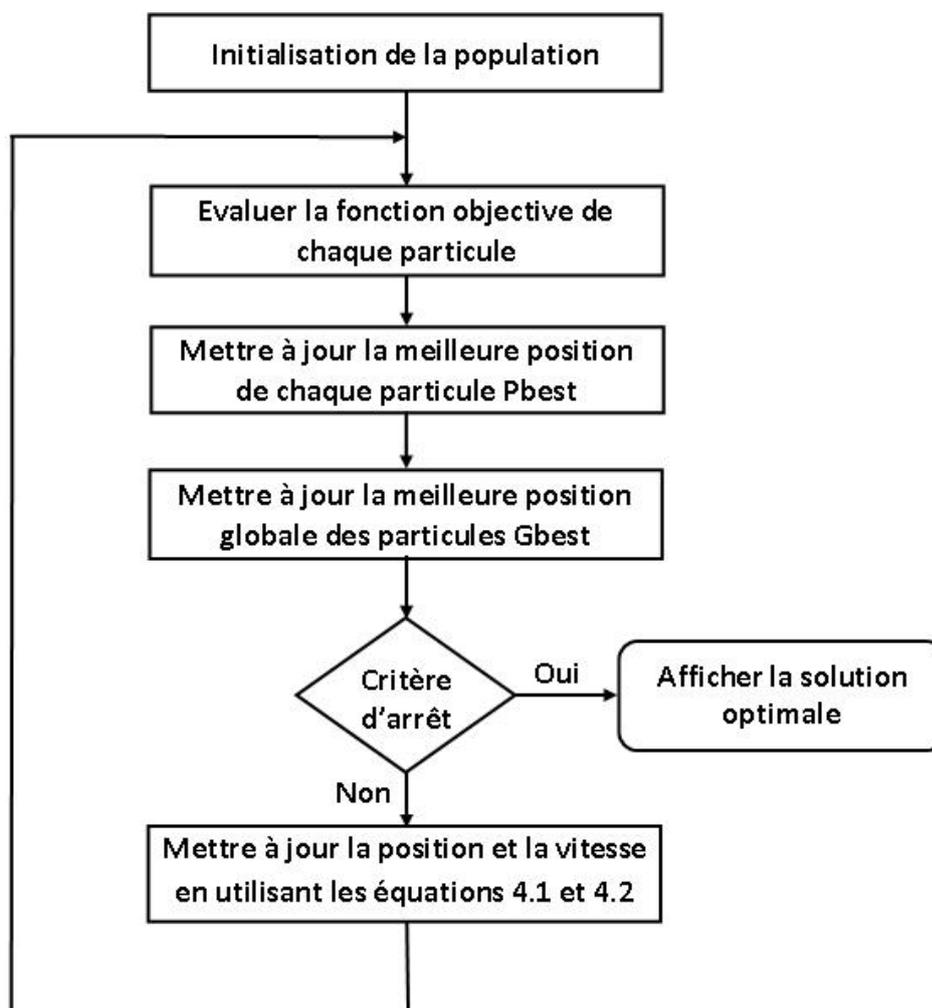


Figure4.4.Organigramme générale du PSO.

#### 4.7. Avantages et inconvénients de PSO

L'algorithme du PSO présente un avantage certain sur les méthodes classiques dans le sens où il permet une exploration aléatoire de l'espace de solution. De plus de sa simplicité à mettre en œuvre et son exécution conduit à l'obtention de très bon résultat, rapidement et facilement avec peu de paramètre à ajuster.

Bien que le PSO trouve de bonnes solutions dans un temps beaucoup plus court que d'autres algorithmes évolutionnaires, l'amélioration de la qualité des solutions ne peut être garantie en augmentant le nombre d'itération.

L'algorithme de PSO a deux inconvénients principaux :

- Le premier inconvénient est que l'essaim peut prématurément converger.

- Le deuxième inconvénient est que les approches stochastiques ont un problème de dépendance, tout changement d'un de leurs paramètres peut avoir un effet sur le fonctionnement de l'algorithme tout comme sur la solution obtenue.

#### 4.8. Exemple d'Application de l'algorithme PSO

Dans cette section on va donner un exemple d'application de l'algorithme PSO. Il s'agit d'utiliser l'algorithme PSO pour résoudre le problème de minimisation de la fonction  $f(x_1, x_2, x_3)$  donnée par ;

$$f(x_1, x_2, x_3) = 10(x_1 - 1)^2 + 20(x_2 - 1)^2 + 30(x_3 - 1)^2 \quad (4.9)$$

La fonction forme associée est :

$$\text{forme } (x_1, x_2, x_3) = 10(x_1 - 1)^2 + 20(x_2 - 1)^2 + 30(x_3 - 1)^2 \quad (4.10)$$

Un programme Matlab a été écrit en utilisant deux fichiers, un fichier principal nommé run\_pso.m et une fonction Objfun.m.

Le programme run\_pso recherche dans une grille de valeur de  $(x_1, x_2, x_3)$  allant de 0 à 10, les valeurs qui donnent la plus petite valeur de la fonction forme .

Voici le programme de la fonction objective :

```
function y=Objfun(x)
y=10*(x(1)-1)^2+20*(x(2)-2)^2+30*(x(3)-3)^2;
```

Le programme run\_pso, lance la fonction Objfun avec différentes valeurs de  $(x_1, x_2, x_3)$  en se déplaçant dans l'espace  $(x_1, x_2, x_3)$  en suivant la méthode de déplacement d'un essaim de particules (C.-à-d. suivant l'algorithme PSO), en utilisant plusieurs itérations et plusieurs essais et en sauvegardant à chaque itération les valeurs de  $(x_1, x_2, x_3)$  qui donnent la plus petite valeur de  $y$ . A la fin du programme il affiche les valeurs optimales de  $(x_1, x_2, x_3)$  et la valeur minimale de la fonction  $f(x_1, x_2, x_3)$ .

En exécutant le programme run\_pso avec 100 particules, on a obtenu le résultat suivant après 15 itérations.

Itération	Particule	x(1)	x(2)	x(3)	Fonction	Forme
1	63	0.0525	1.8670	3.0400	9.3798	
2	32	0.7913	2.4079	3.3507	7.4530	
3	32	0.7913	2.4079	3.3507	7.4530	
4	32	0.7913	2.4079	3.3507	7.4530	
5	70	1.3832	2.3584	3.1776	4.9827	
6	25	1.1906	2.2840	3.0275	1.9988	
7	25	1.1906	2.2840	3.0275	1.9988	
8	25	1.1906	2.2840	3.0275	1.9988	
9	7	1.1105	2.2679	2.9049	1.8289	
10	2	1.1664	2.0490	2.9403	0.4321	
11	60	1.0101	1.9931	2.8949	0.3334	
12	81	0.9603	2.0356	3.0187	0.0517	
13	81	0.9603	2.0356	3.0187	0.0517	
14	95	1.0083	1.9961	3.0083	0.0031	
15	95	1.0083	1.9961	3.0083	0.0031	

-----

Meilleure valeur de la fonction forme =0.0031

Meilleure valeur de x(1) =1.0083

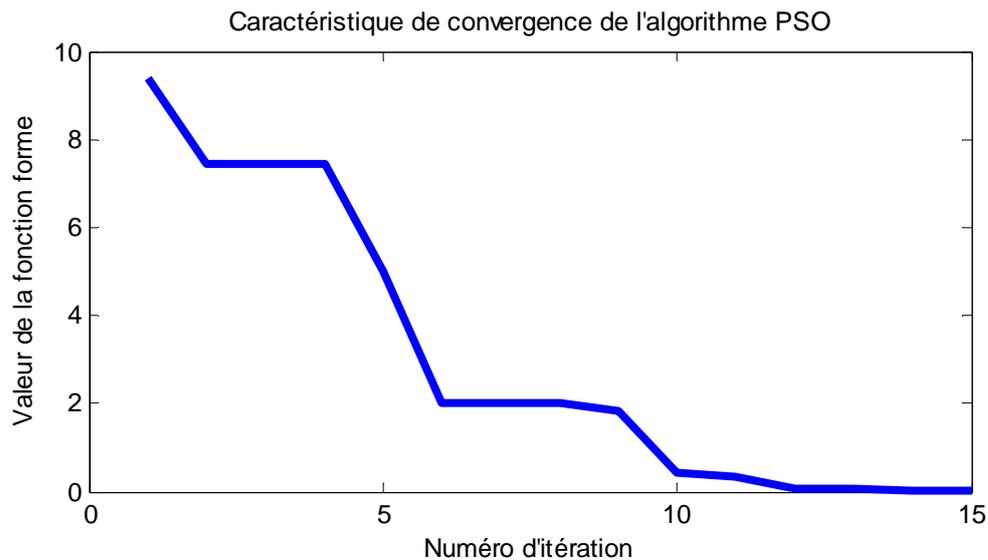
Meilleure valeur de x(2) =1.9961

Meilleure valeur de x(3) =3.0083

Temps d'exécution : 0.153532 secondes.

Le résultat exact est  $(x_1, x_2, x_3) = (1, 2, 3)$ . L'algorithme PSO a réussi à trouver une solution approchée avec une erreur de 0.0031 en 15 itérations et 0.153 s

La figure 4.5 montre la caractéristique de convergence de l'algorithme PSO pour cette application.



**Figure 4.5.** Caractéristique de convergence de l'algorithme PSO.

#### 4.9. Conclusion

Dans ce chapitre nous avons présenté l'optimisation par essaim particulaire. On a d'abord présenté son principe de fonctionnement avant de parler du mécanisme de confinement pour maintenir la population de particules à l'intérieur de l'espace de recherche, ainsi que de la notion de voisinage. On a ensuite fait la modélisation mathématique permettant la conception de l'algorithme avant de faire la simulation de l'algorithme appliquée à la recherche la valeur minimale d'une fonction à trois variables.

Dans le chapitre qui va suivre, on va appliquer l'algorithme PSO pour la recherche des paramètres optimaux d'un régulateur PI utilisé pour la régulation de la vitesse dans la commande vectorielle d'un moteur asynchrone.

# Chapitre 5

Modélisation et simulation d'un  
contrôleur PI-PSO pour la régulation de la  
vitesse d'un moteur asynchrone

### 5.1. Introduction

Dans le chapitre précédent on a fait la modélisation de l'algorithme PSO qu'on a appliqué pour rechercher le minimum d'une fonction à trois variables ainsi que les valeurs des variables qui minimisent la fonction. Dans ce chapitre on va associer l'algorithme PSO à un régulateur PI pour concevoir un contrôleur PI-PSO qui va chercher les paramètres  $K_p$  et  $K_i$  du régulateur PI qui permettent d'obtenir la valeur minimale de l'erreur entre la vitesse réelle et la consigne dans la commande vectorielle d'un moteur asynchrone. Deux méthode de recherche les paramètres  $K_p$  et  $K_i$  vont être utilisée, une méthode off - line et une méthode on - line. On va commencer par appliquer ces méthode à la commande d'un moteur à courant continu avant de terminer par l'application du contrôleur PI-PSO à la régulation de la vitesse d'un moteur asynchrone. Mais avant cela on va faire une introduction au régulateur PI.

### 5.2. Régulateur PI

Le régulateur à action proportionnelle intégrale noté PI couvre une large gamme d'applications industrielles. Il délivre une commande au processus à partir de l'erreur entre la consigne et la sortie régulée  $e = w - y$ .

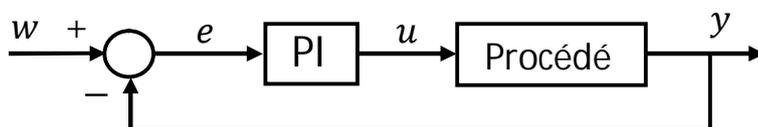


Figure 5.1. Correction d'un processus avec un régulateur PI.

La fonction de transfert d'un régulateur PI est donnée par :

$$C(s) = K_p + \frac{K_i}{s} \quad (5.1)$$

Et la commande obtenue est donnée par :

$$u = K_p e(t) + K_i \int_0^t e(\tau) d\tau \quad (5.2)$$

Où :

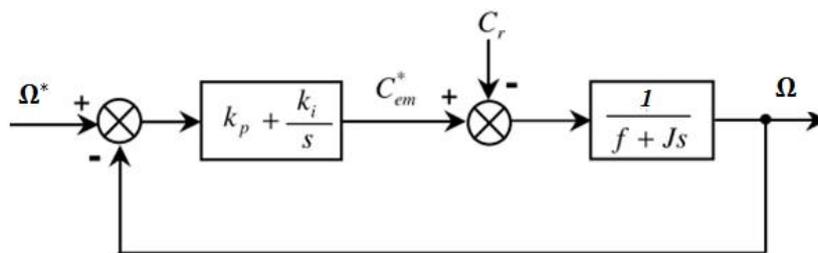
- $K_p$  est le gain de l'action proportionnelle. Si le gain  $K_p$  augmente, l'erreur statique et le temps de réponse sont réduits.

## Chapitre 5 Conception d'un contrôleur PI-PSO pour contrôler la vitesse d'un MAS

- $K_i$  est le gain de l'action intégrale qui permet d'éliminer l'erreur statique. Si le gain  $K_i$  augmente, le système élimine l'erreur statique et rejette l'effet des perturbations plus rapidement, mais le dépassement sur la sortie s'amplifie.

### 5.2.1. Recherche des paramètres du régulateur PI par la méthode de placement des pôles

Considérons la régulation de vitesse d'un moteur par un régulateur PI donnée par Fig.5.2.



**Figure 5.2.** Boucle de régulation de vitesse.

Le calcul des paramètres du régulateur par la méthode du placement des pôles se fait de la manière suivante :

On considère le système corrigé comme étant du deuxième ordre avec deux pôles complexes conjugués donnés en fonction du coefficient d'amortissement  $\xi$  et de la pulsation naturelle  $\omega_n$  du système.

Comme les pôles sont fonction de  $\xi$  et  $\omega_n$ , on place les pôles en choisissant  $\xi$  et  $\omega_n$  qui donnent un amortissement optimal avec un dépassement minimal et un rapide rejet des perturbations de charge.

La fonction de transfert boucle fermée du système corrigé est donnée par :

$$\frac{\Omega(s)}{\Omega^*(s)} = \frac{\frac{1}{J}(K_p s + K_i)}{s^2 + \frac{(K_p + f)}{J}s + \frac{K_i}{J}} \quad (5.3)$$

Le système étant du deuxième ordre on obtient :

$$\begin{cases} \frac{(K_p + f)}{J} = 2\xi\omega_n \\ \frac{K_i}{J} = \omega_n^2 \end{cases} \quad (5.4)$$

## Chapitre 5 Conception d'un contrôleur PI-PSO pour contrôler la vitesse d'un MAS

---

Ce qui donne les paramètres du régulateur PI en fonction du coefficient de frottement  $f$  et de moment d'inertie  $J$  du moteur ainsi que du coefficient d'amortissement  $\xi$  et de la pulsation naturelle  $\omega_n$  choisies pour une bonne correction du système :

$$\begin{cases} K_p = 2J\xi\omega_n - f \\ K_i = J\omega_n^2 \end{cases} \quad (5.5)$$

On choisit  $\xi$  et  $\omega_n$  et on obtient  $K_p$  Et  $K_i$  .

Dans notre travail on va faire une comparaison des performances de deux moteurs en utilisant un PI par placement des pôles et un PI-PSO.

Le premier moteur est un moteur à courant continu de coefficient de frottement  $f = 0.3$  et de moment d'inertie = 0.01 . En utilisant la méthode de placement des pôles avec  $\xi = 0.7$  et  $\omega_n = 70$ , On obtient les paramètres du régulateur PI en utilisant l'instruction Matlab suivante :

$$\left| \begin{array}{l} \gg f = 0.3 ; J = 0.01 ; z = 0.7 ; w = 70 ; K_p = 2 * J * z * w - f , K_i = J * w ^ 2 \end{array} \right.$$

Les paramètres du régulateur PI par placement des pôles pour la régulation de vitesse du moteur à courant continu sont donc donnés par :

$$\left| \begin{array}{l} K_p = 0.68 \\ K_i = 49 \end{array} \right.$$

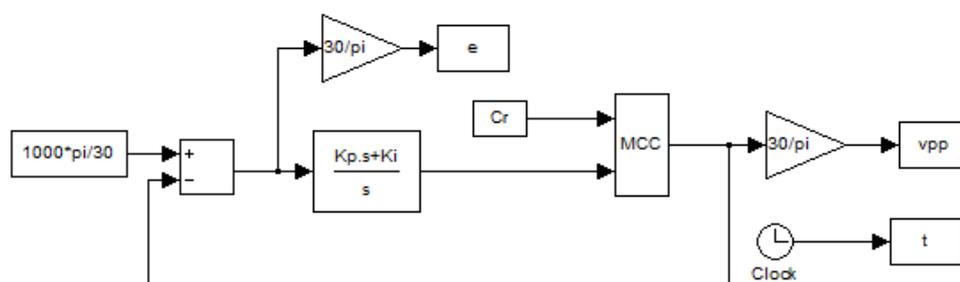
Le deuxième moteur est un moteur asynchrone coefficient de frottement  $f = 0.0114$  et de moment d'inertie = 0.031 . En utilisant la même instruction Matlab avec  $z = 0.7$  et  $w = 10$ , on obtient les paramètres du régulateur PI par placement des pôles pour la régulation de vitesse du moteur asynchrone sont donnés par :

$$\left| \begin{array}{l} K_p = 0.4329 \\ K_i = 3.1 \end{array} \right.$$

### 5.3. Conception off - line d'un Contrôleur PI-PSO pour un moteur à courant continu

L'algorithme PSO est un outil mathématique d'optimisation. Etant donnée une fonction à plusieurs variables qu'on appelle fonction objective, l'algorithme PSO permet de trouver la valeur des variables qui minimisent ou maximisent la fonction objective.

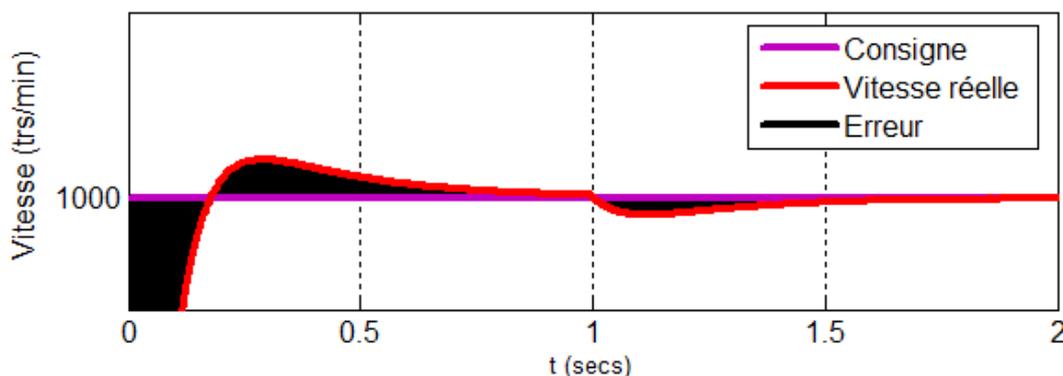
Dans notre cas on va l'utiliser pour trouver les valeurs des paramètres  $K_p$  et  $K_i$  du régulateur PI qui permettent de minimiser l'erreur entre la vitesse de référence et la vitesse réelle d'un moteur à courant continu.



**Figure 5.3.** Modèle Simulink MCC\_PI.m pour la régulation de vitesse d'un MCC.

Avant de concevoir le régulateur PI-PSO, on définit d'abord la fonction objective que l'algorithme PSO doit minimiser.

La fonction objective choisie est la surface comprise entre la courbe de la vitesse et la ligne de référence comme indiquée en noir dans la figure 5.4.



**Figure 5.4.** Surface erreur dans la régulation de vitesse d'un MCC.

## Chapitre 5 Conception d'un contrôleur PI-PSO pour contrôler la vitesse d'un MAS

---

On peut approcher cette surface à la valeur suivante :

$$A = \frac{t_{\text{fin}}}{n} \sum_{i=1}^n |e_i| \quad (5.6)$$

Où  $t_{\text{fin}}$  est le temps de fin de la simulation du modèle de la machine régulé et  $e_i$  est l'erreur obtenue durant le pas numéro  $i$ . Puisque  $t_{\text{fin}}$  est constant, il suffit de minimiser la fonction objective suivante :

$$S = \frac{1}{n} \sum_{i=1}^n |e_i| \quad (5.7)$$

On veut utiliser l'algorithme PSO pour trouver les valeurs de  $K_p$  et  $K_i$  qui permettent de minimiser la valeur de  $S$ .

Le programme PSO utilise son algorithme de recherche du minimum de la fonction objective  $S$  qu'il calcul en exécutant le modèle Simulink **MCC\_PI.mdl** de la machine régulé pour différentes valeurs de  $K_p$  et  $K_i$  choisies dans des intervalles bien définies. Le résultat de cette recherche nous donne les valeurs des paramètres du régulateur PI-PSO off - line qui permet d'obtenir le moins d'erreur possible et ainsi concevoir le régulateur PI-PSO donnant la meilleur performance du système.

Si on veut que le dépassement ne dépasse pas 5% de la consigne de 1000 trs/mon, il faut ajouter la contrainte suivante :

Si la valeur minimale de l'erreur est telle que  $|\min(\mathbf{e})| > 50$  alors on affecte une grande valeur à la fonction objective (disons 500) pour ignorer tous les résultats pouvant donner un dépassement supérieure à 5%.

Voici le programme de la fonction objective :

```
function y = Objfun_mcc(x)
global Kp Ki
Kp = x(1) ; Ki = x(2);
sim('MCC_PI');
y1 = sum(abs(e)) / length(e);
if abs(min(e)) > 50
    y1 = 500;
end;
y = y1 ;
```

## Chapitre 5 Conception d'un contrôleur PI-PSO pour contrôler la vitesse d'un MAS

---

On a écrit un programme nommé **run\_pso\_mcc.m**, qui exécute la fonction **Objfun\_mcc.m** avec différentes valeurs de  $K_p$  et  $K_i$  en se déplaçant dans des intervalles choisis de  $K_p$  et  $K_i$  suivant l'algorithme de déplacement d'un essaim de particule (Algorithme PSO), en utilisant plusieurs itérations et plusieurs essais et en sauvegardant à chaque itération les valeurs de  $K_p$  et  $K_i$  qui donnent la plus petite valeur de la fonction objective (Surface S). A la fin du programme il affiche les valeurs optimaux de  $K_p$  et  $K_i$  du régulateur PI-PSO recherché.

Après exécution du programme **run\_pso\_mcc.m** avec l'intervalle de recherche  $[0, 100]$  pour  $K_p$  et  $[0, 1000]$  pour  $K_i$ , un essaim de 20 particules, un nombre d'itérations de 20 et un nombre d'essais égal à 2, on a obtenu le résultat suivant :

Essai : 1			
Itération	$K_p$	$K_i$	Fonction Objective
1	2.0000	569.0000	0.7871
2	0.7245	601.9284	0.7404
3	0.7245	601.9284	0.7404
4	0.7245	601.9284	0.7404
5	0.7731	773.2164	0.6005
6	0.7731	773.2164	0.6005
7	0.7731	773.2164	0.6005
8	0.7731	773.2164	0.6005
9	1.2910	750.2582	0.5980
10	1.0691	764.0740	0.5876
11	0.8319	848.3895	0.5836
12	1.3163	812.4853	0.5623
13	1.3163	812.4853	0.5623
14	1.3163	812.4853	0.5623
15	1.1628	857.5572	0.5560
16	1.1628	857.5572	0.5560
17	1.1628	857.5572	0.5560
18	1.2521	841.4180	0.5549
19	1.3630	846.6762	0.5516
20	1.3630	846.6762	0.5516

---

Essai : 2			
Itération	$K_p$	$K_i$	Fonction Objective
1	1.7476	535.0291	0.8307
2	1.7476	535.0291	0.8307
3	2.0137	607.1336	0.7428
4	1.0544	654.0552	0.6826
5	1.1972	782.7123	0.5755
6	1.1972	782.7123	0.5755
7	1.1972	782.7123	0.5755

## Chapitre 5 Conception d'un contrôleur PI-PSO pour contrôler la vitesse d'un MAS

8	1.1972	782.7123	0.5755
9	1.1972	782.7123	0.5755
10	1.3646	830.4679	0.5559
11	1.2729	849.8267	0.5528
12	1.2729	849.8267	0.5528
13	1.2729	849.8267	0.5528
14	1.3769	844.0317	0.5521
15	1.3769	844.0317	0.5521
16	1.3769	844.0317	0.5521
17	1.3769	844.0317	0.5521
18	1.3769	844.0317	0.5521
19	1.3769	844.0317	0.5521
20	1.2914	860.9354	0.5503

Résultat Final

Meilleure fonction objective = 0.5503

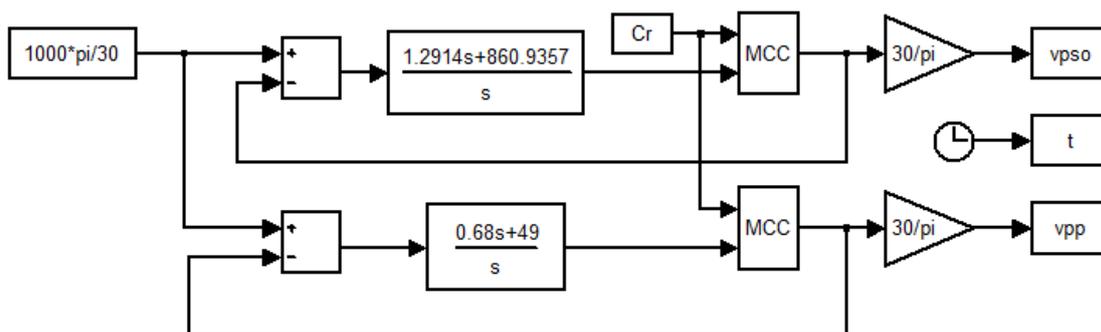
Meilleur  $K_p$  = 1.2914

Meilleur  $K_i$  = 860.9354

Donc les meilleures valeurs de  $K_p$  et  $K_i$  obtenues après exécution du programme sont :

$$K_p = 1.2914 \quad \text{et} \quad K_i = 860.9357$$

On remplace ces valeurs dans notre modèle Simulink **MCC\_pso\_pp.mdl** et on l'exécute pour voir le résultat du PI-PSO puis on le compare avec le résultat d'un PI par placement des pôles.



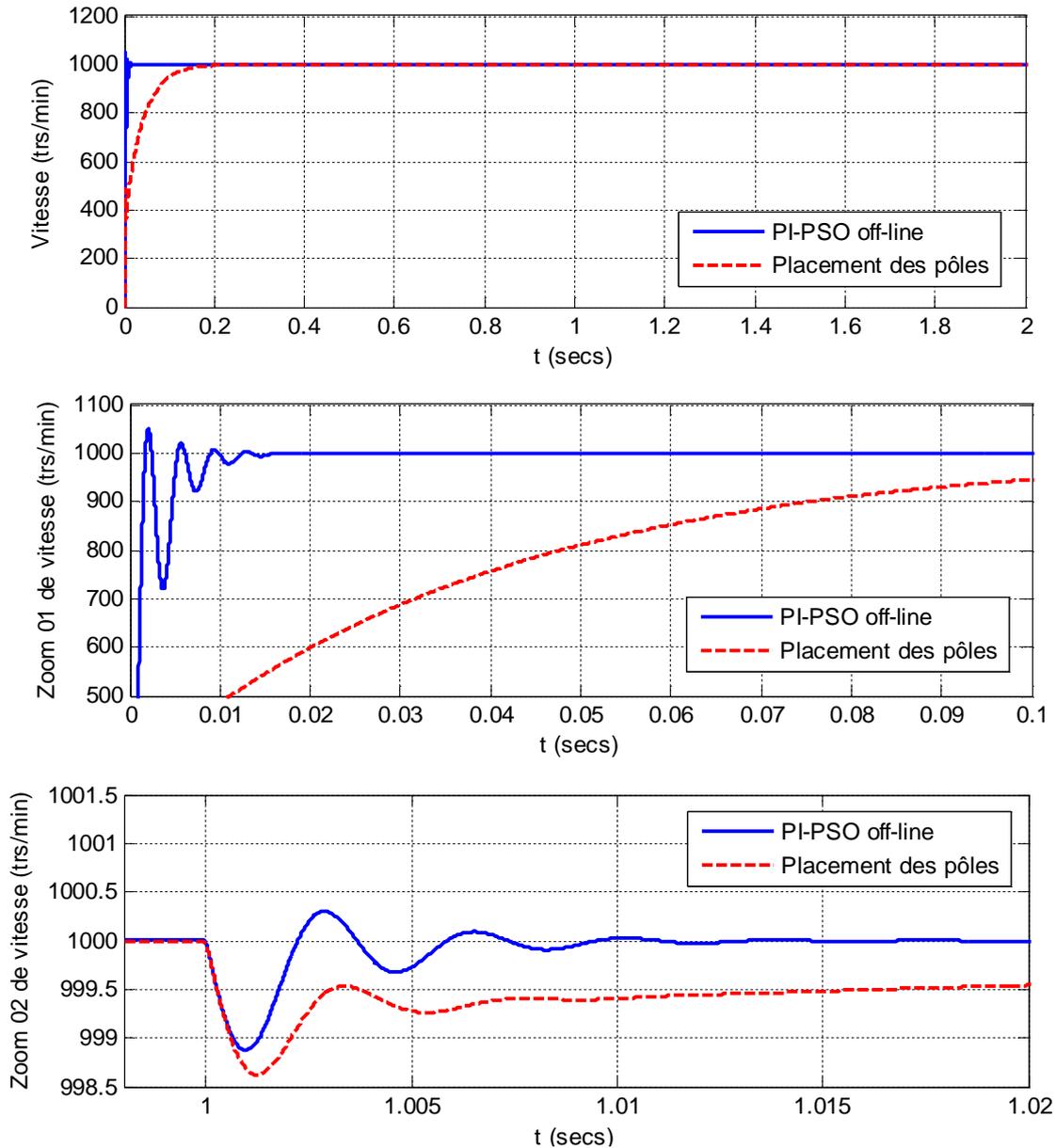
**Figure 5.5.** Modèle Simulink de la régulation simultanée de vitesse d'un MCC avec un PI trouvé par PSO offline et un PI trouvé par placement des pôles.

Après exécution du modèle de la figure 5.5 (**MCC\_pso\_pp.mdl**) avec une charge de 10 N.m au démarrage et une perturbation de +2 N.m de charge à  $t = 1$  sec on obtient le résultat de la figure 5.6.

## Chapitre 5 Conception d'un contrôleur PI-PSO pour contrôler la vitesse d'un MAS

Cette figure montre la supériorité des performances du régulateur PI-PSO off - line sur le régulateur PI obtenu par placement des pôles.

Le temps de réponse 5% du PI-PSO est inférieur à 0.01 secs alors que celui du PI par placement de pôles est supérieur à 0.1 secs



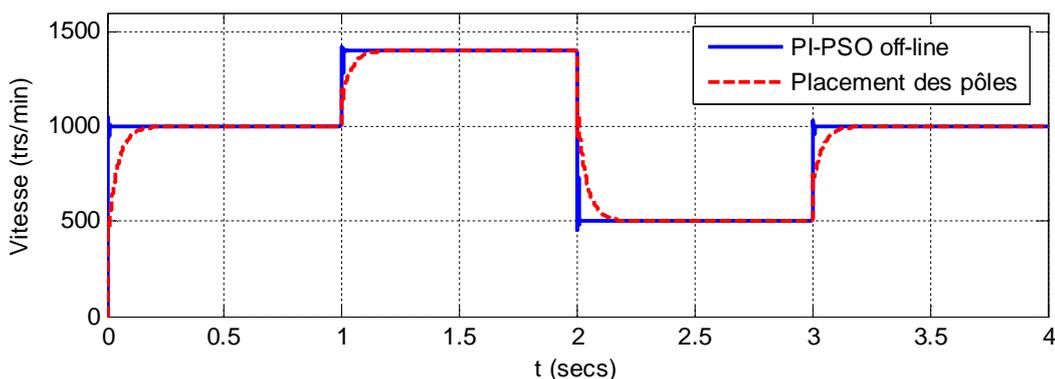
**Figure 5.6.** Performances d'un PI-PSO off line et d'un PI par placement des pôles lors d'un démarrage en charge avec perturbation de charge d'un MCC.

## Chapitre 5 Conception d'un contrôleur PI-PSO pour contrôler la vitesse d'un MAS

La perturbation de charge est rejetée par le PI-PSO en moins de 0.02 secs avec une chute de vitesse de 1.1 trs/min alors que le PI par placement des pôles rejette la perturbation de charge en 0.2 secs avec une chute de vitesse de 1.4 trs/min.

La figure 5.7 montre les performances du régulateur PI-PSO off – line ainsi que ceux du régulateur PI par placement des pôles lors d'une poursuite de vitesse d'un moteur à courant continu. Le moteur est démarré avec une consigne de 1000 trs/min avant d'être instantanément augmentée à 1400 trs/min à l'instant  $t = 1$  sec puis instantanément diminuée à 500 trs/min à l'instant  $t = 2$  secs avant de revenir à 1000 trs/min à l'instant  $t = 3$  secs (Modèle `MCC_pso_pi_pour.mdl`).

Là aussi le régulateur PI-PSO off - line donne de meilleures performances de poursuite que le régulateur PI obtenu par placement des pôles.

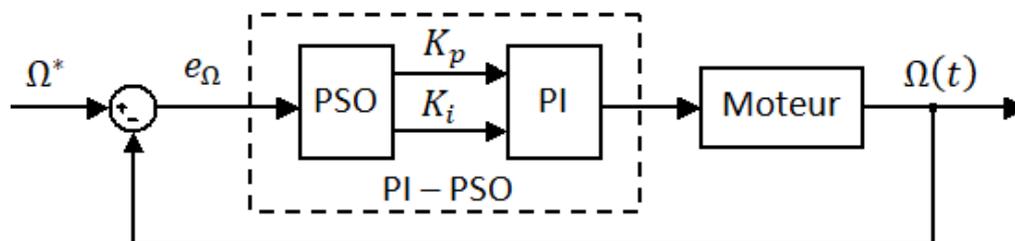


**Figure 5.7.** Performances d'un PI-PSO off line et d'un PI par placement des pôles lors d'une poursuite de la vitesse d'un MCC.

### 5.4. Conception on - line d'un Contrôleur PI-PSO pour un moteur à courant continu

Une recherche des paramètres du contrôleur PI par l'algorithme PSO peut être réalisée durant le fonctionnement du moteur (on-line), et ceci en incluant un bloc de recherche des paramètres  $K_p$  et  $K_i$  par l'algorithme PSO en amont du contrôleur PI comme indiquée dans la figure 5.8 [17].

## Chapitre 5 Conception d'un contrôleur PI-PSO pour contrôler la vitesse d'un MAS



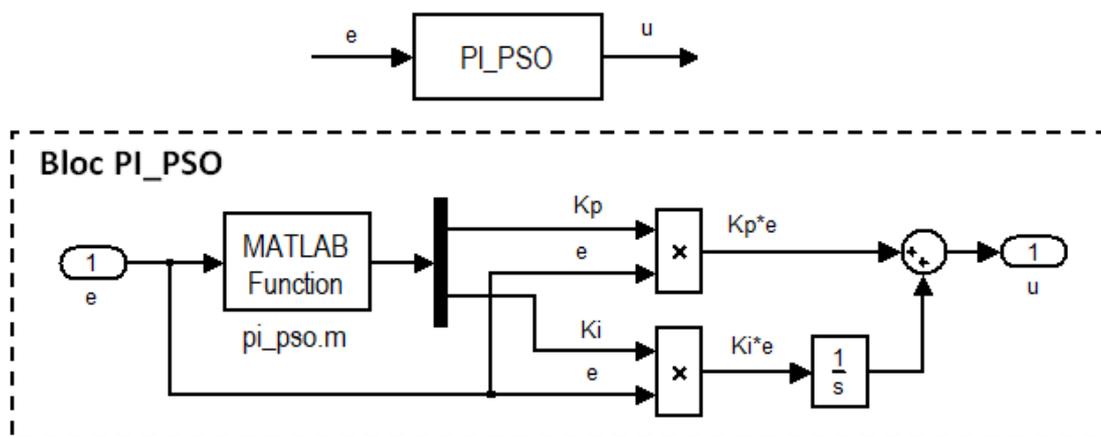
**Figure 5.8.** Contrôleur PI-PSO on-line pour la commande d'un moteur.

Le bloc de recherche utilise l'erreur de vitesse  $e_\Omega$  pour calculer les paramètres  $K_p$  et  $K_i$  qui minimisent la fonction objective suivante [17] par l'algorithme PSO :

$$F_{obj}(K_p, K_i) = \alpha_1 \cdot e_\Omega(k)^2 + \alpha_2 \cdot (K_p \cdot e_\Omega(k) + K_i \cdot e_\Omega(k) \cdot T)^2 \quad (5.8)$$

Où  $\alpha_1$  et  $\alpha_2$  représentent le poids d'importance du premier et du second terme de l'équation (5.6) respectivement,  $T$  est le temps d'échantillonnage ou le pas de simulation dans notre cas et  $K_p$ ,  $K_i$  sont les paramètres du contrôleur PI.

Un modèle Simulink du contrôleur PI-PSO on-line a été conçu en utilisant le logiciel Simulink et il est donné par la figure 5.9.



**Figure 5.9.** Modèle Simulink du contrôleur PI-PSO on-line.

Le bloc **Matlab function** exécute une fonction de la variable erreur de vitesse, nommé **pi\_pso\_mcc.m** et donné en annexe, pour obtenir les valeurs de  $K_p$  et  $K_i$  permettant de minimiser avec l'algorithme PSO la fonction objective de l'équation 5.8 où  $\alpha_1 = 0.9$  Et

## Chapitre 5 Conception d'un contrôleur PI-PSO pour contrôler la vitesse d'un MAS

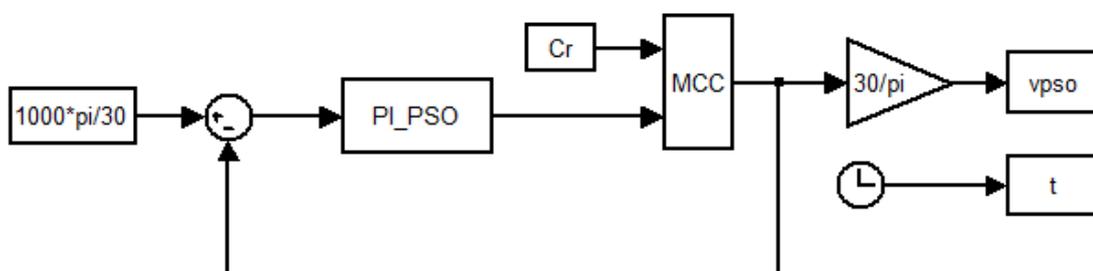
$\alpha_2 = 0.1$ . Les sorties  $K_p$  et  $K_i$  ainsi que l'erreur de vitesse sont utilisés pour construire la réponse du contrôleur PI donnée par l'équation 5.2.

On a remplacé le contrôleur PI par le contrôleur PI-PSO dans la régulation de vitesse d'un moteur à courant continu où la fonction **pi\_pso\_mcc.m** génère les paramètres  $K_p$  et  $K_i$  qui minimisent la fonction objective **Objfun\_PIPSO.m** en une seule itération en utilisant un essaim de 10 particules dans un intervalle de recherche  $[0.5, 2]$  pour  $K_p$  et  $[2, 1000]$  pour  $K_i$ .

La fonction objective **Objfun\_PIPSO.m** est donnée par :

```
function y = Objfun_PIPSO(x)
T = 0.00001;
e = x(1) ; Kp = x(2) ; Ki = x(3) ;
y = 0.9 * e ^ 2 + 0.1 * ( Kp * e + Ki * e * T ) ^ 2 ;
```

La figure 5.10 montre le modèle Simulink **MCC\_PIPSO.mdl** de la régulation de vitesse d'un moteur à courant continu en utilisant un régulateur PI-PSO on-line.



**Figure 5.10.** Modèle Simulink de la régulation de vitesse d'un MCC par un PI-PSO.

Après exécution du modèle **MCC\_PIPSO.mdl** avec une charge de  $10 \text{ N.m}$  au démarrage et une perturbation de  $+2 \text{ N.m}$  de charge à  $t = 1 \text{ sec}$  on obtient le résultat de la figure 5.11.

Là aussi, on voit la supériorité des performances du régulateur PI-PSO on - line sur le régulateur PI obtenu par placement des pôles.

Le temps de réponse 5% du PI-PSO est aux environs de 0.01 secs alors que celui du PI par placement de pôles est supérieur à 0.1 secs

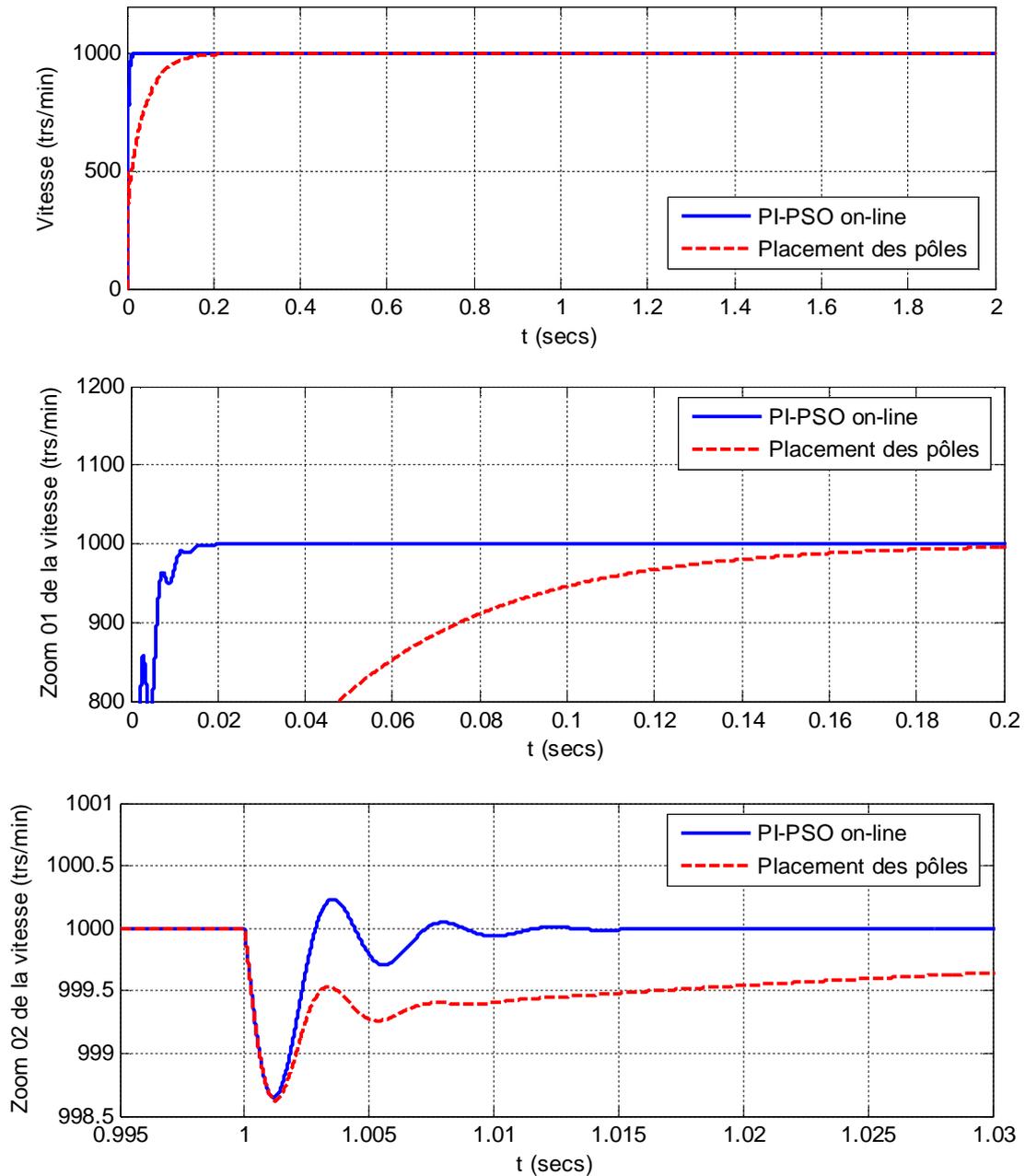
## Chapitre 5 Conception d'un contrôleur PI-PSO pour contrôler la vitesse d'un MAS

---

La perturbation de charge est rejetée par le PI-PSO on-line en moins de 0.015 secs avec une chute de vitesse de 1.4 trs/min alors que le PI par placement des pôles rejette la perturbation de charge en 0.2 secs avec une même chute de vitesse de 1.4 trs/min.

La figure 5.12 montre les performances du régulateur PI-PSO on – line lors d'une poursuite de vitesse d'un moteur à courant continu. Le moteur est démarré avec une consigne de 1000 trs/min avant d'être instantanément augmentée à 1400 trs/min à l'instant  $t = 1$  sec puis instantanément diminuée à 500 trs/min à l'instant  $t = 2$  secs avant de revenir à 1000 trs/min à l'instant  $t = 3$  secs.

## Chapitre 5 Conception d'un contrôleur PI-PSO pour contrôler la vitesse d'un MAS



**Figure 5.11.** Performances d'un PI-PSO on-line et d'un PI par placement des pôles Lors d'un démarrage en charge avec perturbation de charge d'un MCC.

Là aussi le régulateur PI-PSO on - line donne de meilleures performances de poursuite que le régulateur PI obtenu par placement des pôles.

## Chapitre 5 Conception d'un contrôleur PI-PSO pour contrôler la vitesse d'un MAS

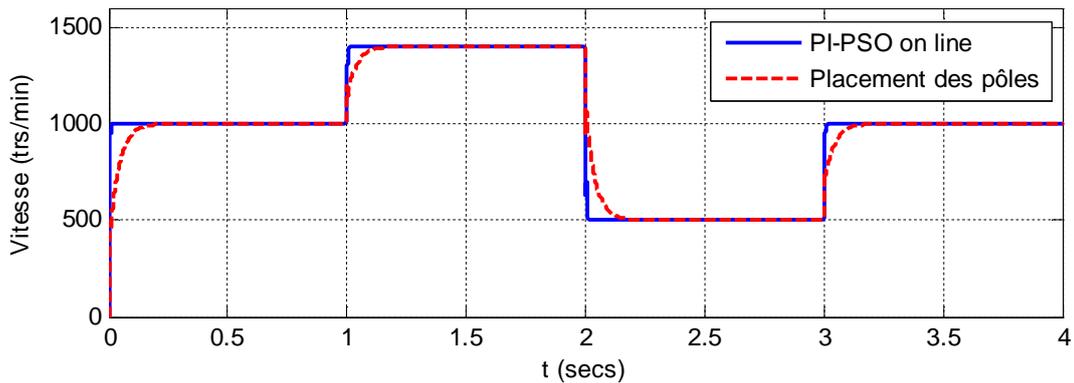


Figure 5.12. Performances d'un PI-PSO on line lors d'une poursuite de la vitesse d'un MCC.

### 5.5. Conception off - line d'un Contrôleur PI-PSO pour un moteur asynchrone

Dans cette partie on va concevoir un régulateur PI-PSO off line pour la régulation de la vitesse d'un moteur asynchrone du modèle.

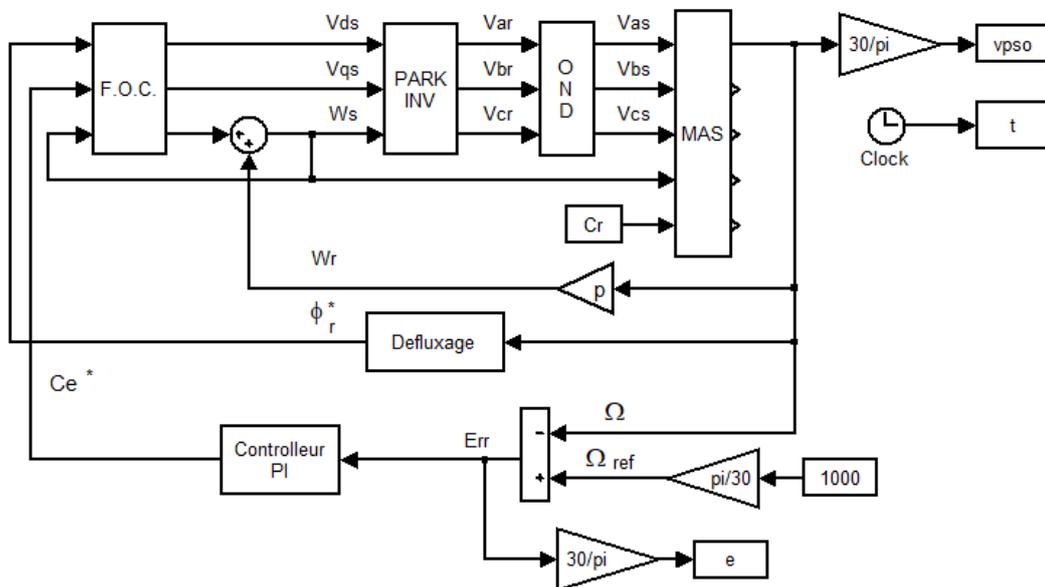


Figure 5.13. Modèle Simulink MAS\_MLIV\_PI.m pour la régulation de vitesse d'un moteur asynchrone.

## Chapitre 5 Conception d'un contrôleur PI-PSO pour contrôler la vitesse d'un MAS

---

Pour cela on va utiliser la même procédure que dans le cas du moteur à courant continu sans prendre en compte la contrainte du dépassement. On a écrit le programme **run\_pso\_mas.m** qui exécute la fonction **Objfun\_mas.m** pour rechercher les valeurs optimales des paramètres  $K_p$  et  $K_i$  du régulateur PI-PSO recherché par l'algorithme PSO.

Voici le programme de la fonction objective :

```
function y = Objfun_mas(x)
global Kp Ki
Kp = x(1) ; Ki = x(2);
sim('MAS_CVECT_PI');
y = sum(abs(e)) / length(e);
```

La fonction **Objfun\_mas .m** exécute le modèle **MAS\_CVECT\_PI.mdl** pour calculer la surface  $S$  qu'on veut minimiser est qui est donnée par l'équation 5.7.

Après exécution du programme **run\_pso\_mas.m** avec l'intervalle de recherche  $[0, 2]$  pour  $K_p$  et  $[0.5, 5]$  pour  $K_i$ , un essaim de 10 particules, un nombre d'itérations de 20 et un nombre d'essais égal à 2, on a obtenu le résultat suivant :

Essai : 1			
Itération	$K_p$	$K_i$	Fonction Objective
1	0.5304	1.0834	35.0427
2	0.5304	1.0834	35.0427
3	0.5630	1.1081	34.2789
4	0.5630	1.1081	34.2789
5	0.5630	1.1081	34.2789
6	0.5014	2.8738	33.0747
7	0.5014	2.8738	33.0747
8	0.5004	2.1033	32.1627
9	0.4936	1.3390	31.3365
10	0.4936	1.3390	31.3365
11	0.5204	1.3115	30.8143
12	0.5204	1.3115	30.8143
13	0.5536	1.6189	29.7137
14	0.5536	1.6189	29.7137
15	0.5536	1.6189	29.7137
16	0.5445	1.3363	29.3051
17	0.5767	1.4295	29.1814
18	0.5767	1.4295	29.1814

## Chapitre 5 Conception d'un contrôleur PI-PSO pour contrôler la vitesse d'un MAS

---

19	0.5665	1.3711	29.1273
20	0.5879	1.3970	28.7982
Essai : 2			
Itération	Kp	Ki	Fonction Objective
1	0.9535	1.0875	89.6479
2	0.7263	0.5000	65.8467
3	0.7105	0.7096	50.2789
4	0.7105	0.7096	50.2789
5	0.4907	0.7743	48.3777
6	0.4911	0.9801	38.7516
7	0.4914	1.1292	33.9446
8	0.6894	1.3306	31.5946
9	0.5522	1.4192	29.7952
10	0.5522	1.4192	29.7952
11	0.6100	1.4218	29.3215
12	0.5423	1.3981	28.7572
13	0.5423	1.3981	28.7572
14	0.5423	1.3981	28.7572
15	0.5423	1.3981	28.7572
16	0.5423	1.3981	28.7572
17	0.5423	1.3981	28.7572
18	0.5423	1.3981	28.7572
19	0.5423	1.3981	28.7572
20	0.5423	1.3981	28.7572
Résultat Final			
Meilleure fonction objective = 28.7572			
Meilleur Kp = 0.5423			
Meilleur Ki = 1.3981			

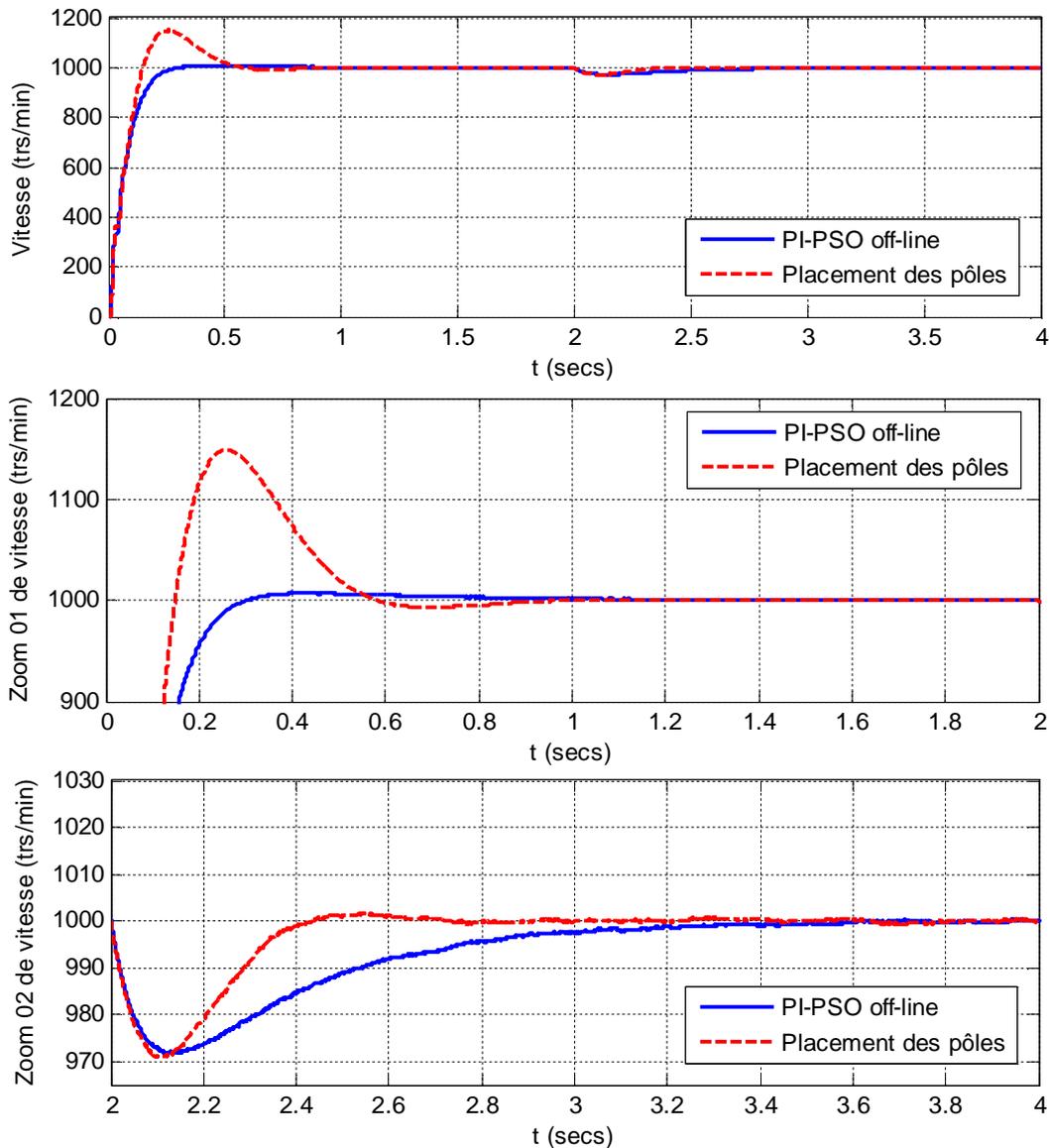
Donc les meilleurs valeurs de  $K_p$  et  $K_i$  obtenues après exécution du programme sont :

$$K_p = 0.5423 \quad \text{Et} \quad K_i = 1.3981$$

On remplace ces valeurs dans notre modèle Simulink **MAS\_VECT\_PI.mdl** et on l'exécute pour voir le résultat du PI-PSO puis on le compare avec le résultat d'un PI par placement des pôles. On rappelle que les paramètres du régulateur par placement des pôles pour la moteur asynchrone sont  $K_p = 0.4329$  et  $K_i = 3.1$

## Chapitre 5 Conception d'un contrôleur PI-PSO pour contrôler la vitesse d'un MAS

Le résultat de simulation est donné par la figure 5.14 où le moteur asynchrone est démarré avec une charge de  $10\text{ N.m}$  avec l'application d'une perturbation de  $+2\text{ N.m}$  de charge à  $t = 2\text{ sec}$ .



**Figure 5.14.** Performances d'un PI-PSO off line et d'un PI par placement des pôles Lors d'un démarrage en charge avec perturbation de charge d'un MAS.

Cette figure montre une meilleure performance du régulateur PI-PSO pendant le démarrage avec un rejet plus lent des perturbations.

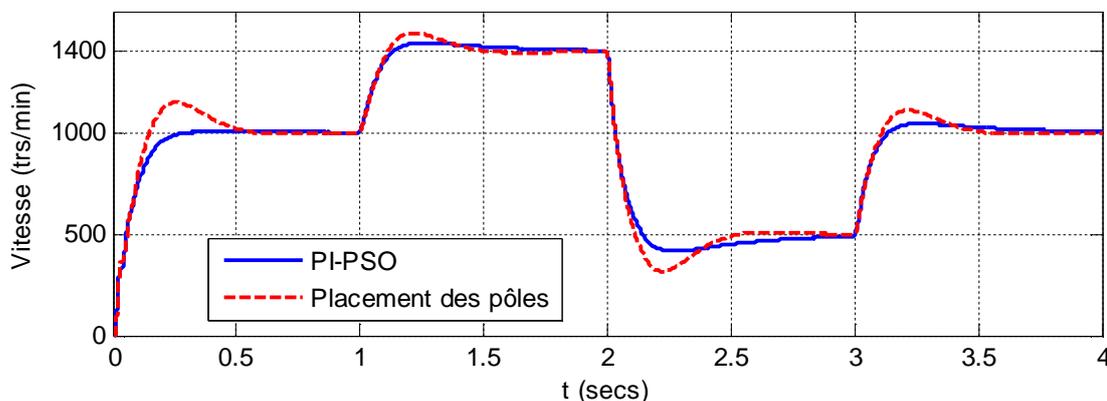
Le PI-PSO fait un dépassement de 0.7% avec un temps de réponse 5% de 0.2 secs, alors que le régulateur par placement des pôles est de 14.9% avec un temps de réponse 5% supérieur à 0.4 secs.

## Chapitre 5 Conception d'un contrôleur PI-PSO pour contrôler la vitesse d'un MAS

La perturbation de charge est rejetée totalement par le PI-PSO en 1.4 secs avec une chute de vitesse de 28 trs/min alors que le PI par placement des pôles rejette totalement la perturbation de charge en 0.7 secs avec une chute de vitesse de 29 trs/min.

La figure 5.15 montre les performances du régulateur PI-PSO off – line ainsi que ceux du régulateur PI par placement des pôles lors d'une poursuite de vitesse d'un moteur asynchrone. Le moteur est démarré avec une consigne de 1000 trs/min avant d'être instantanément augmentée à 1400 trs/min à l'instant  $t = 1$  sec puis instantanément diminuée à 500 trs/min à l'instant  $t = 2$  secs avant de revenir à 1000 trs/min à l'instant  $t = 3$  secs.

La simulation montre que le régulateur par placement des pôles fait un plus grand dépassement que le régulateur PI-PSO, mais atteint plus rapidement la consigne.



**Figure 5.15.** Performances d'un PI-PSO off line et d'un PI par placement des pôles lors d'une poursuite de la vitesse d'un asynchrone.

Si on accepte un dépassement de moins de 10%, on pourrait améliorer les performances du régulateur PI-PSO dans le rejet des perturbations et la poursuite de vitesse en modifiant la fonction objective et en ajoutant des contraintes.

La fonction objective qu'on doit minimiser maintenant est la surface entre la consigne et la vitesse réelle pendant le rejet de la perturbation avec les deux contraintes suivantes :

- Valeur maximale du dépassement inférieur à 10%
- La vitesse atteint la consigne rapidement, en imposant que l'erreur de vitesse soit inférieure à 2 trs/min après un temps  $t = 0.9$  secs.

## Chapitre 5 Conception d'un contrôleur PI-PSO pour contrôler la vitesse d'un MAS

---

Si la consigne est de 1000 trs/min, le pas de simulation est de 0.0001 secs et la perturbation de charge est appliquée de  $t = 1$  secs à  $t = 2$  secs , voici le programme de la fonction objective modifiée **Objfun\_mas\_mod.m** :

```
function y=Objfun_mas_mod(x)
global Kp Ki
Kp=x(1) ; Ki=x(2);
sim('MAS_CVECT_PI');
y1 = sum(abs(e(10001 : 20000))) / length(e(10001 : 20000));
if abs(min(e)) > 100
    y1=500;
end;
if abs(e(9001))>2
    y1=500;
end;
y=y1;
```

Après réexécution du programme **run\_pso\_mas\_mod.m** utilisant la fonction objective modifiée on a obtenu le résultat suivant :

Essai : 1

Itération	Kp	Ki	Fonction Objective
1	1.0000	4.0000	500.0000
2	1.0000	4.0000	500.0000
3	1.0000	4.0000	500.0000
4	1.0000	4.0000	500.0000
5	1.0000	4.0000	500.0000
6	0.4659	1.1485	15.4310
7	0.5791	1.2931	13.5926
8	0.5791	1.2931	13.5926
9	0.5791	1.2931	13.5926
10	0.5791	1.2931	13.5926
11	0.5791	1.2931	13.5926
12	0.5791	1.2931	13.5926
13	0.5791	1.2931	13.5926
14	0.5791	1.2931	13.5926

## Chapitre 5 Conception d'un contrôleur PI-PSO pour contrôler la vitesse d'un MAS

---

15	0.4157	1.9412	10.1220
16	0.4157	1.9412	10.1220
17	0.4157	1.9412	10.1220
18	0.4387	2.0993	9.3810
19	0.4387	2.0993	9.3810
20	0.4387	2.0993	9.3810

-----

Essai : 2

Itération	Kp	Ki	Fonction Objective
1	0.0000	3.0000	500.0000
2	0.0000	3.0000	500.0000
3	0.0000	3.0000	500.0000
4	0.5015	1.1763	15.3128
5	0.5015	1.1763	15.3128
6	0.5867	1.4644	11.6847
7	0.5867	1.4644	11.6847
8	0.5867	1.4644	11.6847
9	0.5867	1.4644	11.6847
10	0.5867	1.4644	11.6847
11	0.5867	1.4644	11.6847
12	0.5867	1.4644	11.6847
13	0.5867	1.4644	11.6847
14	0.5867	1.4644	11.6847
15	0.5034	2.4498	7.8701
16	0.5034	2.4498	7.8701
17	0.5034	2.4498	7.8701
18	0.5034	2.4498	7.8701
19	0.5355	2.6044	7.5355
20	0.5355	2.6044	7.5355

-----

Résultat Final

Meilleure fonction objective = 7.5355

## Chapitre 5 Conception d'un contrôleur PI-PSO pour contrôler la vitesse d'un MAS

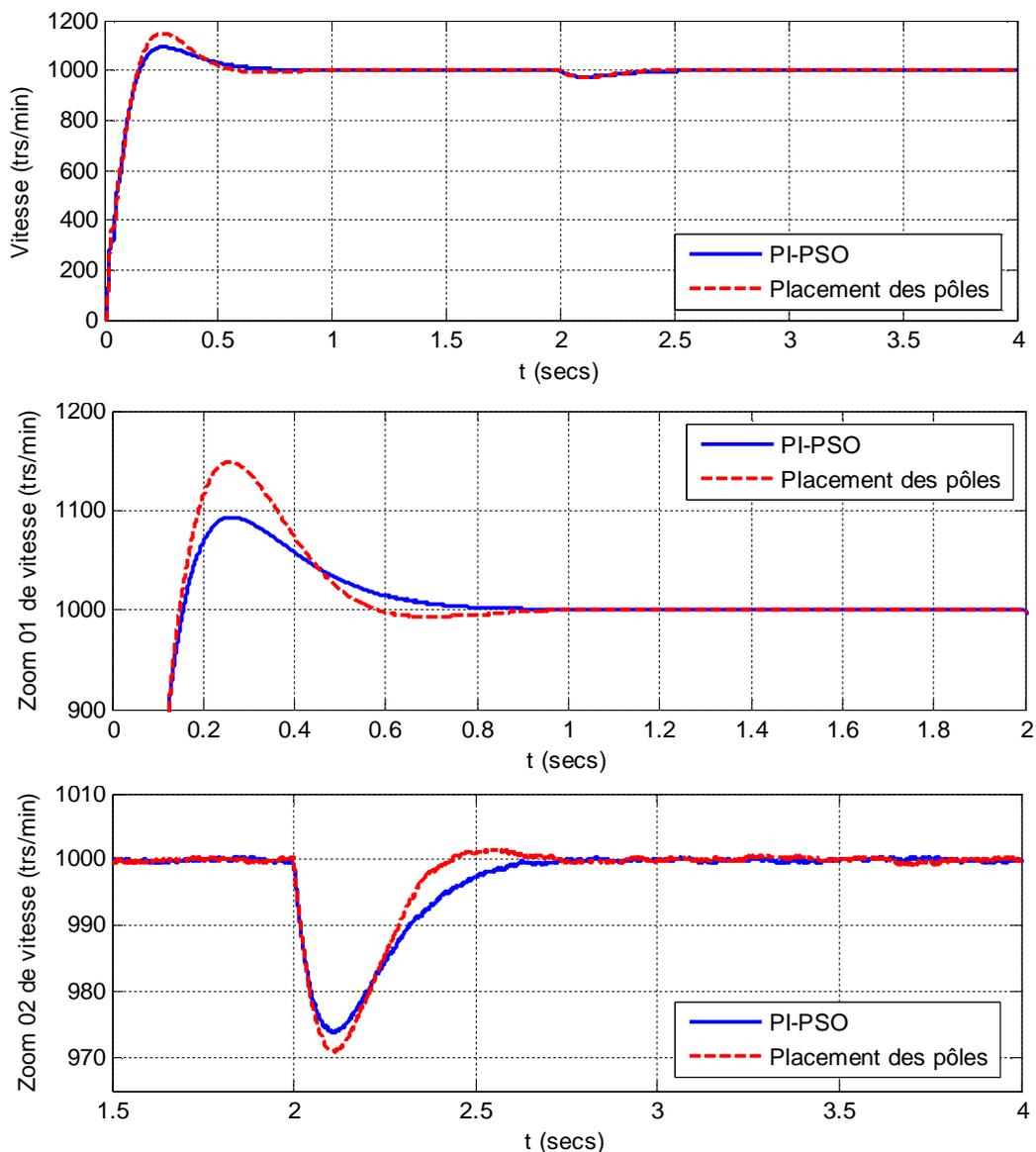
Meilleur\_  $K_p$  = 0.5355

Meilleur\_  $K_i$  = 2.6044

Donc les meilleures valeurs de  $K_p$  et  $K_i$  pour cette fonction objective sont :

$$K_p = 0.5355 \quad \text{et} \quad K_i = 2.6044$$

La figure 5.16 et 5.17 donnent les performances du PI-PSO amélioré pendant le démarrage en charge, après application de 20% de perturbation de charge et pendant la poursuite de la vitesse.



**Figure 5.16.** Performances d'un PI-PSO amélioré et d'un PI par placement des pôles lors d'un démarrage en charge avec perturbation de charge d'un MAS.

## Chapitre 5 Conception d'un contrôleur PI-PSO pour contrôler la vitesse d'un MAS

La modification de la fonction objective nous a permis de concevoir un régulateur PI-PSO dont les performances sont meilleures que le régulateur PI par placement des pôles, que ce soit au démarrage, lors de l'application des perturbations de charge ou lors de la poursuite de la vitesse.

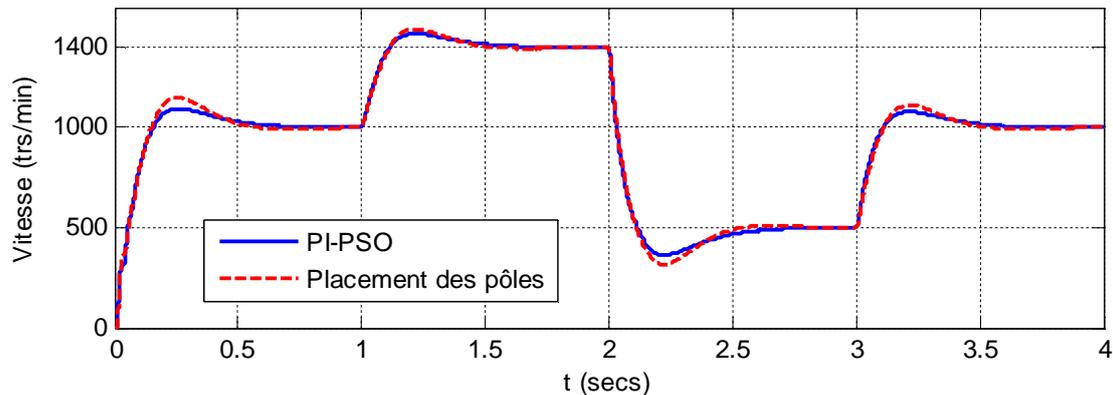


Figure 5.17. Performances d'un PI-PSO amélioré et d'un PI par placement des pôles lors d'une poursuite de la vitesse d'un asynchrone.

### 5.6. Conception on - line d'un Contrôleur PI-PSO pour un moteur asynchrone

De la même manière que dans le cas du moteur à courant continu, la recherche des paramètres du contrôleur PI par l'algorithme PSO peut être réalisée on-line durant le fonctionnement du moteur asynchrone, et ceci en incluant un bloc de recherche des paramètres  $K_p$  et  $K_i$  par l'algorithme PSO en amont du contrôleur PI comme indiquée dans la figure 5.8.

Le bloc de recherche utilise l'erreur de vitesse  $e_n$  pour calculer les paramètres  $K_p$  et  $K_i$  qui minimisent une fonction objective de l'équation 5.8.

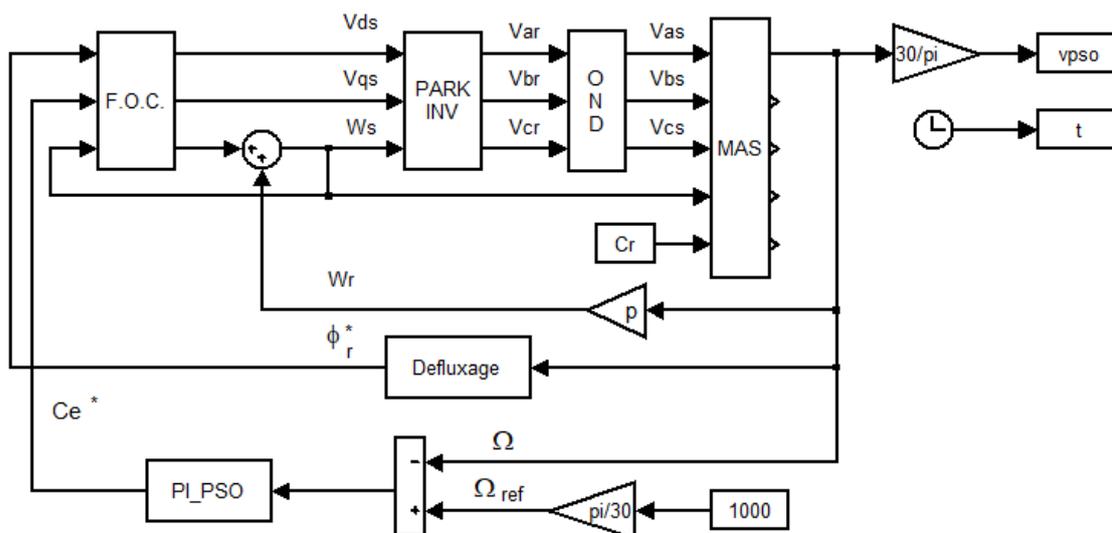
Le modèle Simulink du contrôleur PI-PSO on-line de la figure 5.9 a été utilisée pour simuler le régulateur PI-PSO on-line de la vitesse d'un moteur asynchrone. Le bloc **Matlab function** exécute la fonction **pi\_pso\_mas.m** donnée en annexe pour obtenir, par l'algorithme PSO, les valeurs de  $K_p$  et  $K_i$ .

On a remplacé le contrôleur PI par le contrôleur PI-PSO dans la régulation de vitesse d'un moteur à courant continu où la fonction **pi\_pso\_mas.m** génère les paramètres  $K_p$  et  $K_i$  qui

## Chapitre 5 Conception d'un contrôleur PI-PSO pour contrôler la vitesse d'un MAS

minimisent la fonction objective **Objfun\_PIPSO.m** en dix itérations en utilisant un essaim de 10 particules dans un intervalle de recherche  $[0.5, 2]$  pour  $K_p$  et  $[1, 5]$  pour  $K_i$ .

La figure 5.18 montre le modèle Simulink **CVECT\_MAS\_PIPSO.mdl** de la régulation de vitesse d'un moteur asynchrone en utilisant le régulateur PI-PSO on-line.



**Figure 5.18.** Modèle Simulink de la régulation de vitesse d'un MAS par un PI-PSO.

Après exécution du modèle **CVECT\_MAS\_PIPSO.mdl** avec une charge de  $10 \text{ N.m}$  au démarrage et une perturbation de  $+2 \text{ N.m}$  de charge à  $t = 1 \text{ sec}$  on obtient le résultat de la figure 5.19.

Comme dans le cas du PI-PSO off-line, le PI-PSO donne une meilleure performance pendant le démarrage mais rejette plus lentement les perturbations.

Le PI-PSO ne fait aucun dépassement avec un temps de réponse 5% de 0.23 secs, alors que le régulateur par placement des pôles est de 14.9% avec un temps de réponse 5% supérieur à 0.4 secs.

La perturbation de charge est rejetée totalement par le PI-PSO en 1.4 secs avec une chute de vitesse de 31 trs/min alors que le PI par placement des pôles rejette totalement la perturbation de charge en 0.7 secs avec une chute de vitesse de 29 trs/min.

La figure 5.20 montre les performances du régulateur PI-PSO off – line ainsi que ceux du régulateur PI par placement des pôles lors d'une poursuite de vitesse d'un moteur asynchrone.

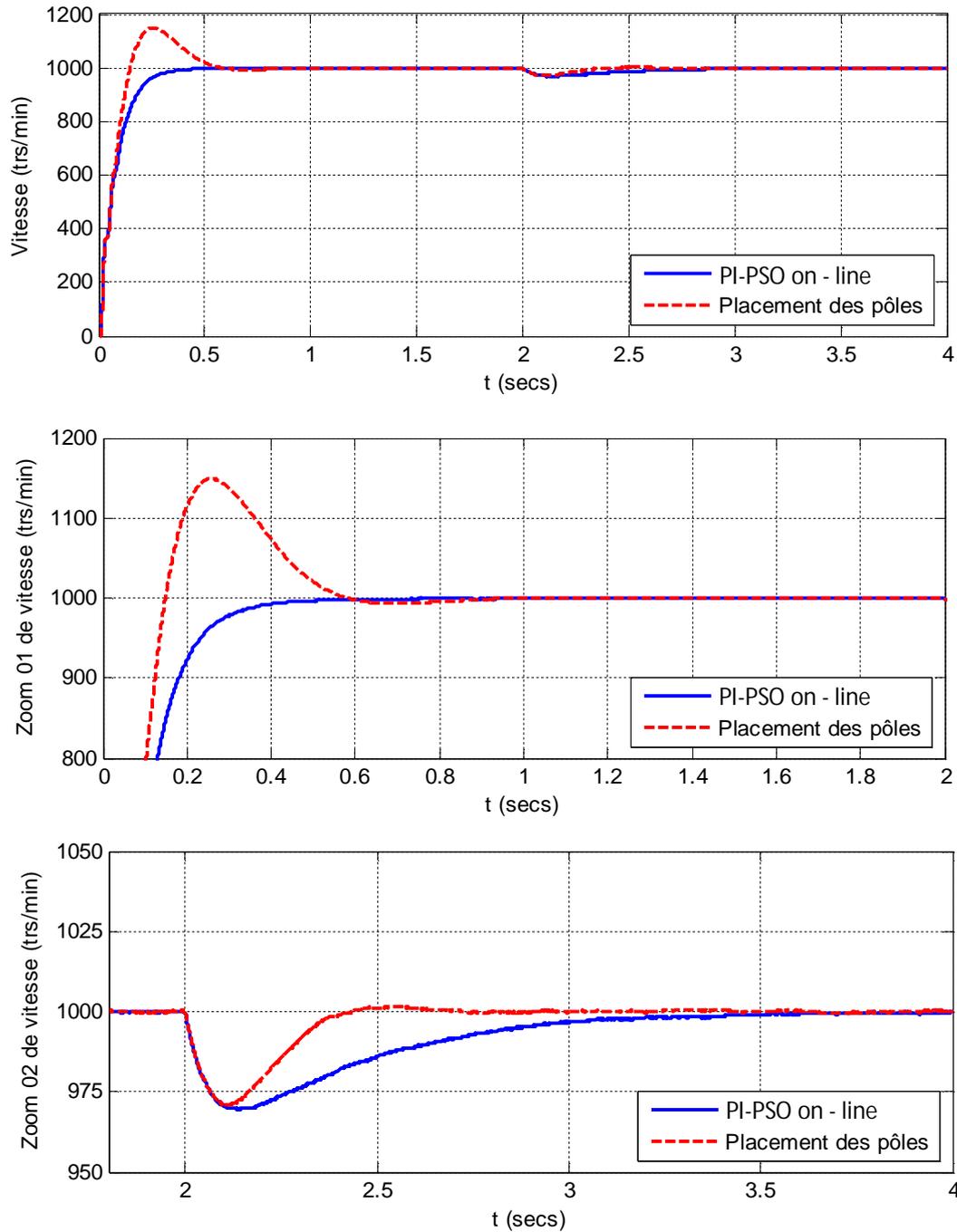
## Chapitre 5 Conception d'un contrôleur PI-PSO pour contrôler la vitesse d'un MAS

---

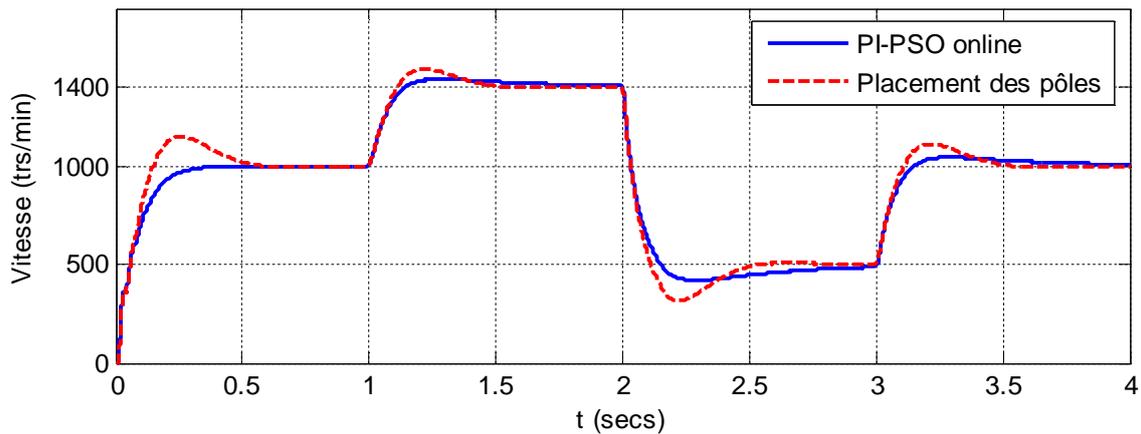
Le moteur est démarré avec une consigne de 1000 trs/min avant d'être instantanément augmentée à 1400 trs/min à l'instant  $t = 2$  sec puis instantanément diminuée à 500 trs/min à l'instant  $t = 2$  secs avant de revenir à 1000 trs/min à l'instant  $t = 3$  secs.

La simulation montre que le régulateur par placement des pôles fait un plus grand dépassement que le régulateur PI-PSO, mais atteint plus rapidement la consigne.

## Chapitre 5 Conception d'un contrôleur PI-PSO pour contrôler la vitesse d'un MAS



**Figure 5.19.** Performances d'un PI-PSO on-line et d'un PI par placement des pôles lors d'un démarrage en charge avec perturbation de charge d'un MAS.



**Figure 5.20.** Performances d'un PI-PSO on line lors d'une poursuite de la vitesse d'un MAS.

### 5.7. Conclusion

Dans ce chapitre on a fait la conception des régulateur PI-PSO off-line et on-line pour la régulation de vitesse dans la commande vectorielle d'un MAS. On a d'abord présenté le principe de la conception sur un MCC où on a conçu les régulateurs PI-PSO off-line et on-line pour la vitesse d'un MCC et comparé ses performance avec le régulateur PI obtenu par placement des pôles. On a ensuite appliqué ce principe à la conception des régulateur PI-PSO off-line et on-line au moteur asynchrone. Les résultats de simulation ont montrés une meilleure performance du PI-PSO comparé au régulateur PI par placement des pôles.

# Conclusion générale

## Conclusion Générale

---

Dans ce présent travail, on avait comme objectif la conception, la modélisation et la simulation de régulateurs PI-PSO pour la régulation de la vitesse dans la commande vectorielle d'une machine asynchrone. Le PI-PSO étant un régulateur PI dont les gains sont recherchés par l'algorithme d'optimisation par essaim de particules (Particle Swarm Optimisation) afin de minimiser une fonction objective en relation directe avec les performances de la machine.

On a d'abord développé les modèles mathématiques du moteur asynchrone utilisés par la commande vectorielle ainsi que ceux de l'onduleur à deux niveaux de tensions avec les commandes MLI sinus triangle. Puis on a développé les modèles mathématiques de la stratégie de commande vectorielle. Ceci nous a permis de faire la simulation de la régulation de la vitesse d'un moteur asynchrone à commande vectorielle en utilisant le régulateur PI classique.

La simulation de la vitesse du moteur asynchrone à commande vectorielle en utilisant le régulateur PI classique a montré la difficulté du régulateur à éliminer le dépassement sans dégrader sa réaction face aux perturbations de charge.

On a ensuite présenté la méthode d'optimisation par essaim particulaire. On a d'abord présenté son principe de fonctionnement avant de parler du mécanisme de confinement pour maintenir la population de particules à l'intérieur de l'espace de recherche, ainsi que de la notion de voisinage. On a ensuite fait la modélisation mathématique permettant la conception de l'algorithme avant de faire la simulation de l'algorithme appliquée à la recherche la valeur minimale d'une fonction à trois variables. La simulation a montré que l'algorithme PSO a réussi à trouver la solution minimisant une fonction à trois variables avec une erreur de 0.0031 en 15 itérations et 0.153 s.

On a ensuite procédé à l'application de l'algorithme PSO pour la recherche des paramètres optimaux d'un régulateur PI utilisé pour la régulation de la vitesse dans la commande vectorielle d'un moteur asynchrone.

On a fait en premier lieu la conception des régulateur PI - PSO off - line pour la régulation de vitesse dans la commande vectorielle d'un MAS. On a présenté le principe de la conception sur un MCC où la fonction objective est construite à partir de la surface de l'erreur.

On a ensuite proposé une méthode on - line pour la vitesse d'un MCC en incluant un bloc de recherche des paramètres  $K_p$  et  $K_i$  par l'algorithme PSO en amont du contrôleur PI et dont

## Conclusion Générale

---

la fonction objective est construite à partir de la surface de l'erreur et des valeurs précédentes des gains  $K_p$  et  $K_i$ .

Des travaux de simulations ont montrés que le régulateur PI – PSO est plus performant que le régulateur classique car il peut éliminer le dépassement sans dégrader sa réaction face aux perturbations de charge

Le principe de la conception des régulateur PI-PSO off - line et on - line d'un MCC a été appliqué au moteur asynchrone et les régulateurs conçus ont été utilisés pour simuler la commande vectorielle d'un moteur asynchrone.

Les résultats de simulation ont montrés une meilleure performance du PI-PSO comparé au régulateur PI par placement des pôles.

# Bibliographie

## Bibliographie

---

- [1]. Housseini B., « Prototypage rapide à base de FPGA d'un algorithme de contrôle avancé pour le moteur à induction », Mémoire de maîtrise en génie électrique, Université du Québec 2010.
- [2]. Smail B., « Contribution Au Diagnostic De La Machine Asynchrone Par Estimation Paramétrique », Thèse de Doctorat, Université de Poitiers 2002.
- [3]. P. Vas, « Electrical machines and drives : a space vector theory approach ». Oxford University Press 1992.
- [4]. J.P. Caron, J.P. Hautier, « Modélisation et Commande de MAS », Edition Technique, 1995.
- [5]. B.K. Bose, « Modern Power Electronics and AC drives », Prentice-Hall, Inc. 2002, pp. 573-574.
- [6]. G. Grellet, G. Clerc, « Actionneurs Electriques (Principes – Modèles – commandes) », Edition Eyrolles, 1997.
- [7]. Larid .M, Agaida A., « Etude comparative de la robustesse d'un contrôleur à gain variable (VGPI) et d'un contrôleur (PI) classique dans la commande d'un moteur Asynchrone », Mémoire d'ingénieur en électrotechnique, Université Dr Moulay Tahar de Saida, 2004.
- [8]. Khatri O., Dekhil M., « Réglage de vitesse d'une MAS par un régulateur PI à gain variable alimentée par un onduleur de tension à MLI vectorielle », Mémoire d'ingénieur en électrotechnique, Université Dr Moulay Tahar de Saida, 2005.
- [9]. Adjimi N., Belaidi W., « Modélisation et commande d'un onduleur MLI », Mémoire de Master en électrotechnique, Université Larbi Ben M'hidi, Oum-Bouaghi, 2019.
- [10]. Benhadjer A., 'Etude, simulation et réalisation d'un onduleur triphasé pour les trains électro-diesel', Mémoire de fin d'étude, Ecole supérieure en sciences appliquées, Tlemcen , 2019 .
- [11]. G. Segquier, F. Labrique, R. Baussier, « Les convertisseurs de l'électronique de puissance », Volume 4, Technique et documentation, Lavoisier 1995

## Bibliographie

---

- [12]. Tahir K, Doumi M., « Commande vectorielle d'une machine asynchrone sans capteur mécanique », Mémoire de Master en électrotechnique, Université Dr Moulay Tahar de Saida, 2011.
- [13]. Douadi F., « Méthodologie d'optimisation par les techniques intelligentes d'un contrôleur PID pour un système CSTR », Mémoire de master, Université de Technologie Ferhat-Abbas, Sétif, 2014.
- [14]. Bestaoui A., « Gestion de spectre dans un réseau de radio cognitive en utilisant l'algorithme d'optimisation par essaim de particules », Mémoire de Master en informatique, Université Abou Bakr Belkaid, Tlemcen, 2015.
- [15]. Ait Kaid D., « Commande d'une machine asynchrone sans capteur mécanique, à l'aide de régulateurs fractionnaires », Mémoire de magister, Université Mouloud Mammeri, Tizi-Ouzou, 2011.
- [16]. Attaoui F., « Méthodes d'optimisation pour l'autoréglage de contrôleur PID d'ordre fractionnaire », Mémoire de Master, Université Badji Mokhtar, Annaba, 2019.
- [17]. Benameur L., « Contribution à l'optimisation complexe par des techniques de swarm intelligence », Thèse de Doctorat en Sciences de l'Ingénieur, Informatique et Télécommunication, Université Mohammed V, Rabat, Maroc, 2010.
- [18]. R. Laina, F. Lamzouri, E. Boufounas, A. El Amrani, I. Boumhidi, « Intelligent control of a DFIG wind turbine using a PSO evolutionary algorithm », ScienceDirect, Procedia Computer Science, vol. 127, pp. 471–480, 2018
- [19]. M. Sayed, S. M. Gharghory, H. A. Kamal, « Gain tuning PI controllers for boiler turbine unit using a new hybrid jump PSO », ScienceDirect, Journal of Electrical Systems and Information Technology, vol. 2, pp. 99–110, 2015.

# Annexe

## Annexe

---

### Paramètres de la machine asynchrone utilisée dans la simulation

Nombre de paires de pôles	$p = 2$
Puissance nominale	1.5 Kw
Fréquence nominale	50 Hz
Vitesse nominale	1420 tr/mn
Tension nominale	220/380 V
Courant nominal	6.4/3.7 A
Résistance statorique	$R_s = 4.85 \Omega$
Résistance rotorique	$R_r = 3.805 \Omega$
Inductance statorique	$L_s = 274 \text{ mH}$
Inductance rotorique	$L_r = 274 \text{ mH}$
Inductance mutuelle	$L_m = 258 \text{ mH}$
Moment d'inertie	$J = 0.031 \text{ kg.m}^2$
Coefficient de frottement	$f = 0.00114 \text{ kg.m}^2/\text{s}$

## Programmes Matlab de l'algorithme PSO :

```
tic
clc
clear all
close all
warning off
global Kp Ki
%rng default
LB=[0.5 1];           %Borne inférieure des variables
UB=[2 5];           %Borne supérieure des variables
% Valeur des paramètres PSO
m=2;                 % nombre de variables
n=10;                % taille de la population
wmax=0.9;           % Valeur maximale du coefficient d'inertie
wmin=0.4;           % Valeur minimale du coefficient d'inertie
c1=2;                % Premier facteur d'accélération
c2=2;                % Deuxième facteur d'accélération
% programme principal-----
maxite=20;           % Nombre maximum d'itérations
maxrun=2;            % Nombre maximum d'essais
for run=1:maxrun
    % initialisation du pso -----
    for i=1:n
        for j=1:m
            x0(i,j)=round(LB(j)+rand()*(UB(j)-LB(j)));
        end
    end
    x=x0;             % population initiale
    v=0.1*x0;         % vitesse initiale
    for i=1:n
        f0(i,1)=Objfun(x0(i,:));
    end
    [fmin0,index0]=min(f0);
    pbest=x0;         % valeur initiale de pbest
    gbest=x0(index0,:); % valeur initiale gbest
    % algorithm PSO -----
    ite=1;
    tolerance=1;
    while ite<=maxite && tolerance>10^-12
        w=wmax-(wmax-wmin)*ite/maxite; % MAJ du coefficient d'inertie
        % Mise A Jour de la vitesse des particules
        for i=1:n
            for j=1:m
                v(i,j)=w*v(i,j)+c1*rand()*(pbest(i,j)-x(i,j))...
                    +c2*rand()*(gbest(1,j)-x(i,j));
            end
        end
        % MAJ de la position des particules
        for i=1:n
            for j=1:m
                x(i,j)=x(i,j)+v(i,j);
            end
        end
        % Prendre en charge les violations des bornes
        for i=1:n
            for j=1:m
```

## Annexe

---

```
        if x(i,j)<LB(j)
            x(i,j)=LB(j);
        elseif x(i,j)>UB(j)
            x(i,j)=UB(j);
        end
    end
end
% Calcul de la forme
for i=1:n
    f1(i,1)=Objfun(x(i,:));
end
% Mise à jour de pbest et fitness
for i=1:n
    if f1(i,1)<f0(i,1)
        pbest(i,:)=x(i,:);
        f0(i,1)=f1(i,1);
    end
end
[fmin,index]=min(f0); % Extraire la meilleure particule
ffmin(ite,run)=fmin; % Sauvegarder la meilleure forme
ffite(run)=ite; % Sauvegarder la valeur de l'itération
% Mise à jour de gbest et de la meilleure forme
if fmin<fmin0
    gbest=pbest(index,:);
    fmin0=fmin;
end
% calcul de la tolérance
if ite>100;
    tolerance=abs(ffmin(ite-100,run)-fmin0);
end
% affichage des résultats de l'itération
if ite==1
    disp(sprintf('Iteration Best_Kp Best_Ki Fonct_obj'));
end
disp(sprintf('%8g %8.4f %8.4f %8.4f',ite,gbest(1),gbest(2),fmin0));
ite=ite+1;
end
gbest;Kp=gbest(1);Ki=gbest(2);
fvalue=Objfun([Kp Ki]);
fff(run)=fvalue;
rgbest(run,:)=gbest;
disp(sprintf('-----'));
end
disp(sprintf('\n'));
disp(sprintf('*****'));
disp(sprintf('Final Results-----'));
[bestfun,bestrun]=min(fff)
best_Kp=rgbest(bestrun,1)
best_Ki=rgbest(bestrun,2)
disp(sprintf('*****'));
toc
% Caractéristiques de convergence du PSO
plot(ffmin(1:ffite(bestrun),bestrun),'-b','LineWidth',3);
xlabel('Iteration');
ylabel('Fitness function value');
title('PSO convergence characteristic')
```