

CHAPITRE III

COMMANDE PAR LA LOGIQUE FLOUE DE LA MACHINE ASYNCHRONE TRIPHASEE

3.1. INTRODUCTION

La logique floue est une nouvelle méthode de réglage qui a été introduite à large échelle au Japon. Cependant, plus récemment elle est de plus en plus appliquée en Europe.

Les méthodes de réglage conventionnelles se basent sur une modélisation adéquate du système et un traitement analytique à l'aide de fonction de transfert ou d'équations d'état. Par contre le réglage par logique floue donne une approche plus tôt programmatique, permettant d'inclure les expériences acquises par les opérateurs.

En effet les bases théoriques de la logique floue ont été établies en 1965 par le professeur L. A. Zadeh de l'université de Californie de Berkeley, à cette époque cette nouvelle théorie n'a pas été prise au sérieux, elle permet le traitement des variables non exactes dont la valeur peut varier entre 0 et 1.

Cette nouvelle théories a été appliquée initialement dans des domaines non techniques, tel que le commerce ou la médecine, dans le but de compléter les systèmes experts. Les premières applications de la logique floue au niveau des systèmes de réglage ne parurent que dès 1975. Ce sont les Japonais qui commencent à l'utiliser dans des produits industriels pour résoudre les problèmes de réglage et de commande, et récemment en Europe, une intense activité de recherche à été introduite afin d'introduire ce principe de réglage.

Dans ce chapitre, nous avons étudié la conception d'un réglage ou d'une commande par logique floue en décrivant les différents éléments constitutifs d'un régulateur flou. Le principe de la commande floue est ensuite abordé en présentant les différents types de régulateurs flous, puis on donnant leurs avantages et leurs inconvénients.

3.2. REGLAGE PAR LOGIQUE FLOUE

3.2.1. Structure d'un réglage par logique floue :

Un réglage par logique floue comporte : le système à régler, l'organe de commande ainsi que le régulateur par logique flous. Il est représenté par la figure suivante:

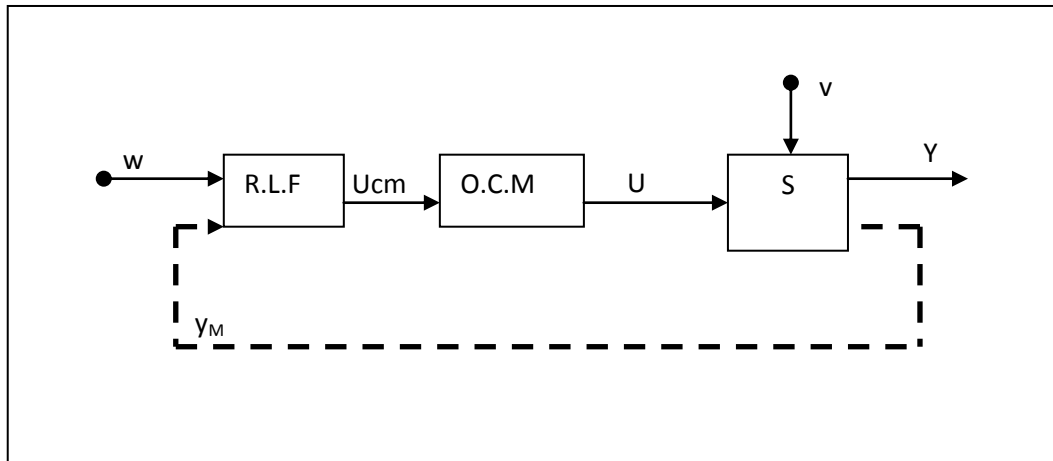


Fig.3.1 : Structure d'un régulateur par logique floue.

- S : Système à régler
- OCM : Organe de commande
- RLF : Régulateur par logique floue
- w : Grandeur de consigne
- u_{cm} : Signal de commande fourni par le RLF
- u : Grandeur de commande fournie par l'OCM
- v : Perturbation
- y : Grandeur à régler (ou sortie)
- y_M : Vecteur contenant les grandeurs mesurées

On constate une certaine affinité avec le réglage par contre réaction d'état, en général, le nombre des grandeurs réunies dans le vecteur y_M est inférieur au nombre des grandeurs d'état.

3.2.2. Structure et configuration interne d'un régulateur par logique floue :

Le régulateur par logique floue ne traite pas une relation mathématique bien définie, mais utilise des inférences avec plusieurs règles se basant sur des variables linguistiques.

Le régulateur par logique floue comprend les quatre modules suivants :

- * l'interface de fuzzification,
- * la base de connaissances,
- * le moteur d'inférence floue,
- * l'interface de défuzzification.

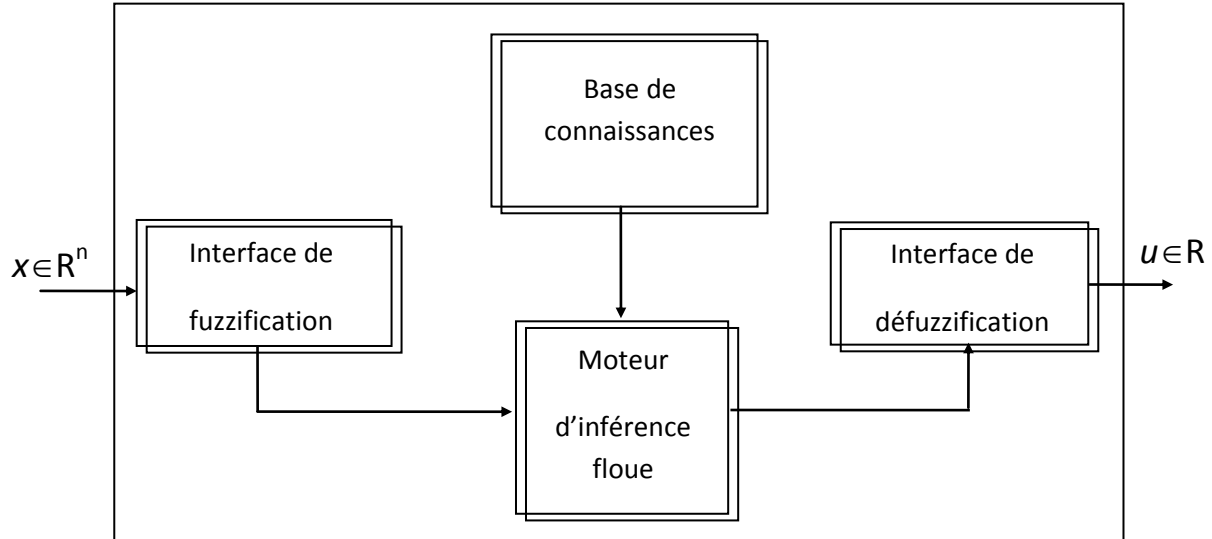


Fig3.2 : Structure de base d'un RLF

3.2.2.1. Interface de fuzzification :

Le choix de l'opérateur de fuzzification dépend de la confiance que l'on accorde aux mesures effectuées. Ainsi si la mesure x_0 est exacte, les sous ensemble floue X_0 doit être représentée par un fait précis. Par conséquent, on utilise comme opérateur de fuzzification la transformation dite de singleton.

La fonction d'appartenance de A' est telle que :

$$\mu_{A'}(x) = \begin{cases} 1 & \text{pour } x = x_0 \\ 0 & \text{pour } x \neq x_0 \end{cases} \quad (3.1)$$

L'aspect de cette fonction d'appartenance est illustré dans la figure (3.3)

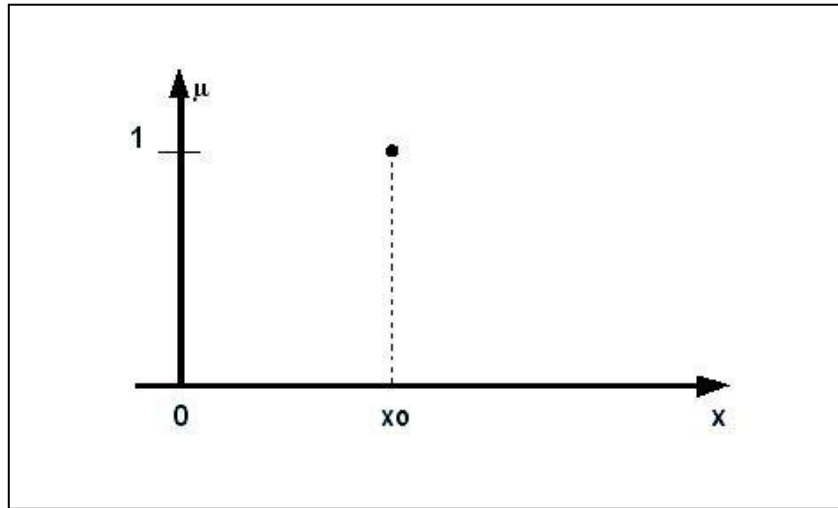


Fig.3.3 : Fuzzification pour une mesure exacte.

Par contre, si la mesure de la variable est incertaine, par exemple à cause de bruit, le sous ensemble flou X_0 doit être représentée par un fait imprécis. On utilise alors une méthode de fuzzification qui associe à la variable mesurée x_0 une fonction d'appartenance telle que, par

exemple:

$$\mu_{x_0}(x) = \max \left\{ 0, 1 - \frac{|x - x_0|}{\varepsilon} \right\} \quad (3.2)$$

La représentation graphique de cette fonction est représentée par la *figure (3.4)*

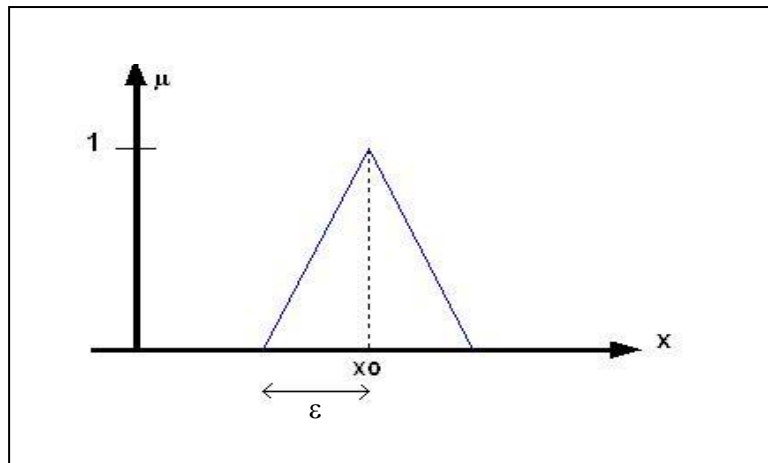


Fig.3.4 : Méthode de fuzzification pour mesure incertaine.

3.2.2.2. Base de connaissances :

Elle comprend la base de données et la base des règles floues.

3.2.2.2.1. La base de données :

Elle Contient la définition des ensembles flous, les facteurs d'échelle pour la normalisation des ensembles de référence et la partition de l'espace flou d'entrée et de sortie.

3.2.2.2. La base des règles :

Le comportement dynamique d'un système est caractérisé par un ensemble de règles linguistiques basé sur la connaissance d'un expert. Les connaissances d'expert sont habituellement de la forme :

Si (*un ensemble de conditions est satisfait*)

alors

(*un ensemble de conséquences peut être induit*)

La collection des règles floues de commande, qui sont exprimées sous la forme de règles floues « *si - alors* », constituent la base des règles ou l'ensemble des règles du *RLF*.

Principalement, les règles de commande floue constituent un moyen approprié pour exprimer la stratégie de commande et les connaissances dans le domaine en question.

De plus, plusieurs variables peuvent intervenir dans les antécédents et les conséquences de ces règles. Dans ce cas, le système est du type multivariable flou (*MIMO*). Par exemple, dans le cas d'un système flou à deux entrées avec une sortie, les m règles floues de commande ont la forme :

R_1 : si x est A_1 et y est B_1 alors u est C_1

R_2 : si x est A_2 et y est B_2 alors u est C_2

.....

.....

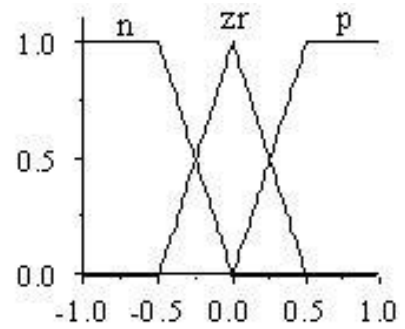
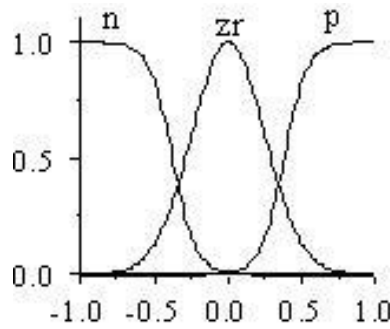
R_m : si x est A_m et y est B_m alors u est C_m

Les variables linguistiques pour un *RLF* sont les variables d'état, l'erreur de la variable d'état (e), la variation de erreur (de), l'intégration de cette erreur etc.

Les règles sont regroupées sous forme de table donnant la valeur de la commande sous la forme linguistique en fonction des deux entrées (e , de)

- Si Les deux variables d'entrée et la sortie sont décrites par trois ensembles flous :

	N	ZR	P
N	N	N	ZR
ZR	N	ZR	P
P	ZR	P	P



3.2.2.3. Le moteur d'inférence :

C'est un mécanisme de décision. Il permet à partir d'un fait observé de la base des règles floues une décision en exploitant le raisonnement approximatif. Dans les inférences de régulateur par logique floue interviennent les opérateurs ET et OU. L'opérateur ET s'applique aux variables à l'intérieur d'une règle tandis que l'opérateur OU lie les différentes règles.

3.2.2.3.1. Structure générale de l'inférence :

« **SI** l'entrée **x** appartient au **sous-ensemble A** **ET** que l'entrée **y** appartient au **sous-ensemble B** **ALORS** la sortie appartient au **sous-ensemble C**. »

Il existe plusieurs méthodes pour calculer la valeur représentative d'un ensemble de sortie, dont les principales sont:

- Méthode d'inférence *MIN / MAX*
- Méthode d'inférence *MAX / PROD*
- Méthode d'inférence *SOMME / PROD*

3.2.2.3.2. Méthodes d'inférence MIN/MAX :

C'est la méthode la plus universelle mais qui n'est guère applicable en raison du temps de calcul très long.

Cette méthode réalise, au niveau de la condition, l'opérateur **OU** par la formation maximum et l'opérateur **ET** par la formation du minimum. La conclusion dans chaque règle, introduite par **ALORS**, lie le facteur d'appartenance de la condition avec la fonction

d'appartenance de la variable de sortie **xr** par l'opérateur **ET**, réalisé dans le cas présent par la formation du minimum.

Enfin l'opérateur **OU** qui lie les différentes règles est réalisé par la formation du maximum.

3.2.2.3.3. Méthode d'inférence MAX/PROD :

Cette méthode réalise, en général, au niveau de la condition, l'opérateur **OU** par la formation maximum et l'opérateur **ET** par la formation du minimum. Par contre, la conclusion dans chaque règle, introduite par **ALORS**, lie le facteur d'appartenance de la condition avec la fonction d'appartenance de la variable de sortie **xr** par l'opérateur **ET**, est réalisé cette fois-ci par la formation du produit.

Enfin l'opérateur **OU** qui lie les différentes règles est réalisé par la formation du maximum.

3.2.2.3.4. Méthode SOMME/PROD :

Cette méthode est la plus utilisée car son temps de calcul est court. Par opposition aux méthodes d'inférences précédentes, cette méthode réalise, au niveau de la condition, l'opérateur **OU** par la formation de la somme, plus précisément par la valeur moyenne, tandis que l'opérateur **ET** est réalisé par la formation du produit.

La conclusion de chaque règle, introduite par **ALORS**, liant le facteur d'appartenance de la condition avec la fonction d'appartenance de la variable de sortie **xr** par l'opérateur **ET**, est réalisé par la formation du produit. Enfin l'opérateur **OU** qui lie les différentes règles est réalisé par la formation de la somme, donc de la valeur moyenne.

3.2.2.4. La défuzzification :

La défuzzification est l'opération qui consiste à extraire d'un ensemble flou une valeur numérique considérée comme valeur significative. Dans le domaine de la commande en temps réel, un critère de choix de la méthode de défuzzification est la simplicité de calcul.

3.2.2.4.1. Méthode du Maximum :

La méthode du maximum fournit en sortie comme valeur significative u , celle correspondant au maximum de la fonction d'appartenance $\mu_C(u)$. Cependant, cette méthode

présente un certain inconvénient lorsqu'il existe plusieurs valeurs pour lesquelles $\mu_{C'}(u)$ est maximale.

3.2.2.4.2. Méthode de la Moyenne des Maximas :

La défuzzification détermine les valeurs de u pour lesquelles la fonction d'appartenance est maximale, ensuite calcule la moyenne entre elles comme résultat de la défuzzification.

3.2.2.4.3. Méthode du Centre de Gravité :

La plus fréquente est le centre de gravité de la surface de l'ensemble flou $\mu_{C'}(u)$ qui

est défini par :

$$u = \frac{\int_U \mu_{C'}(u) \cdot u du}{\int_U \mu_{C'}(u) du} \quad (3.2)$$

Où $\mu_{C'}(u)$ est la fonction d'appartenance de l'ensemble flou C' représentant la sortie globale du système flou. Souvent on utilise une version discrète de l'intégrale :

$$u = \frac{\sum_i u_i \cdot \mu_{C'}(u_i)}{\sum_i \mu_{C'}(u_i)} \quad (3.3)$$

Il est à noter que le centre de gravité est généralement difficile à calculer. Aussi, cette méthode exige un temps de calcul important.

3.2.2.4.4. Méthode des Hauteurs Pondérées :

Cette méthode effectue une moyenne pondérée entre les centres de gravité des conséquences des règles :

$$u = \frac{\sum_{i=1}^m w_i c_i}{\sum_{i=1}^m w_i} \quad (3.4)$$

Où c_i est le centre de gravité de l'ensemble C_i conséquence de la i ème règle et w_i le degré d'activation de cette même règle.

Cette méthode est très simple à implémenter, du fait que le centre de gravité de la fonction d'appartenance de la conséquence de chaque règle est connu à priori.

3.2.2.4.5. Méthode des Hauteurs Pondérées Modifiée :

On fait intervenir dans la détermination de la sortie u en plus du centre de gravité ci la largeur δ_i de chaque fonction d'appartenance de la conséquence par :

$$u = \frac{\sum_{i=1}^m w_i c_i / \delta_i^2}{\sum_{i=1}^m w_i / \delta_i^2} \quad (3.5)$$

La largeur δ_i de la fonction d'appartenance $\mu_{C_i}(u)$ peut être prise comme la base (ou le support) des fonctions d'appartenance du type triangulaire ou trapézoïdale et comme la variance des fonctions d'appartenance du type gaussienne.

3.3. APPLICATION DE LA LOGIQUE FLOUE A LA COMMANDE DE LA MAS

Dans cette section, nous présentons deux types de régulateurs flous pour le réglage de la vitesse d'une MAS à savoir : Le régulateur de type Mamdani et le régulateur de type Sugeno.

3.3.1. Régulateur de type Mamdani :

En 1974, E.H. Mamdani est le premier qui a appliqué la technique de réglage par la logique floue utilisant le premier régulateur flou, celle-ci est construite d'un organe de décision utilisant des règles subjectives et imprécises données par des experts qui connaissent bien le système.

3.3.1.1. Synthèse du régulateur :

La majorité des régulateurs utilisent le schéma simple proposé par Mamdani [32] pour les systèmes MIMO mono-entrée/ mono-sortie. Ce schéma est représenté par la figure (3.5) suivante :

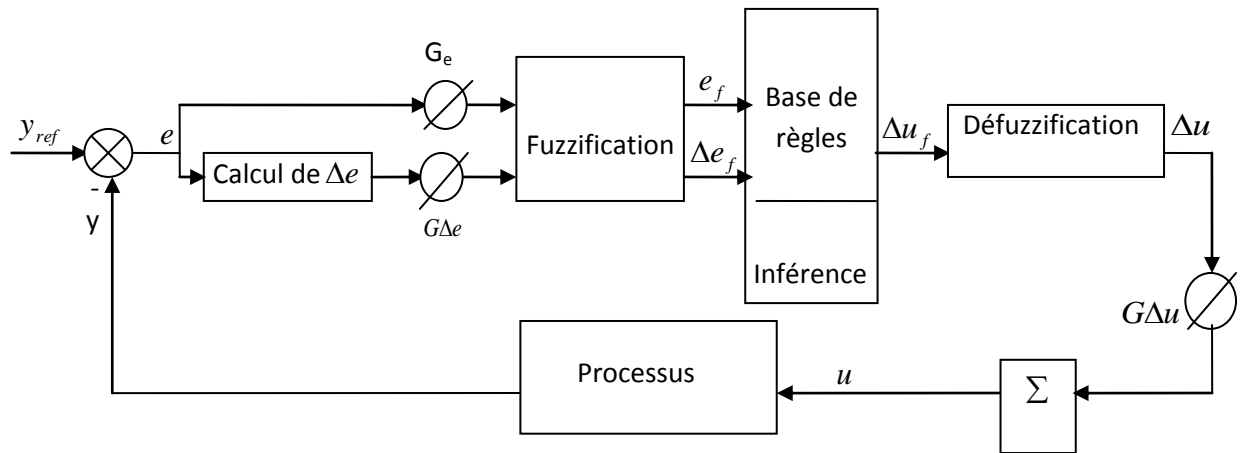


Fig.3.5 : Schéma bloc de régulation par régulateur flou.

D'après ce schéma le système est composé :

- d'un bloc de fuzzification utilisé pour convertir l'erreur et sa variation en valeur floues ;
- des règles du régulateur flou ;
- d'un bloc de défuzzification utilisé pour convertir la variation de la commande floue en valeur numérique ;
- d'un bloc intégrateur.

Les méthodes choisies pour le succès des algorithmes flous permettent de formuler un ensemble de décisions en termes linguistiques, utilisant les ensembles flous pour décrire l'amplitude de l'erreur, de sa variation et de la commande appropriée. En combinant ces règles, on peut dresser des tables de décisions permettant de donner les valeurs de la sortie du régulateur correspondant aux situations d'intérêt.

Les facteurs d'échelle doivent être choisis sur la base de l'étude du système de sorte que, lors de petits phénomènes transitoires, le domaine admissible pour l'erreur et sa variation ne soit pas dépassé [33], [34].

La configuration d'un système flou multi-entrées muti-sorties de type Mamdani permet de représenter une relation non linéaire entre un ensemble d'entrée,

noté, $x = [x_1, \dots, x_n]^T \in X \subseteq \mathcal{R}^n$, et une sortie $y \in \mathcal{R}$, avec $X = X_1 \times \dots \times X_n$ et $X_i \in \mathcal{R}$.

Cette relation est décrite par un ensemble de règles floues de la forme suivante [35], [36] :

R_K : SI x_1 est F_1^K et.....et x_n est F_n^K ALORS y est B^K (K=1.....M)

Où $F_i^K \in \{F_i^1, \dots, F_i^{m_i}\}$; avec F_i^j , $j=1, \dots, m_i$, $i=1, \dots, n$ sont des ensembles flous définis sur X_i , B^K des ensembles flous définis dans \mathfrak{R} , et $M = \prod_{i=1}^n m_i$ est le nombre total des règles du système flou, i.e. la base de connaissance comporte toutes les combinaisons possibles des ensembles flous des variables d'entrée.

Si on choisit la fuzzification singleton et on considère le connecteur « ET » comme le produit algébrique, le degré de vérité ou d'activation de chaque règle est donné par [35] :

$$\mu_K(x) = \prod_{i=1}^n \mu_{F_i^K}(x_i) ; \mu_{F_i^K} \in \{ \mu_{F_i^1}, \dots, \mu_{F_i^{m_i}} \} \quad (3.6)$$

Avec $\mu_{F_i^K}(x_i)$ la fonction d'appartenance associée à l'ensemble flou F_i^j .

En utilisant la méthode de défuzzification du centre de gravité, la sortie finale du système flou est donnée par [35], [36] :

$$y = \frac{\sum_{K=1}^M \mu_K \bar{y}}{\sum_{K=1}^M \mu_K} \quad (3.7)$$

Les avantages du système flou de Mamdani peuvent être résumés par les points suivants:

- 1- La simplicité dans la représentation des règles floues : Dans ce type de système les conséquences aussi bien que les prémisses des règles flous prennent la forme d'ensembles flous, ce qui les rends facile à interpréter.
- 2- La flexibilité dans l'implémentation : Ceci est du à la liberté dans le choix des formes d'implication floues, du fuzzificateur et du défuzzificateur.

Le principal inconvénient de ce type de système est le nombre élevé de règles floues surtout lorsque le système est non linéaire et complexe.

3.3.2. Régulateur de la vitesse

Pour le réglage de la vitesse d'un moteur asynchrone, on utilise une structure à deux régulateurs de même type (régulateur de type Mamdani), ces régulateurs sont le régulateur flou de vitesse et le régulateur flou de flux.

Les sorties des deux régulateurs sont les courants statoriques de référence suivant l'axe q et d (i_{qsref}, i_{dsref}), et après la transformation inverse de Park on aboutit aux courants réels statoriques qui donnent la vitesse réelle du moteur.

La structure globale de cette commande est représentée par la figure (3.6).

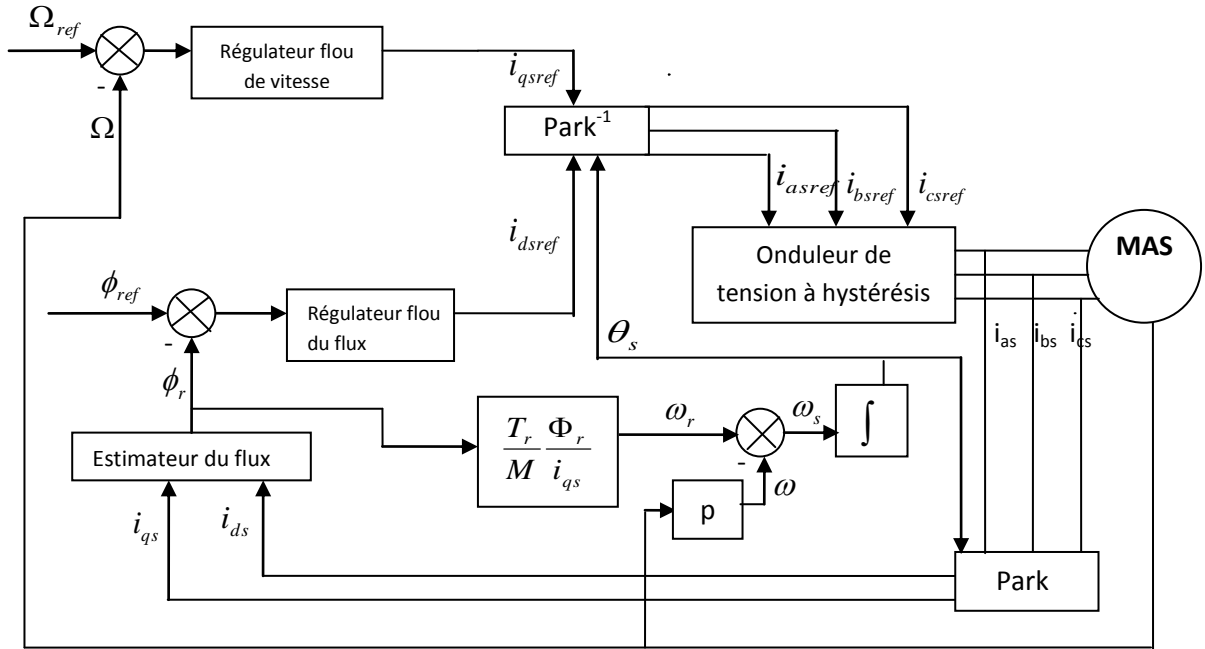


Fig.3.6 : Structure globale du réglage de vitesse par la logique floue de la MAS

Les deux régulateurs possèdent des fonctions d'appartenance identiques, la différence réside dans les gains de normalisation (facteurs d'échelle).

Le calcul de la commande pour les deux régulateurs est le même. Il se déroule comme suit:

- 1- échantillonner la sortie
- 2- calculer l'erreur notée par e_i :
- 3- calculer la variation de cette erreur notée par Δe_i :

$$e_i = \text{consigne} - \text{sortie} \quad (3.8)$$

$$\Delta e_i = e_i - e_{i-1} \quad (3.9)$$

Où : e_{i-1} est l'erreur à l'instant précédent.

- 4- calculer les valeurs de normalisation de e_i et Δe_i par :

$$\begin{aligned} X_{e_i} &= G_e e_i \\ X_{\Delta e_i} &= G_{\Delta e} \Delta e_i \end{aligned} \quad (2.28)$$

Où : G_e et $G_{\Delta e}$ représentent les facteurs d'échelle (normalisation), on fait varier ces facteurs jusqu'à ce qu'on trouve un phénomène transitoire de réglage convenable. En effet, ces derniers en un grand effet sur les performances de la commande.

5- calculer la variation de la commande Δu_i

Pour cela on parcourt les étapes suivantes :

5.1- fuzzification :

Calculer les degrés d'appartenance de e_i et Δe_i pour les différentes classes à l'aide de leurs fonctions d'appartenance.

5.2- inférence floue :

Calculer les fonctions d'appartenance résultante de la variable linguistique Δu_i à Chacune de ces classes.

5.3- défuzzification:

Par cette étape se fait retour aux grandeurs de sorties réelles. Il s'agit de calculer, à partir des degrés d'appartenance à tous les ensembles flous de la variable de sortie, l'abscisse qui correspond à la valeur de cette sortie. Différentes méthodes sont utilisées, parmi eux la méthode du centre de gravite de la fonction d'appartenance résultante de l'inférence correspond à la valeur de sortie du régulateur est:

$$\Delta u_k = \frac{\sum_{k=1}^3 u_k r_k}{\sum_{k=1}^3 \mu_k} \quad (3.10)$$

avec: μ_k : degré d'activation de la k^{eme} regle

r_k : L'abscisse du centre de gravite de la k^{eme} classe

6- calculer la sortie physique de la commande par:

$$u_k = u_{k-1} + G_{\Delta u} \Delta u_k \quad (3.11)$$

Où

$G_{\Delta u}$: Gain associé à la commande u_k

Δu_k : Variation de la commande

Dans cette partie, nous allons nous intéresser à la conception d'un régulateur flou pour la commande de la vitesse d'un moteur asynchrone. Comme mentionné précédemment

dans la théorie de la logique floue, le nombre de combinaison et de variantes est quasiment infini. Devant ce large éventail, notre choix sera d'abord imposé par la simplicité de mise en œuvre. Nous recherchons un régulateur flou que l'on peut implanter au sein de ce type de commande. Une des contraintes est la limitation du temps de calcul tout en conservant les propriétés du régulateur flou.

Compte tenu de ce qui vient d'être dit, nous avons retenu pour le régulateur:

- un nombre limité à trois ou sept ensembles flous pour chaque variable;
- des variables d'entrée dont les fonctions d'appartenance des ensembles flous sont de formes triangulaires;
- des singletons pour les fonctions d'appartenance de la variable de sortie;
- des gains variables à l'entrée et à la sortie du régulateur permettant d'ajuster son fonctionnement et de varier sa plage de sensibilité.

3.3.2.1. Régulateur à trois ensembles flous

Le régulateur admet pour les variables d'entrée et la variable de sortie trois ensembles flous NG (Négatif grand) ZE (Environ zéro) et PG (Positif grand) comme c'est montré par la figure (3.7):

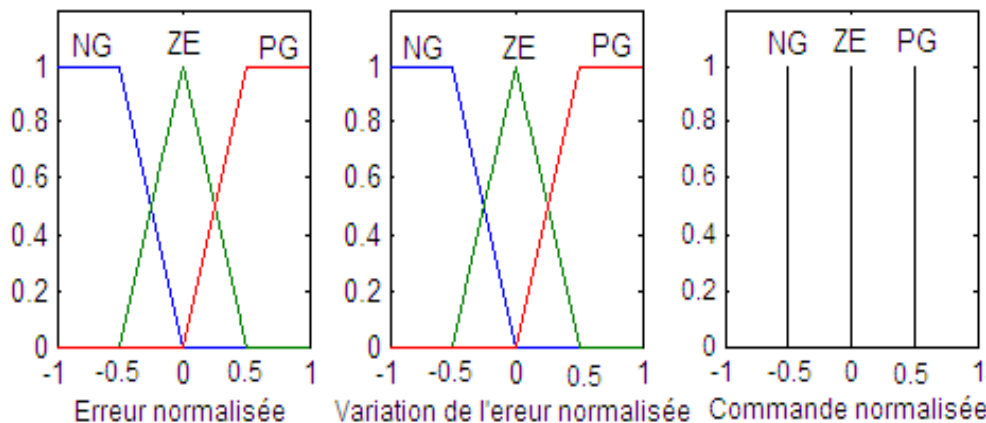


Fig.3.7: Fonction d'appartenance des différentes variables linguistiques.

Les règles floues relient le signal de sortie avec les signaux d'entrées par des conditions linguistiques et l'établissement de ces règles sont généralement basés sur l'expérience de l'opérateur et / ou du savoir-faire de l'ingénieur en régulation et commande.

L'évaluation des règles floues étant une opération déterministe, il est tout à fait envisageable de mettre sous forme de tableau (3.1) ce régulateur.

e $\Delta \dot{e}$	NG	ZE	PG
NG	NG	NG	ZE
ZE	NG	ZE	PG
PG	ZE	PG	PG

Tableau.3.1: Base de règle

Les valeurs contenues dans ces cellules ont été déterminées de façon logique en étudiant toutes les combinaisons possibles des variables d'entrée.

Par exemple, il est clair que, si l'erreur est fortement négative et sa variation aussi, le signal de commande doit être également. Au contraire, si l'erreur est environ zéro et sa variation aussi, il en sera de même de la commande. Si l'erreur est environ zéro et sa variation est fortement négative ou si l'erreur est fortement négative et sa variation est environ zéro, le signal de commande doit être fortement négatif.

3.3.2.2. Résultats de simulation

Afin de valider la méthode de commande floue avec trois fonctions d'appartenance pour chaque variables d'entrées et de sortie appliquée à une machine asynchrone triphasée soumis à une perturbation de couple de 10Nm puis une inversion du sens de marche de - 1000tr/min à 1000tr/min. les figures (3.8.a) et (3.8.b) illustrent les courbes de vitesse, le couple et les flux rotoriques ainsi que les courants statoriques suivant les deux axes (d q), pour une durée de perturbation de 0.5s.

D'après les résultats de simulation obtenus, on constate que la vitesse suit sa valeur de référence sans dépassement et l'application d'un couple résistant n'affecte guère la vitesse de rotation souhaitée. Le flux rotorique est installé, il suit sa valeur de référence suivant l'axe (d) avec une composante en quadrature (q) nulle. L'application du couple résistant conduit à une augmentation du couple électromagnétique développé ainsi que le courant statorique qui a un comportement sinusoïdal. L'analyse des résultats montre que les performances de poursuite de la consigne sont satisfaisantes ; un rejet total de la perturbation et un démarrage sans dépassement au bout de 0.25s.

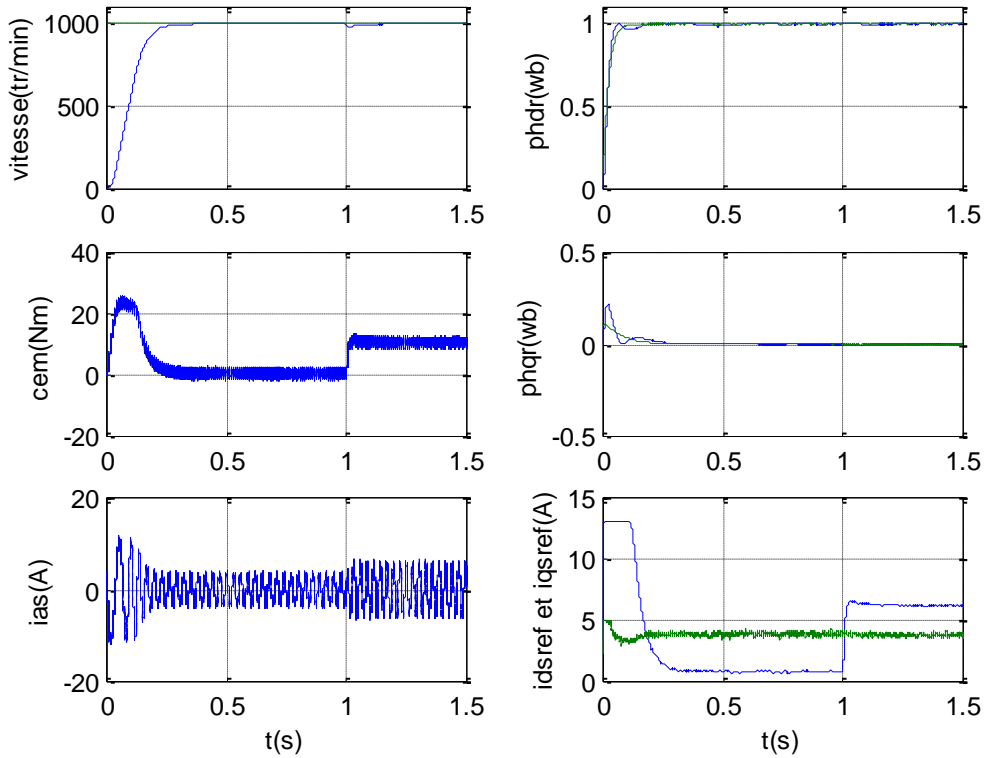


Fig.3.8.a : Comportement dynamique de la MAS lors d'un démarrage avec variation de la charge .

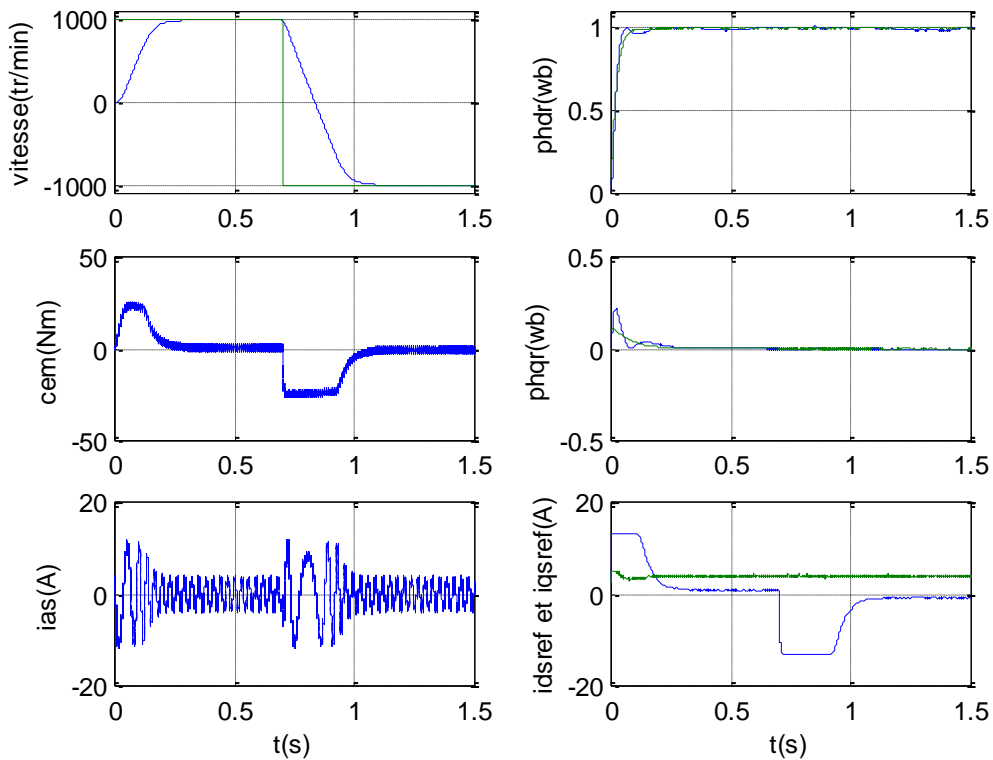


Fig.3.8.b : Comportement dynamique de la MAS lors d'une inversion du sens de la marche.

3.3.2.3. Régulateur à sept ensembles flous

Dans le cas du régulateur à trois ensembles flous, les réglages se font surtout par l'action sur les facteurs d'échelle et à la sortie du régulateur. Par contre, si on passe à un nombre plus important d'ensembles flous, sept par exemple, un choix plus large s'offre notamment sur la matrice d'inférence et la répartition des fonctions d'appartenance. Bühler [37] insiste bien sur le fait que ce n'est pas le choix des opérateurs pour réaliser l'inférence qui est important mais plutôt la matrice d'inférence elle-même et surtout, la répartition des fonctions d'appartenance sur l'univers de discours. On peut en effet obtenir des caractéristiques non linéaires très prononcées. Cependant l'optimisation pour une application donnée peut être longue et fastidieuse. Il convient de ne faire varier que la forme ou la distribution des fonctions d'appartenance, soit des variables d'entrée, soit de la sortie, sinon on peut aboutir à une certaine compensation des régulateurs envisagés.

Nous avons utilisé dans cette partie une répartition uniforme des fonctions d'appartenance sur l'univers de discours des entrées et de la sortie, Les fonctions sont symétriques par rapport à zéro comme montré à la figure (3.9):

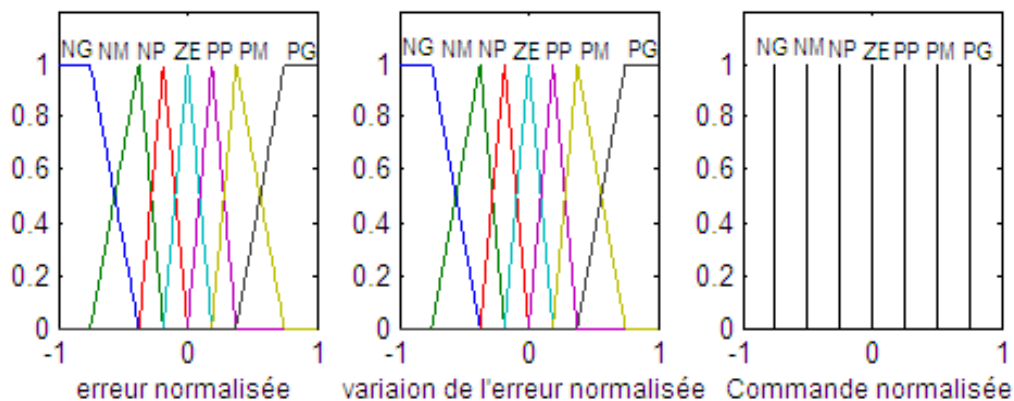


Fig.3.9: Fonctions d'appartenance des différentes variables linguistiques

En utilisant les directives de synthèse de règles données dans la section précédente. La matrice d'inférence obtenue sera décrite par le tableau (3.2) suivant:

Δe	NG	NM	NP	ZE	PP	PM	PG
NG	NG	NG	NG	NG	NM	NP	ZE
NM	NG	NG	NG	NM	NP	ZE	PP
NP	NG	NG	NM	NP	ZE	PP	PM
ZE	NG	NM	NP	ZE	PP	PM	PG
PP	NM	NP	ZE	PP	PM	PG	PG
PM	NP	ZE	PP	PM	PG	PG	PG
PG	ZE	PP	PM	PG	PG	PG	PG

Tableau.3.2: Base de règles

Les ensembles flous sont notés comme suit:

- NG : Négatif grand
- NM : Négatif moyen
- NP : Négatif petit
- ZE : Environ zéro
- PP : Positif petit
- PM : Positif moyen
- PG : Positif grand

3.3.2.4. Résultats de simulation

Dans cette partie, nous simulons le comportement dynamique de la machine asynchrone lors d'un démarrage pour une consigne de 1000 tr/min avec variation de charge et puis lors d'une inversion du sens de marche de 1000 tr/min à -1000 tr/min avec le régulateur flou de type Mamdani à sept ensembles flous de formes triangulaires appliqués aux variables d'entrées et des singletons pour la variable de sortie. Il apparaît, suite aux différents résultats présentés aux figures (3.10.a) et (3.10.b), que les performances de poursuite de la consigne sont satisfaisantes et le rejet de la perturbation est très rapide.

On remarque, que les résultats sont identiques à ceux obtenus par un régulateur à trois ensembles flous. De ce fait, il est judicieux de travailler uniquement avec trois ensembles flous pour réduire la taille du régulateur et par conséquent le temps de calcul.

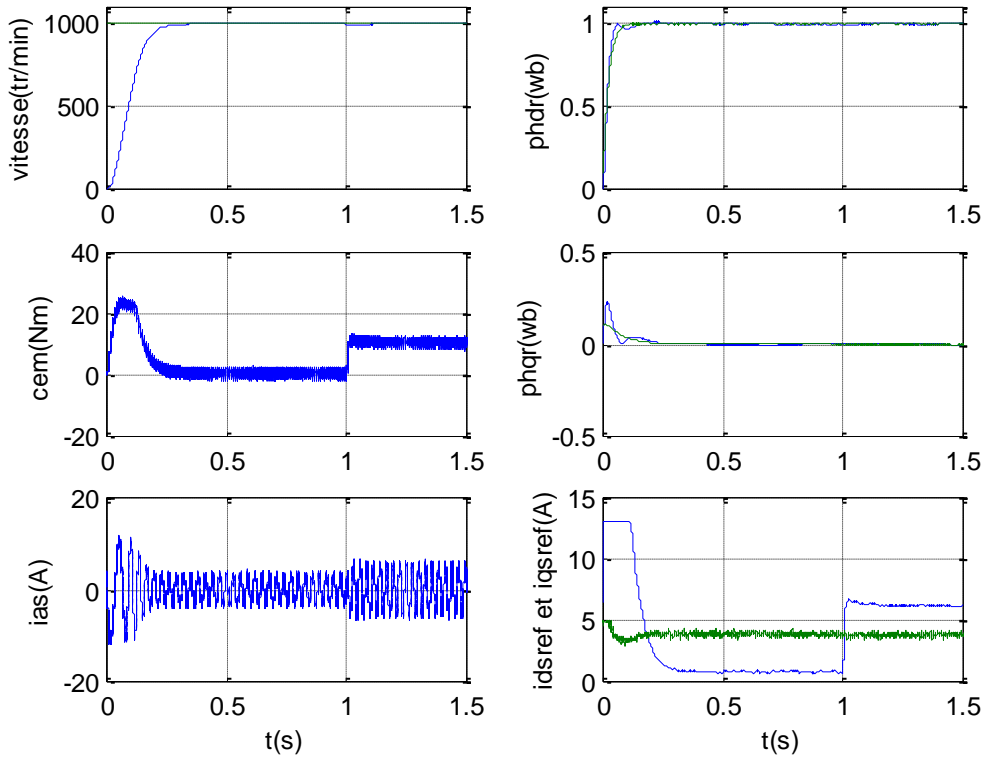


Fig.3.10.a: Comportement dynamique de la MAS lors d'un démarrage avec variation de la charge

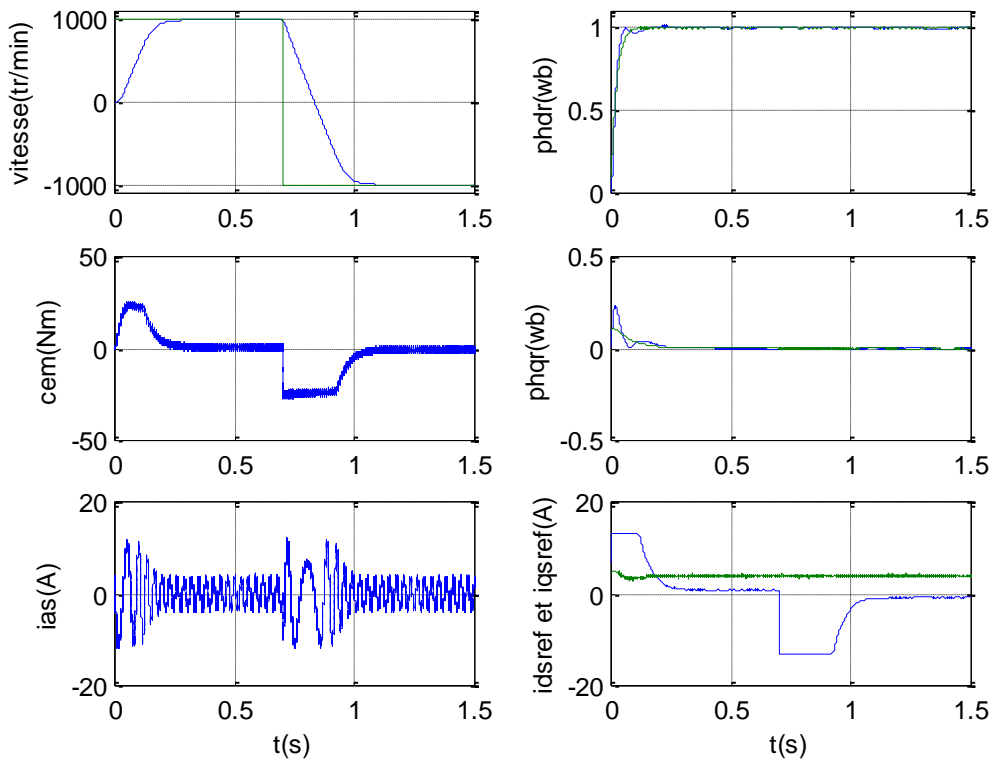


Fig.3.10.b: Comportement dynamique de la MAS lors d'une inversion de vitesse.

3.3.3. Régulateur de type Sugeno

Dans la section précédente, nous avons présenté les régulateurs flous à conclusions symboliques appelés régulateur de type Mamdani, ces régulateurs utilisent en effet des conclusions symboliques de même nature que les prémisses et l'obtention de meilleures performances pour ce type de régulateur dû à l'augmentation de nombres de classes. Cependant l'augmentation du nombre de règles rend difficiles l'obtention de ces règles et augmente aussi le temps de traitement. Donc, pour diminuer le temps de calcul, il faut réduire la taille du régulateur en minimisant le nombre de règles constituant la base de connaissance. Pour résoudre ce problème, Sugeno et ses collaborateurs ont proposé des systèmes flous à conclusion fonctionnelle ou système flou de Takagi-Sugeno (TS). Ces régulateurs cherchaient à développer une méthode systématique de génération des règles floues à partir d'un ensemble de données entrée-sortie.

Notons par $x = [x_1, \dots, x_n]^T$ les entrées du système flou, et par y sa sortie. Pour chaque x_i est associée m_i ensembles flous F_i^j dans X_i , tel que pour $x_i \in X_i$, il existe au moins un degré d'appartenance $\mu_{F_i^j}(x_i) \neq 0$, ou $i=1,2,\dots,n$ et $j=1,2,\dots,m_i$. La base de règles du

système flou comporte $M = \prod_{i=1}^n m_i$ règles floues de la forme [37]:

$$R_K : \text{SI } x_1 \text{ est } F_1^K \text{ et } \dots \text{ et } x_n \text{ est } F_n^K \text{ ALORS } y = f_K(x) \quad (2.34)$$

Où $F_i^K \in \{F_i^1, \dots, F_i^{m_i}\}$ et $f_K(x)$ est une fonction numérique

Dans l'espace de sortie. En général $f_K(x)$ est une fonction polynomiale en fonction des variables d'entrées.

Si $f_K(x)$ est une fonction linéaire de la forme :

$$f_K(x) = a_0^K + \sum_{i=1}^n a_i^K x_i \quad (3.12)$$

Alors nous avons un système flou de Takagi-Sugeno d'ordre 1 (TS1).

Si par contre, $f_K(x)$ est un polynôme d'ordre zéro, i.e.

$$f_K(x) = a^K \quad (3.13)$$

On a donc un système flou de Takagi-Sugeno d'ordre zéro (TS0).

Etant donnée que chaque règle possède une conclusion numérique, la sortie totale du système flou est obtenue par le calcul d'une moyenne pondérée, et de cette manière, le temps consommé par la procédure de défuzzification est évité. En fait, la sortie du système flou est donnée par la relation suivante [36] [38] :

$$y(x) = \frac{\sum_{K=1}^M \mu_K(x) f_K(x)}{\sum_{K=1}^M \mu_K(x)} \quad (3.14)$$

$$\mu_K(x) = \prod_{i=1}^n \mu_{F_i^K}(x) , \quad F_i^K \in \{F_i^1, \dots, F_i^{m_i}\}$$

qui représente le degré de confiance ou d'activation de la règle R_K .

Dans le cas d'un système flou TS1, la sortie est donnée par:

$$y(x) = \frac{\sum_{K=1}^M \mu_K(x) [a_0^K + a_1^K x_1 + \dots + a_n^K x_n]}{\sum_{K=1}^M \mu_K(x)} \quad (3.15)$$

Et dans le cas d'un système flou TS0, la sortie se simplifie à

$$y(x) = \frac{\sum_{K=1}^M \mu_K(x) a^K}{\sum_{K=1}^M \mu_K(x)} \quad (3.16)$$

3.3.3.1. Conception d'un régulateur flou

La conception d'un régulateur flou revient à déterminer sa structure et le choix de ces paramètres (phase d'acquisition). La conception d'un régulateur flou nécessite le choix des paramètres suivants [39]:

- définition des variables d'E/S du système ;
- choix de la partition floue, c'est-à-dire, l'association d'un ensemble de termes linguistiques caractérisés par des fonctions d'appartenance définies sur des univers de discours appropriés pour chaque variable d'E/S ;
- la forme des fonctions d'appartenance de l'espace d'E/S (triangulaire, gaussienne,.....etc) ;
- la base de règles ;

- le choix de la méthode d'inférence et de la stratégie de défuzzification.

Les systèmes flous utilisés pour la commande des procédés industriels sont caractérisés approximativement et qualitativement par la pensée humaine. Cependant leurs performances sont liées à deux facteurs importants:

- 1- la disponibilité de l'expertise (le savoir-faire) ;
- 2- La validité des techniques d'acquisition de connaissances et la justesse des données acquises.

L'inconvénient majeur de l'approche traditionnelle pour la conception des régulateurs flous réside au niveau de l'obtention des connaissances auprès d'un expert humain. Pour palier ce problème, la recherche s'est orientée vers l'élaboration des techniques optimales et systématiques pour la conception de ces régulateurs. La solution apportée est l'introduction de la notion d'extraction automatique des connaissances sans besoin relatif de l'expert [39].

Les méthodes d'extraction automatique des connaissances sont basées sur l'apprentissage pour déduire l'information suffisante pour traiter un problème quelconque.

Dans notre travail on applique l'approche connexionniste, cette approche consiste à combiner la théorie des réseaux de neurones avec celle de la logique floue afin de concevoir ce qu'on appelle un réseau neuronal flou (FNN, fuzzy neural network).

Dans cette approche, l'architecture du réseau dépend du type de règles et des méthodes d'inférence, d'agrégation et de défuzzification choisies [39].

Dans ce cas nous nous intéressons à la structure ANFIS (adaptive neural fuzzy based inference system). Cette architecture exploite complètement le modèle flou de Sugeno connu sous le nom TSK (TSK fuzzy model), proposé par Takagi, Sugeno et Kang.

Pour simplifier, nous considérons un système flou ayant deux entrées et une sortie, un régulateur flou du premier ordre ayant deux règles peuvent représentés comme suit:

$$R^1 : \text{SI } (x_1 \text{ est } A_1) \text{ ET } (x_2 \text{ est } B_1) \text{ ALORS } (F_1 = a_0^1 + a_1^1 x_1 + a_2^1 x_2) \quad (3.17)$$

$$R^2 : \text{SI } (x_1 \text{ est } A_2) \text{ ET } (x_2 \text{ est } B_2) \text{ ALORS } (F_2 = a_0^2 + a_1^2 x_1 + a_2^2 x_2)$$

L'architecture du réseau de neurone correspondant est montrée par la figure (3.11) [39], [38].

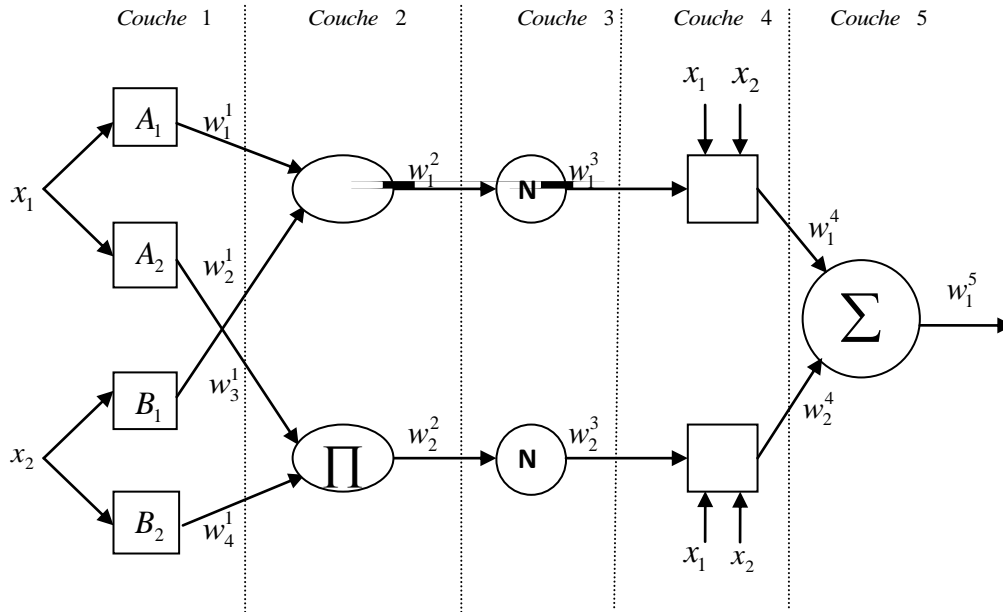


Fig.3.11: Réseau de neurone basé sur le modèle flou de Sugeno.

Couche 1: chaque neurone de cette couche réalise l'opération de fuzzification donnée par:

$$W_i^1 = \mu_{A_i}(x_1) \quad \text{Pour } i=1,2 \quad (3.18)$$

$$W_i^1 = \mu_{B_{i-2}}(x_2) \quad \text{Pour } i=3,4 \quad (3.19)$$

Où x_1, x_2 sont les entrées du nœud i et A_i et B_{i-2} sont les ensembles flous.

$\mu_{A_i}(x_1)$ et $\mu_{B_{i-2}}(x_2)$ sont les fonctions d'appartenance de la prémisse qui peuvent être choisies comme des paramètres ajustables.

Couche 2: couche d'inférence : chaque nœud calcule le degré d'activation ou de vérité de chaque règle, appliquant une conjonction Min ou Prod.

$$W_i^2 = \mu_{A_i}(x_1) \mu_{B_i}(x_2) ; \quad i=1,2 \quad (3.20)$$

Couche 3: couche de normalisation, la sortie de chaque neurone est donnée par:

$$W_i^3 = \frac{W_i^2}{W_1^2 + W_2^2} \quad i=1,2 \quad (3.21)$$

couche 4: un neurone de cette couche est appelé un neurone adaptatif, sa sortie est:

$$W_i^4 = W_i^3(a_0^i + a_1^i x_1 + a_2^i x_2); \quad i=1,2 \quad (3.22)$$

Où (a_0^i, a_1^i, a_2^i) sont des paramètres des conséquences.

Couche 5: un seul neurone dans cette dernière couche réalise l'opération de défuzzification. Le nœud de cette couche réalise une fonction fixe qui effectue la somme des signaux entrants. La sortie de ce nœud est donnée par la relation suivante:

$$W_i^5 = \sum_{i=1}^2 W_i^4 = \frac{\sum_{i=1}^2 W_i^3 (a_0^i + a_1^i x_1 + a_2^i x_2)}{\sum_{i=1}^2 W_i^3} \quad (3.23)$$

3.3.3.2. Algorithme d'apprentissage

La détermination des paramètres d'un régulateur flou de Sugeno constitue la phase la plus délicate dans sa conception, compte tenu du nombre important des paramètres à déterminer (paramètres des prémisses et des conséquences).

Pour estimer les paramètres des réseaux neuronaux, on applique un algorithme d'apprentissage basé sur le filtre de Kalman étendu.

Soit un régulateur flou de Sugeno caractérisé par un ensemble de données entrée-sortie $(x(k), d(k))$ et un vecteur de paramètres θ regroupant tous les paramètres des prémisses ainsi des conséquences. Ces paramètres sont ajustables afin d'approcher la sortie du régulateur flou le mieux que possible à la sortie désirée $d(k)$. L'application du filtre de Kalman étendu consiste à linéariser à tout instant la sortie du régulateur autour du vecteur $\hat{\theta}$ estimé. Ceci revient à écrire:

$$\begin{aligned} d(k) &= \Delta u(x(k); \hat{\theta}(k-1)) + \psi^T(k)(\theta - \hat{\theta}(k-1)) \\ \psi(k) &= \frac{\partial \Delta u(x(k); \theta)}{\partial \theta} / \hat{\theta}(k-1) \end{aligned} \quad (3.24)$$

La forme de la solution est :

$$\begin{aligned} \hat{\theta}(k) &= \hat{\theta}(k-1) + p(k)\psi(k)e(k) \\ e(k) &= d(k) - \Delta u(x(k); \hat{\theta}(k-1)) \end{aligned} \quad (3.25)$$

$$\text{Où} \quad p(k) = \frac{\alpha_1 I}{\alpha_2 + \psi^T(k)\psi(k)} \quad \alpha_1 > 0, \alpha_2 > 0 \quad (3.26)$$

$$\psi = \frac{\partial u}{\partial \theta} \quad \text{Avec } \theta \text{ est le vecteur des paramètres de prémisse et de conséquence}$$

Pour notre cas, les entrées du réseau sont l'erreur et sa variation, celles-ci sont caractérisées par trois ensembles flous de type gaussienne : NG, ZE et PG, définis par l'expression suivantes:

$$\mu(x) = \exp \left\{ -0.5(v_i(x - c_i))^2 \right\} \text{Où } x_i \in \{e, \Delta e\} \quad (3.27)$$

Où c_i sont la moyenne et v_i l'inverse de la variance.

Les règles floues combinant l'erreur et sa variation nous donne la sortie du régulateur, sont groupées dans le tableau (3.3):

e Δe	NG	ZE	PG
NG	f_1	f_4	f_7
ZE	f_2	f_5	f_8
PG	f_3	f_6	f_9

Tableau.3.3: Règles d'inférence floue.

Pour la partie conséquence, les entrées sont les mêmes que celle de la partie prémisse.

Le régulateur flou de Sugeno du premier ordre ayant neuf règles qui prennent la forme suivante :

$$\text{Si } (e \text{ est } F_1^i) \text{ et } (\Delta e \text{ est } F_2^i) \text{ alors } f_i = a_0^i + a_1^i e + a_2^i \Delta e \quad \text{avec } i=1 : 9$$

La sortie du régulateur flou de Sugeno est donnée par la relation suivante :

$$Y = \sum_{K=1}^9 \mu_K y_K / \sum_{K=1}^9 \mu_K \quad \text{avec } \mu_l = \prod_{i=1}^n \mu_{F_i^i}(x_i) ; 1 \leq l \leq m$$

Où les paramètres $\theta = (c, v, a_0^i, a_1^i, a_2^i)$ sont des paramètres ajustables.

La figure (3.15) montre une représentation schématique, sous forme d'un réseau, d'un système flou TS1 à deux entrées et deux ensembles flous pour chaque variable.

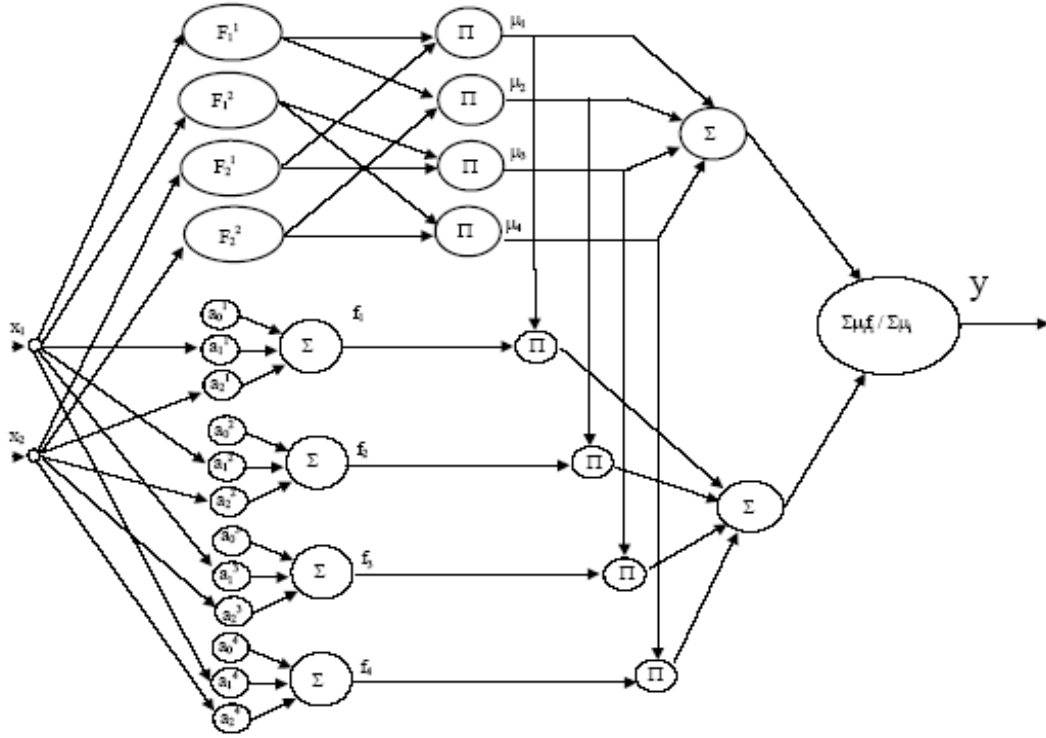


Fig.3.12: Représentation d'un système flou de Takagi-Sugeno d'ordre 1.

Dans notre travail, l'erreur est la différence entre la sortie du régulateur flou de Mamdani qui est caractérisé par sept ensembles flous de type triangulaire pour chacune des variables linguistiques d'entrée et de sortie, et la sortie du régulateur flou de Sugeno avec trois ensembles flous de type gaussienne pour les variables linguistiques de l'erreur et de sa variation.

Pour l'apprentissage, la méthode appliquée est le calcul du gradient $\psi = \frac{\partial \Delta u}{\partial \theta}$

Avec $\theta = [c \quad v \quad a_0 \quad a_1 \quad a_2]$ est le vecteur des paramètres des prémisses et des conséquences. Par conséquent nous avons :

$$\frac{\partial \Delta u}{\partial \theta} = \left[\frac{\partial \Delta u}{\partial c} \quad \frac{\partial \Delta u}{\partial v} \quad \frac{\partial \Delta u}{\partial a_0} \quad \frac{\partial \Delta u}{\partial a_1} \quad \frac{\partial \Delta u}{\partial a_2} \right] \quad (3.28)$$

Où

$$\frac{\partial \Delta u}{\partial c_i} = \frac{v_i^2 (x_i - c_i) \sum_{k \in J} \mu_k (f_k - \Delta u)}{\sum_{l=1}^M \mu_l} \quad (3.29)$$

$$\frac{\partial \Delta u}{\partial v_i} = - \frac{v_i (x_i - c_i)^2 \sum_{k \in J} \mu_k (f_k - \Delta u)}{\sum_{l=1}^M \mu_l} \quad (3.30)$$

$$\frac{\partial \Delta u}{\partial a_0} = \frac{\mu_i}{\sum_{l=1}^M \mu_l} ; \quad \frac{\partial \Delta u}{\partial a_1} = \frac{\mu_i e}{\sum_{l=1}^M \mu_l} ; \quad \frac{\partial \Delta u}{\partial a_2} = \frac{\mu_i \Delta e}{\sum_{l=1}^M \mu_l} \quad (3.31)$$

Les paramètres initiaux des prémisses et des conséquences pour les courants statoriques suivants les axes d et q sont regroupés dans les tableaux suivants :

Pour le courant iqs

	f1	f2	f3	f4	f5	f6	f7	f8	f9
a_0	-0.6019	-0.4118	-0.6548	-0.2756	0.1489	0.2639	-0.4294	1.2080	0.9341
a_1	0.6680	-0.3421	0.1488	0.4383	0.5679	0.5592	-0.4909	1.3121	0.6647
b	0.3836	0.5560	-0.0022	0.6839	1.0311	0.8758	0.5868	0.8827	-0.0721

Tableau.3.4.a: Valeurs initiales des conséquences.

	Erreur			Variation de l'erreur		
	NG	ZE	PG	NG	ZE	PG
c	-0.7911	-0.0919	0.9971	-0.3813	0.5294	0.7000
v	1.8031	2.1167	1.7492	2.5896	2.2894	0.0000

Tableau.3.4.b: Valeurs initiales des prémisses.

Pour le courant ids

	f1	F2	f3	f4	f5	f6	f7	f8	f9
a_0	-0.2884	-0.1826	-0.2582	-0.1663	0.1763	0.1605	-0.1601	1.2253	0.7806
a_1	0.4798	0.4492	0.6789	0.6339	0.8664	0.9554	-0.3586	1.0063	0.1950
B	0.3470	0.7282	0.7147	0.7741	1.2278	0.9886	0.9423	0.8692	-0.3093

Tableau.3.5.a: Valeurs initiales des conséquences.

	Erreur			Variation de l'erreur		
	NG	ZE	PG	NG	ZE	PG
c	-0.6676	0.0153	1.3160	-0.6371	0.6197	0.7000
v	1.6507	2.1173	1.7074	1.8879	2.0029	0.0000

Tablea.3.5.b: Valeurs initiales des prémisses.

Les résultats d'apprentissage sont les suivants :

- Pour le régulateur de vitesse

	f1	f2	f3	f4	f5	f6	f7	f8	f9
a_0	- 0.59774	- 0.40852	- 0.65068	- 0.26879	0.14865	0.26387	- 0.42971	1.20561	- 0.93621
a_1	0.67033	- 0.33943	0.15366	0.44387	0.56977	0.56634	- 0.49965	1.30963	0.65989
b	0.38795	0.55655	- 0.00250	0.69971	1.03060	0.88847	0.58814	0.88216	- 0.07615

Tableau.3.6.a: Résultats d'apprentissage des paramètres des conséquences

Pour le régulateur de vitesse.

	Erreur			Variation de l'erreur		
	NG	ZE	PG	NG	ZE	PG
c	-0.792569	-0.086815	1.003847	-0.374479	0.512763	0.700000
v	1.802806	2.106354	1.744851	2.596804	2.292571	0.000000

Tableau.3.6.b: Résultats d'apprentissage des paramètres des prémisses

Pour le régulateur de vitesse.

- Pour le régulateur de flux

	f1	f2	f3	f4	f5	f6	f7	f8	f9
a_0	- 0.28854	- 0.18249	- 0.25733	- 0.16972	0.17725	0.16163	- 0.15863	1.22673	0.778221
a_1	0.48044	0.44979	0.68062	0.63341	0.86693	0.95626	- 0.35746	1.00810	0.19231
b	0.34761	0.72784	0.71457	0.77369	1.22699	0.98698	0.93660	0.87005	-0.30691

Tableau.3.7.a: Résultats d'apprentissage des paramètres des conséquences Pour le régulateur du flux.

	Erreur			Variation de l'erreur		
	NG	ZE	PG	NG	ZE	PG
c	-0.665338	0.013904	1.313998	-0.638577	0.616821	0.700000
v	1.648059	2.118184	1.706100	1.880863	2.004643	0.000000

Tableau.3.7.b: Résultats d'apprentissage des paramètres des prémisses

Pour le régulateur du flux.

3.3.3.3. Résultats de simulation

Afin de réduire la taille du régulateur, nous avons proposé un régulateur de type Sugeno construit uniquement de trois fonctions d'appartenance et un ensemble de neuf règles servant à la commande avec l'élimination de l'étape de défuzzification. Ce régulateur est appliqué au réglage de la vitesse d'une machine asynchrone.

Les résultats obtenus montre une bonne performance du réglage. En effet, la vitesse suit sa référence au bout de 0.2s, le courant statorique est bien limité durant le régime transitoire.

Le flux rotorique est maintenu, il suit sa valeur de référence suivant l'axe (d) avec une composante en quadrature (q) nulle.

La comparaison, entre ces résultats et ceux obtenus par l'utilisation des régulateurs flous de type Mamdani, relève que les réponses dynamiques de la machine sont identiques ce qui montre l'efficacité de l'algorithme d'apprentissage utilisé.

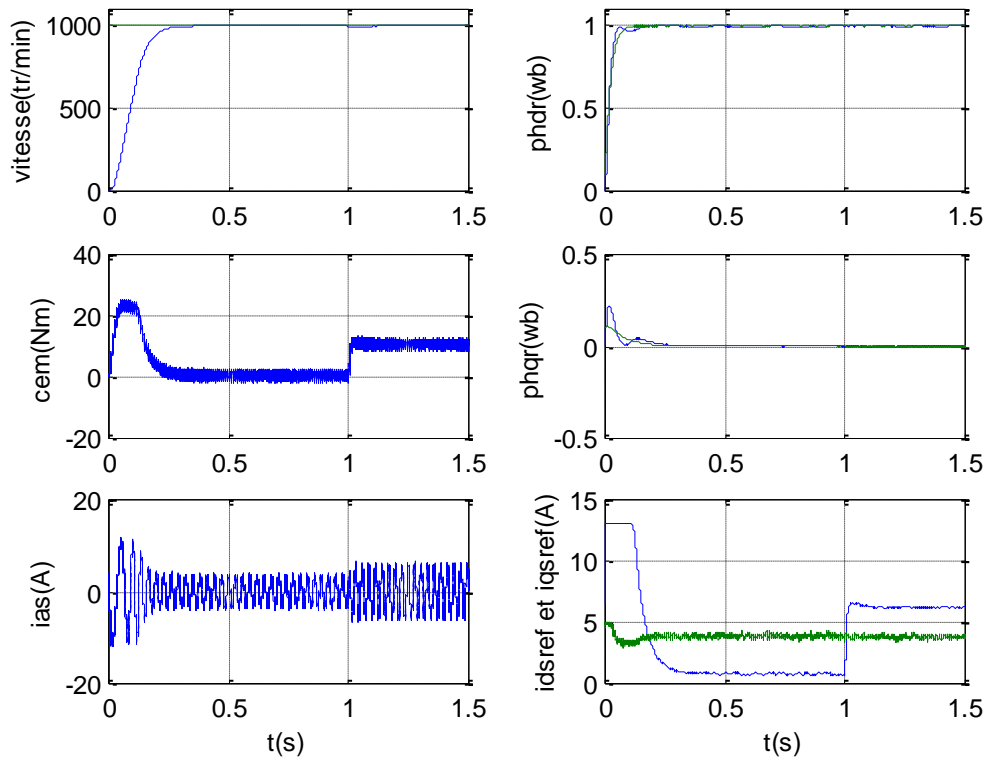


Fig.3.13.a: Comportement dynamique de la MAS lors d'un démarrage avec variation de la charge

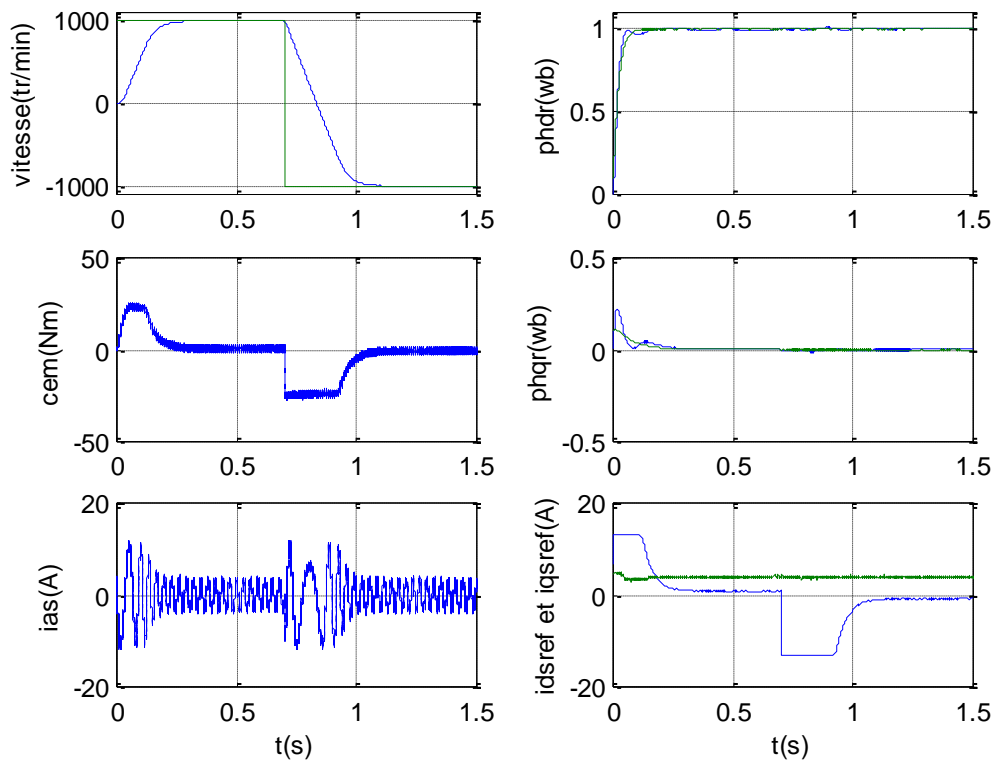


Fig.3.13.b : Comportement dynamique de la MAS lors d'une inversion de vitesse .

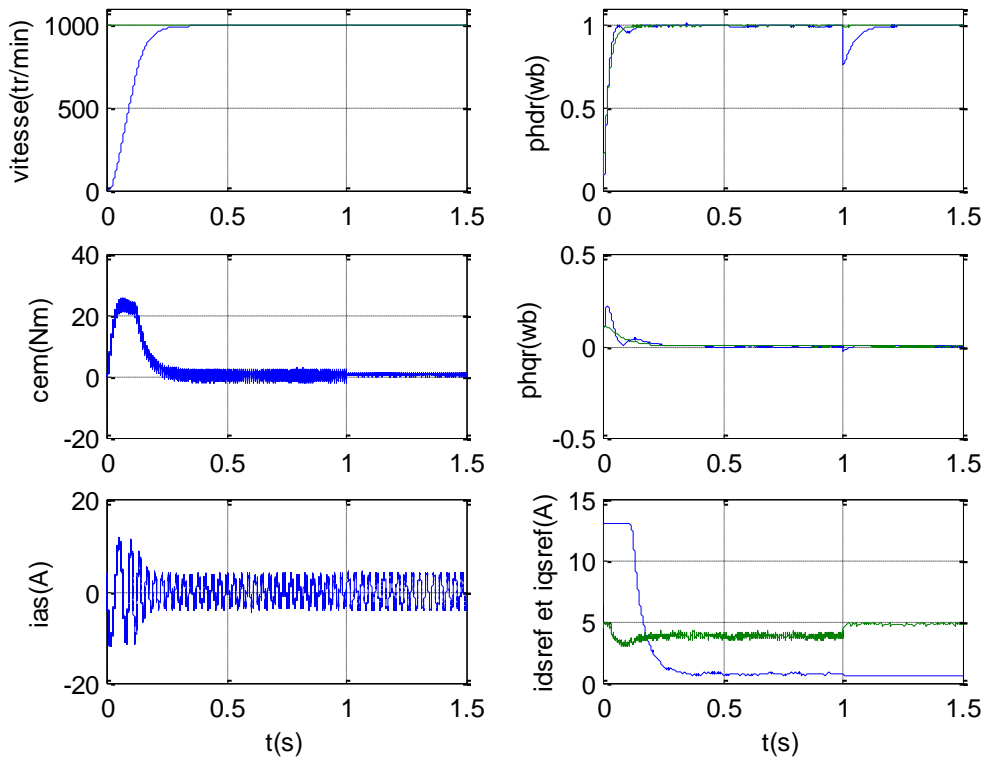


Fig.3.13.c: Comportement dynamique de la MAS lors d'un démarrage avec variations paramétriques et couple résistant

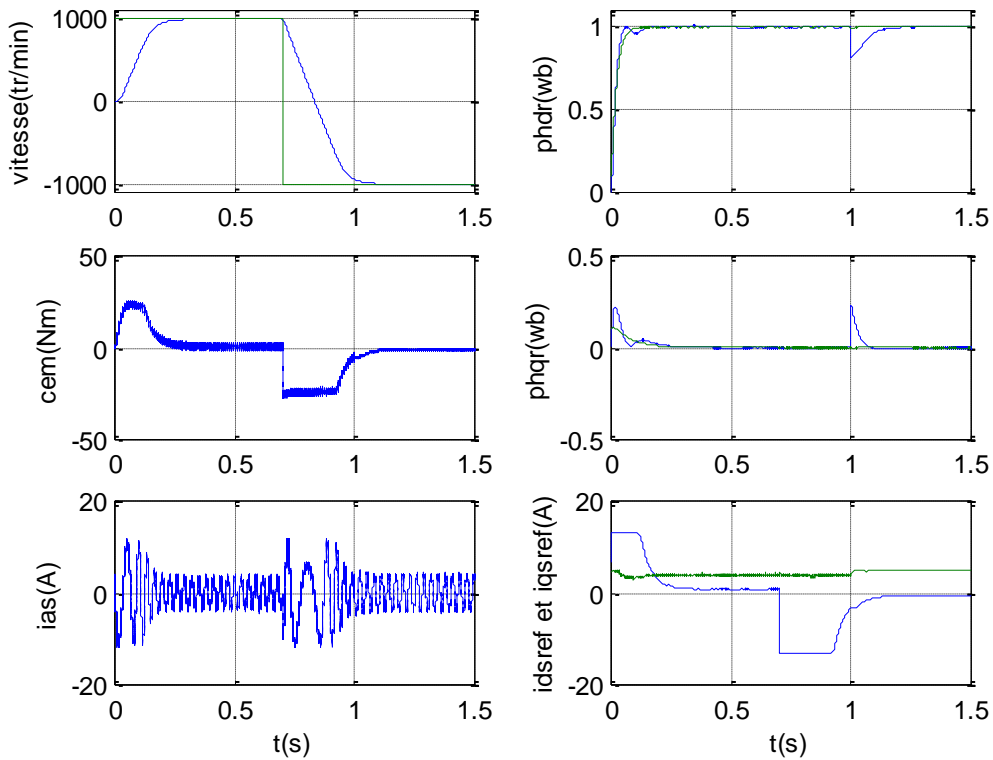


Fig.3.13.d: Comportement dynamique de la MAS lors d'un démarrage avec variations paramétriques et inversion de vitesse

3.3.4. Avantages et inconvénients de la commande par logique floue :

Certainement, la commande par logique floue comporte un certain nombre d'avantages et d'inconvénients. Les avantages essentiels sont :

- La non-nécessité d'une modélisation du processus à commander, ainsi que d'une analyse mathématique profonde,
- La possibilité de bénéficier et d'implémenter des connaissances et des expertises humaines sur le système à commander,
- La maîtrise de systèmes à régler avec un comportement complexe (fortement non-linéaire et difficile à modéliser), aussi, cette approche offre la possibilité d'utiliser des processeur spécialisé (dit processeurs flous), afin d'augmenter la précision ainsi la vitesse de calcul,

Par contre, les inconvénients sont :

- Le manque de méthodes systématiques précises pour la conception et la synthèse d'une commande par logique floue (par exemple dans le choix des grandeurs à mesurer, dans la détermination des facteurs de normalisation, dans le choix de la stratégie, soit de la fuzzification, les inférences soit de la défuzzification,...),
- L'impossibilité de la démonstration de la stabilité du circuit de commande en toute généralité,
- La cohérence des inférences non garantie a priori (apparition de règles d'inférence contradictoires possible),
- La possibilité d'apparition de cycles limites à cause du fonctionnement non linéaire.

3.4. CONCLUSION

Dans ce chapitre, nous avons représenté des régulateurs flous ainsi que ses délégués constituants. On a étudié deux types de régulateurs flous à savoir : le régulateur flou de Sugeno et celui de Mamdani, le premier utilise des valeurs numériques pour les conclusions alors que le deuxième utilise des sous-ensembles flous pour les conclusions. Nous avons aussi exposé les différents avantages qu'offre la logique floue dans le traitement d'informations incertaines ou imprécises.

