

IMPLÉMENTATION D'UN RÉGULATEUR PI FOU ADAPTATIF SUR UN AUTOMATE PROGRAMMABLE INDUSTRIEL S7-300 POUR LE CONTRÔLE DE LA VITESSE D'UNE MACHINE ASYNCHRONE

FAROUK BETTACHE¹, MOUNIR BOUHEDDA², AMEL ABBADI³

Mot clés : Automate programmable industriel ; Machine asynchrone ; Régulateur proportionnel-intégral ; Adaptatif-flou proportionnel-intégral ; Logique floue ; Variateur de fréquence.

Cet article présente la conception et l'implémentation d'un régulateur proportionnel-intégral (PI) adaptatif-flou dans un automate programmable industriel S7-300 de SIEMENS dans le but d'améliorer les performances du régulateur PID interne de l'API. Les entrées du régulateur PI adaptatif-flou sont l'erreur de réglage $e(t)$ et sa variation $\Delta e(t)$. Ils sont choisis pour représenter le plus fidèlement possible la dynamique du système. Les sorties du régulateur flou sont reliées au régulateur PID de l'API afin d'adapter ses gains K_p et T_i en utilisant les techniques de la théorie de la logique floue. Ce régulateur PI adaptatif-flou est utilisé pour contrôler en boucle fermée la vitesse d'une MAS par l'intermédiaire d'un variateur de fréquence VFD. En contrepartie, la commande scalaire est choisie au niveau du VFD pour réaliser cette régulation. Un test de validation est effectué afin de comparer les résultats du régulateur conçu, obtenus par le biais du simulateur PLCSIM avec ceux obtenus par le module flou conçu sous Matlab. Les résultats expérimentaux obtenus montrent la différence entre les performances des régulateurs PI et PI-flou. En utilisant le régulateur PI flou, le temps de réponse a été réduit à 50 % par rapport à celui obtenu par le régulateur PI classique.

1. INTRODUCTION

Du fait de sa simplicité de construction, d'utilisation et d'entretien, de sa robustesse et de son faible coût, la machine asynchrone est la plus utilisée dans les applications industrielles requérant la variation de la vitesse [1,2]. Quand la machine asynchrone est alimentée par un réseau à fréquence fixe, il est difficile de faire varier sa vitesse. En outre, au démarrage, le couple est faible et le courant appelé est très élevé. Grâce aux progrès de l'électronique de puissance et à travers l'équipement électrique appelé variateur de fréquence (Variable-Frequency Drive VFD), il est bien possible actuellement de faire démarrer cette machine convenablement et de la faire fonctionner avec une vitesse réglable dans une large plage. Ceci vient du fait que le VFD joue le rôle d'une interface et permet de convertir l'alimentation électrique de la machine (caractérisée par des valeurs fixes en fréquence et tension) en une alimentation électrique à fréquence et tension variables, ce qui permet la modification de la vitesse et du couple de la machine [3, 4]. Par conséquent, le VFD est largement utilisé dans diverse chaînes de productions automatique et systèmes de commande [5]. Ce dernier est souvent couplé d'une part, à un API pour automatiser le fonctionnement de la machine et/ou faire la régulation d'une grandeur physique telle que la vitesse ou le couple. D'autre part, il est couplé à un IHM (interface Homme Machine) afin de permettre à l'opérateur d'introduire et de superviser les paramètres ainsi que les données de la machine, par exemple introduire la vitesse de consigne et visualiser la vitesse de rotation de la machine en tr/min [6].

Les régulateurs PI conventionnels caractérisés par les coefficients k_p et k_i , ont été largement appliqués dans de nombreux problèmes de régulation industrielle.

La plupart des systèmes physiques présentent des non-linéarités et leurs paramètres sont souvent mal connus et/ou variables dans le temps [7]. Pour commander de telles classes de systèmes, les méthodes conventionnelles de contrôle ont certaines limites, les paramètres de contrôle ne peuvent pas être ajustés lorsque le système subit des

variations paramétriques, et le système ne peut pas offrir de bonnes performances lorsque les paramètres de contrôle sont maintenus constants durant l'opération de contrôle [8].

Les régulateurs basés sur les concepts de l'intelligence artificielle telles que les techniques de la logique floue et les réseaux de neurones, sont des alternatives intéressantes, et permettent d'obtenir des performances élevées en termes de robustesse [9, 11]. Aujourd'hui, le réglage par la logique floue avec sa structure non-linéaire a montré des bonnes performances en robustesse dans le contrôle de la machine asynchrone (MAS) [12]. Il s'agit d'une nouvelle technique traitant la commande numérique des processus et de prise de décision. Cette technique repose sur la théorie des ensembles flous qui permet de traiter et de raisonner à l'aide de variables et des quantifications linguistiques obtenu d'un opérateur humain expérimenté. Les régulateurs flous peuvent être considérés comme des PID non-linéaires et leurs paramètres sont déterminés en temps réel en se basant sur l'erreur et sa dérivée [13].

Dans ce travail, un régulateur flou auto-adaptatif a été conçu et implémenté dans un API (s7-300) pour contrôler la vitesse d'une machine asynchrone triphasée à travers le VFD. Les paramètres du régulateur PI sont ajustés en fonction des variables d'états du système et des exigences des performances de la régulation en boucle fermée. Dans notre cas le temps de réponse est considéré comme critère de comparaison.

2. RÉGULATEUR PI ADAPTATIF-FLOU

Le régulateur PI-flou peut agir sur les paramètres du régulateur PI (k_p , k_i) d'une manière intelligente en se basant sur la théorie de la logique floue, ce qui rend le régulateur PI adaptable au fonctionnement du système non-linéaire [9]. Le schéma de principe de l'utilisation du régulateur flou est illustré par Fig. 1. Les gains k_p et k_i du régulateur PI-flou sont calculés à partir de l'erreur et de sa variation ($e(t)$ et $\Delta e(t)$). Ces gains sont déterminés en temps réel pour forcer le système à suivre la référence spécifiée et s'adaptent en continu pour corriger au fur et à mesure l'évolution du système en agissant sur la loi de régulation

¹Université de Médéa, Laboratoire de Mécanique, Physique et Modélisation Mathématique, Médéa, Algérie, E-mail : bettache_farouk@yahoo.fr

²Université de Médéa, Laboratoire des Systèmes Électroniques Avancés, Médéa, Algérie, E-mail : bouhedda.m@gmail.com

³Université de Médéa, Laboratoire de Recherche en Électrotechnique et Automatique, Médéa, Algérie, E-mail : amel.abbadi@yahoo.fr

[9]. L'erreur instantanée est utilisée pour agir sur les paramètres du régulateur au prochain instant d'échantillonnage.

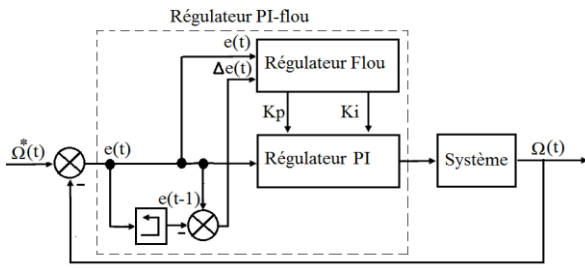


Fig. 1 – Principe d'adaptation du régulateur PI par la logique floue.

Lors du fonctionnement du régulateur en temps réel, une matrice floue permet d'adapter les gains de façon à optimiser les caractéristiques de la réponse temporelle. Une base de règles sert à établir une unique table de décision en fonction des valeurs $e(t)$ et $\Delta e(t)$. Les changements sur les deux paramètres du PI sont calculés en multipliant un élément de la table par les paramètres du régulateur qui sont déterminés par une méthode analytique. L'adaptation des paramètres de régulateur PI se fait en utilisant

$$\begin{cases} k_p = k_{p0} + \Delta k_p \\ k_i = k_{i0} + \Delta k_i \end{cases} \quad (1)$$

3. STRUCTURE DU RÉGULATEUR PI-FOU

Figure 2 illustre la structure de contrôle à logique floue, elle comprend trois composants principaux, qui sont la fuzzification, la base de règles et la défuzzification.

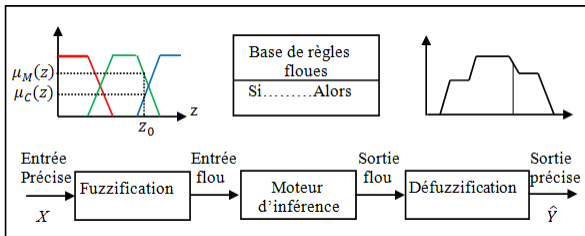


Fig. 2 – Structure de contrôle à logique floue.

3.1. FUZZIFICATION

L'étape de fuzzification consiste à déterminer le degré d'appartenance d'une valeur pour une variable d'entrée et de sortie à un ensemble flou [10].

Les fonctions d'appartenance de l'erreur $e(t)$ et de sa variation $\Delta e(t)$ sont choisies identiques de formes triangulaires et sont définies sur l'intervalle $[-3, 3]$ comme le montre Fig. 3a.

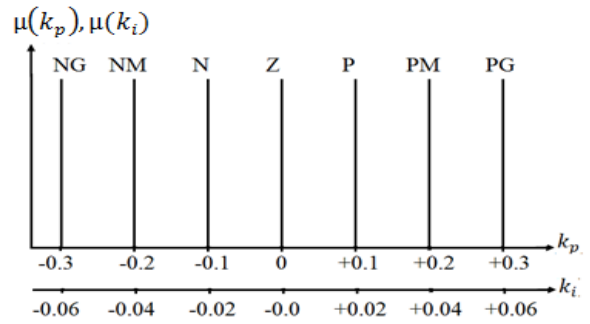
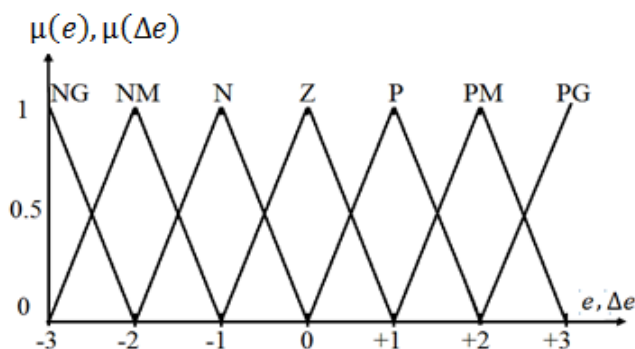


Fig. 3 – Fonctions d'appartenance des entrées ($e, \Delta e(t)$) et de sorties (k_p, k_i)

$$\mu_A = \begin{cases} 0 & \text{pour } x < a \\ \frac{x-a}{b-a} & \text{pour } a \leq x \leq b \\ \frac{c-x}{c-b} & \text{pour } b \leq x \leq c \\ 0 & \text{pour } x > c \end{cases} \quad (2)$$

Les fonctions d'appartenance des gains (k_p, k_i) sont choisies de formes singleton sur l'intervalle $[-0.3, 0.3]$ et $[-0.06, 0.06]$ respectivement (voir Fig. 3b). Les sous-ensembles flous associés à ces variables sont désignées par les termes linguistiques négatif grande (NG), négatif moyen (NM), négatif (N), zéro (Z), positif (PS), moyen positif (MP), positif grand (PG). Le degré d'appartenance issu d'une fonction triangulaire est donné par l'expression mathématique (2) où la forme est donnée par Fig. 4.

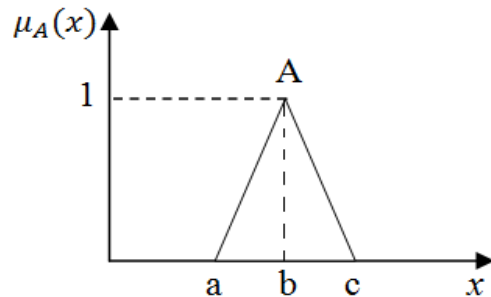


Fig. 4 – Fonction d'appartenance triangulaire.

3.2. BASE DES RÈGLES

Les règles floues sont extraites des connaissances fondamentales et l'expérience humaine sur le processus [10]. Ces règles se présentent sous la forme de :

« Si $e(t)$ est A_i ET $\Delta e(t)$ est B_i Alors k_p est C_i et k_i est D_i ».

où : A_i, B_i, C_i et D_i : sont les sous-ensembles flous des variables d'entrées et de sorties et ET opérateur de base.

Pour choisir les règles, il suffit d'observer la réponse du système en boucle fermée pour un changement de la valeur de la charge ou de la consigne.

Les paramètres du régulateur PI sont choisis de façon à obtenir une réponse plus rapide, avec un dépassement réduit et un bon amortissement des oscillations pendant un temps court. Quand l'erreur est grande, k_p devrait être grand et k_i petit et quand l'erreur est petite, k_p devrait être petit et k_i grand. La raison est que k_p fournit une réponse dynamique rapide et k_i élimine l'erreur statique et assure la stabilité du système [1].

A partir de ces règles on peut arriver à établir les tables d'inférence floue représentées sur les tableaux (1) et (2).

Tableau 1
Règles d'inférence du paramètre k_p

e Δe	NG	NM	N	Z	P	PM	PG
NG	PG	PG	PM	PM	P	Z	Z
NM	PG	PG	PM	P	P	Z	N
N	PM	PM	PM	P	Z	N	N
Z	PM	PM	P	Z	N	NM	NM
P	P	P	Z	N	N	NM	NM
PM	N	Z	N	NM	NM	NM	NG
PG	Z	Z	NM	NM	NM	NG	NG

Tableau 2
Règles d'inférence du paramètre k_i

e Δe	NG	NM	N	Z	P	PM	PG
NG	NG	NG	NM	NM	Z	Z	Z
NM	NG	NG	N	N	N	Z	Z
N	NG	NM	N	N	Z	P	Z
Z	NM	NM	N	Z	P	PM	PM
P	NM	N	Z	P	P	PM	PM
PM	Z	Z	P	P	PM	PG	PG
PG	Z	Z	P	PM	PM	PG	PG

La méthode d'inférence floue de type Sugeno d'ordre zéro a été appliquée pour l'évaluation de ces règles. Ce choix est justifié par sa facilité et sa simplicité dans l'étape de défuzzification Singleton [2]. Figure 5 explicite l'évaluation des règles selon la méthode Sugeno.

Le résultat de l'implication $\mu_c(z_i)$ est donné par l'opérateur **min** qui est défini par la relation :

$$\mu_c(z_i) = \min [\mu_{A_i}(x) \text{ et } \mu_{B_i}(y)]. \quad (3)$$

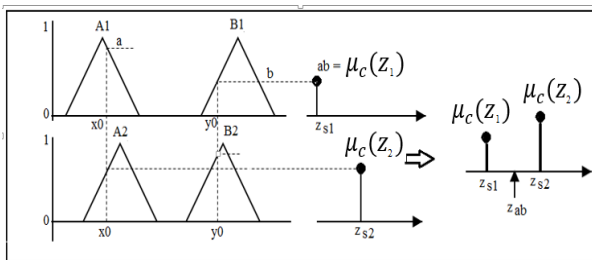


Fig. 5 – Base de règles selon la méthode Sugeno.

3.3 DÉFUZZIFICATION

La dernière étape consiste à transformer la valeur de sortie floue en valeur réelle en appliquant la méthode de défuzzification. La méthode utilisée est celle du centre de gravité; elle revient à faire une somme pondérée des centres des ensembles flous inférés multipliés par les degrés d'appartenance correspondant. Figure 6 donne un exemple ou la solution précise est donné par l'expression (4).

$$z_0 = \frac{\sum_i \mu_i(z_i) \cdot z_i}{\sum_i \mu_i(z_i)} = \frac{\mu_c(z_1) \cdot z_1 + \mu_c(z_2) \cdot z_2 + \dots + \mu_c(z_n) \cdot z_n}{\mu_c(z_1) + \mu_c(z_2) + \dots + \mu_c(z_n)} \quad (4)$$

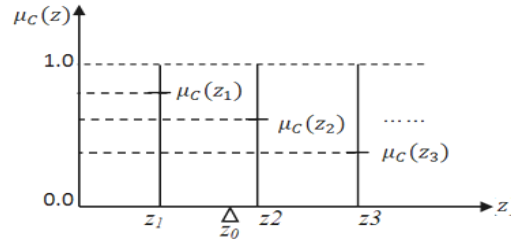


Fig. 6 – Solution floue de type Sugeno d'ordre zéro.

4. MISE EN ŒUVRE DU RÉGULATEUR PI-FLOU AUTO-ADAPTATIF

Le schéma fonctionnel proposé pour l'implémentation du régulateur PI flou auto-adaptatif est illustré par la Fig. 7. L'API utilisé représente la partie commande qui donne les consignes et reçoit les informations issues de la partie opérative, et effectue également des échanges avec l'extérieur via une interface homme-machine (IHM). La partie opérative est constituée par un pré-actionneur représenté par un variateur de fréquence VFD qui agit sur la machine asynchrone triphasée.

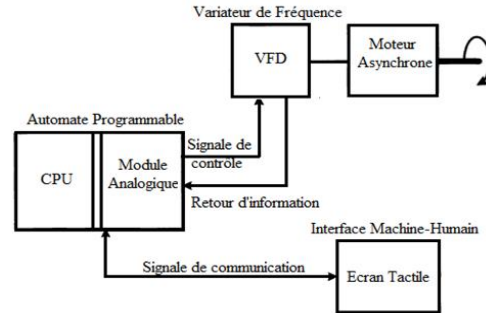


Fig. 7 – Schéma synoptique de la mise en œuvre du régulateur PI-flou auto-adaptatif.

4.1. PARTIE OPÉRATIVE

Cette partie est constituée de trois éléments essentiels : Le variateur de vitesse, l'IHM et la machine asynchrone utilisée.

4.1.1. Variateur de fréquence (ATV12H018M2)

Ce dispositif permet de varier la vitesse de la machine en faisant varier sa fréquence d'alimentation. Il est mis en connectivité à travers les entrées/sorties suivantes :

- **R/L1 - S/L2/N** : Entrées d'alimentation monophasé du variateur.
- **U/T1-V/T2-W/T3** : Sorties d'alimentation
- **LI1- LI2 - LI3** : Entrées logiques programmable (24 V).
- **II** : Entrée analogique en tension (0-10v).
- **AO1** : Sortie analogique en tension.

Le VFD est connecté à l'API à travers l'entrée AI1 et la sortie AO1 analogiques. Ces derniers représentent l'entrée de point de référence et la de fréquence mesurée sous forme de tensions continues comprises entre 0 et 10 V. La fréquence d'alimentation de la machine est proportionnelle à la tension présente aux bornes AI1 et AO1. De même, les entrées logiques LI1, LI2 et LI3 du VFD sont reliés à l'API. Elles représentent les entrées de la mise en marche en avant, la mise en marche en arrière et l'arrêt d'urgence de la machine. D'autre part, les bornes (U, V et W) du VFD sont connectées aux machines asynchrones qui lui délivrent une alimentation variable en tension et en fréquence avec un rapport V/F = cte (commande scalaire). Figure 8 montre le schéma de câblage du variateur de

fréquence (ATV12H018M2) avec l'automate programmable (S7-300) et avec la machine asynchrone.

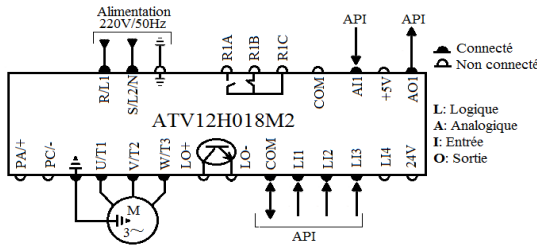


Fig. 8 – Le schéma de câblage du VFD (ATV12H018M2).

4.1.2. Interface homme machine

Une IHM reliée à l'API via le port RS-232, est utilisée pour imposer la vitesse de référence et visualiser la vitesse courante de la machine en tr/min, ainsi que la mise en marche et l'arrêt de la machine via des touches tactiles.

Figure 9 montre l'interface conçue.

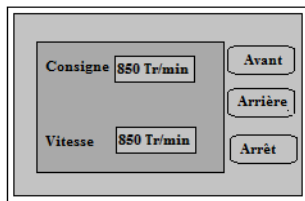


Fig. 9 – L'IHM conçue.

4.1.3. Machine asynchrone

La machine à induction triphasé utilisée est caractérisée par : quatre pôles 50 Hz, 0,75 kW, 1 430 tr/mn, 380 V, 2,1 A. Celui-ci est régulé via le variateur de fréquence VFD. La vitesse de la machine varie linéairement en fonction de la fréquence (voir Fig. 10).

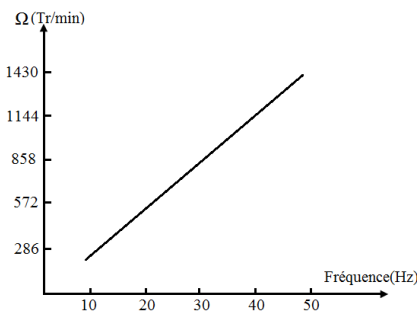


Fig. 10 – Variation de la vitesse en fonction de la fréquence.

4.2. PARTIE COMMAND

Un programme permettant le contrôle de la vitesse de la machine par un régulateur PI-flou adaptatif en boucle fermée a été implémenté sur un API S7-300 de SIEMENS. Ce dernier est caractérisé par deux interfaces digitales et analogiques intégrés, et un port de communication série RS232. Tous ces interfaces ont été utilisés pour recevoir et transmettre différents types d'informations (analogiques, digitales et numériques) de et vers la partie opérative.

A cet effet, trois fonctions prédéfinies du langage Ladder à savoir : Scale, Unscale et Norm ont été utilisées dans le programme final pour adapter et mettre en échelle les données reçus et envoyés de et vers l'API (voir Fig. 11). En contrepartie, ces données sont exploitées par le régulateur PI-flou auto-adaptatif pour faire la régulation de la vitesse de la machine.

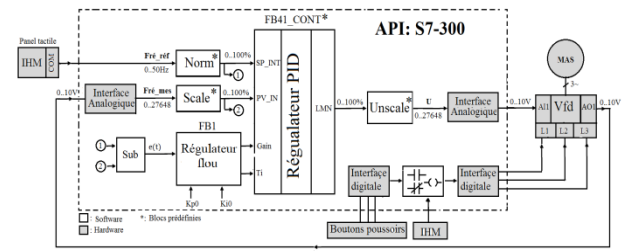
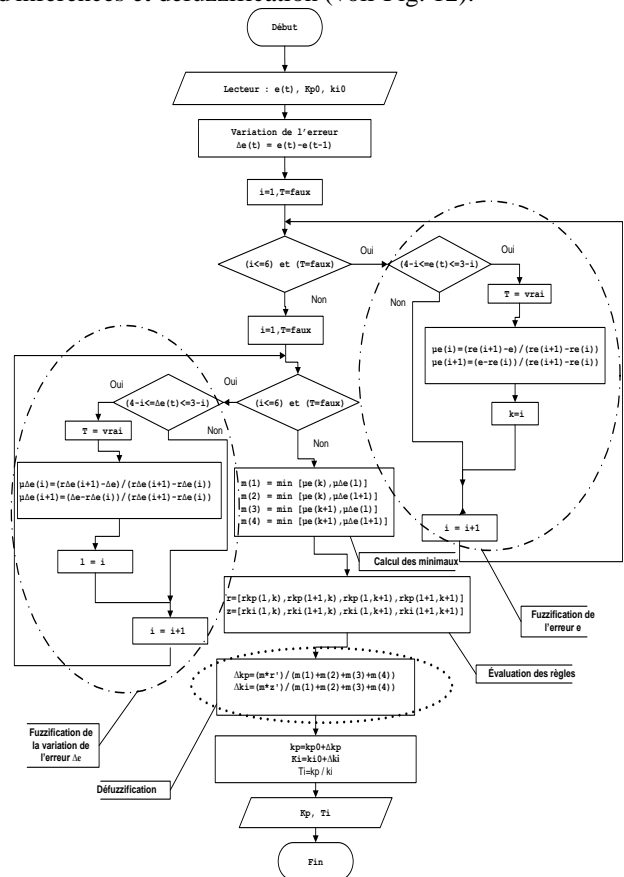


Fig. 11 – Schéma synoptique de l'implémentation software et hardware.

Ce régulateur est constitué par un régulateur PI représenté par le bloc FB41_CONT prédéfini en langage Ladder, et un régulateur flou représenté par le bloc fonctionnel FB1 conçu à l'aide du langage SCL (Structure Control Laguage). Les entrées SP_INT, PV_IN et la sortie LMN du bloc FB41 ont été utilisés pour introduire la consigne et la fréquence mesurée et pour générer la commande ; tandis que les entrées Gain et Ti ont été utilisées pour définir les paramètres k_p et T_i du régulateur PI. Le bloc FB1 est caractérisé par trois entrées à savoir : k_{p0} , k_{i0} pour définir les valeurs initiales des gains et l'erreur $e(t)$ afin d'introduire l'erreur précalculée entre la consigne et la fréquence mesurée. FB1 délivre deux sorties K_p , T_i dans le but d'adapter les paramètres du régulateur PI. À chaque période d'échantillonnage, le bloc FB1 opère les fonctions de fuzzification, évaluation des règles d'inférences et défuzzification (voir Fig. 12).



Où μ_e et $\mu_{\Delta e}$: degrés d'appartenance de l'erreur et la variation de l'erreur
 $re = [-3, -2, -1, 0, 1, 2, 3]$: Abscisses de l'univers des discours de l'erreur
 $r\Delta e = [-3, -2, -1, 0, 1, 2, 3]$: Abscisses de l'univers des discours de la variation de l'erreur
 $rkp(l, k)$: Valeur Singleton d'indice (l, k) du tableau (2)
 $rki(k, l)$: Valeur Singleton d'indices (l, k) du tableau (3)
 $m = [m(1) m(2) m(3) m(4)]$: Vecteur des minimaux.

Fig. 12 – Organigramme du régulateur PI-flou auto adaptative.

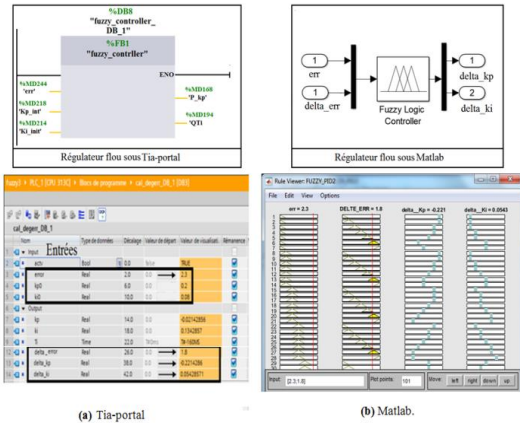


Fig. 13 – Comparaison des résultats obtenus des régulateur flous sous TIA-PORTAL (a) et sous Matlab (b) pour $(e(t), \Delta e(t)) = (2,3, 1,8)$.

Avant son implémentation expérimentale, le régulateur flou (FB1) a fait l'objet d'une validation par comparaison de ses sorties $(\Delta k_p, \Delta k_i)$ pour n'importe quelles valeurs des entrées $(e(t), \Delta e(t))$, avec ceux obtenus par le module "Fuzzy Inference System (FIS)" prédéfini de Matlab en utilisant exactement les mêmes paramètres flous (Univers du discours, méthode de fuzzification, règles, ...).

A titre d'exemple la paire d'entrée $(e, \Delta e) = (2,3, 1,8)$, produit de part et d'autre les mêmes sorties numériques suivants $(\Delta k_p, \Delta k_i) = (-0,221, 0,0543)$ comme il est montré dans Fig. 13.

5. RÉSULTATS EXPÉRIMENTAUX ET INTERPRÉTATIONS

Figure 14 montre l'installation expérimentale établie par les composants.



Fig. 14 – Installation expérimentale.

5.1. EXPÉRIENCE 1: RÉGULATEUR PI CLASSIQUE

L'objectif de cette expérience, est de montrer les performances du régulateur PI classique envers les variations de consigne sans l'utilisation du régulateur flou conçu dans la boucle de régulation de la vitesse. À cet effet, les valeurs $k_p = 0,269$ et $k_i = 0,068$ ($T_i = 3,955$ s) ont été choisis de façon empirique, comme gains à ce régulateur. Les performances obtenues sont illustrées par Fig. 15.

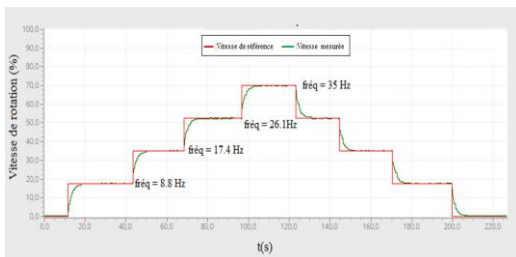


Fig. 15 – Réponses indicielles du régulateur PI pour $k_p = 0,269$ et $k_i = 0,068$.

5.2. EXPÉRIENCE 2 : RÉGULATEUR PI-FLOU

Pour cette expérience, les mêmes valeurs des paramètres (k_p, k_i) précédents sont utilisés comme valeurs initiales à l'entrée du régulateur flou réinséré dans la boucle de régulation de la vitesse (Fig. 11). Les performances obtenues sont illustrées par Fig. 16.

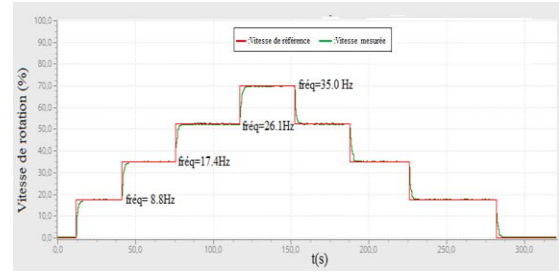


Fig. 16 – Réponses indicielles du régulateur PI-flou pour $k_{p0} = 0,269, k_{i0} = 0,068$.

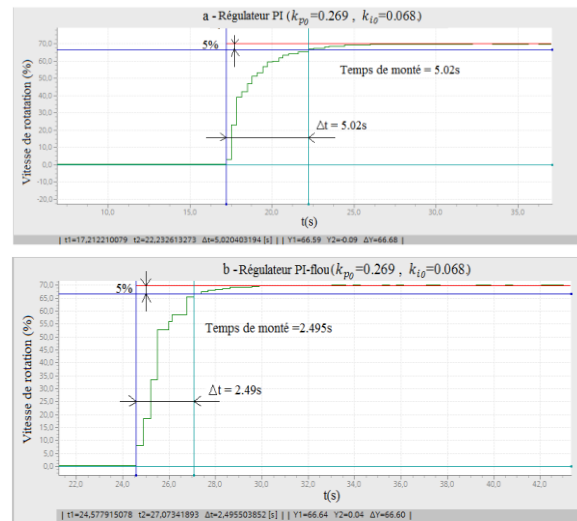


Fig. 17 – Comparaison des temps de réponse du régulateur PI classique (a) et du régulateur PI-flou (b).

6. INTERPRÉTATION DES RÉSULTATS

Que ce soit pour le correcteur PI classique ou PI-flou, deux remarques peuvent être déclinées :

- Un excellent tenu des régulateurs face aux variations des consignes que ce soit à la montée ou à la descente ce qui est clairement montré par les Figs. 15 et 16.
- Très petites oscillations de l'ordre $\pm 0,5$ % autour de point de référence.

Le régulateur PI-flou à l'avantage par rapport au régulateur PI classique de réagir plus rapidement au changement de consigne où, pour les mêmes paramètres k_{p0} et k_{i0} , le temps de réponse a été réduit à la moitié comme dans Fig. 17. Les valeurs des temps de réponse obtenus sont comme suite :

- Régulateur PI-flou : $\Delta t = 2,49$ s.
- Régulateur PI classique : $\Delta t = 5,02$ s.

7. CONCLUSION

L'objet du travail présenté dans cet article est d'apporter des améliorations aux performances (temps de réponse, dépassement, etc) a la réponse de la machine asynchrone commande par PI classique implémenter sur un automate programmable industrielle (S7-300).

A cet effet, on a procédé à la mise en œuvre d'un régulateur flou afin d'adapter les gains (proportionnel et intégral) du régulateur PID interne de l'API (S7-300).

Ce dernier est relié à un variateur de fréquence pour commander la vitesse de la machine asynchrone selon la loi de commande scalaire.

Le régulateur flou réalisé est constitué par des sous-ensembles flous de forme triangulaire pour les entrées $e(t)$ et $\Delta e(t)$, des sous-ensembles flous en singleton pour les sorties k_p et k_i et une défuzzification de type Sugeno.

Un test de validation du régulateur flou conçu à l'aide du langage SCL a été effectué, avant son implémentation sur l'automate, par comparaison des sorties résultantes de celui-ci avec celles obtenus par le module Fuzzy Inference System (FIS) de Matlab en utilisant les mêmes paramètres et les mêmes entrées.

Les résultats obtenus ont montré des performances très satisfaisantes, se traduisant par des comportements excellents, aussi bien en poursuite, qu'en régulation, avec une réponse rapide de la machine lors de la variation des consignes. D'après les résultats expérimentaux, le régulateur PI adaptatif-flou a montré de bonnes performances vis-à-vis le temps de réponse du système à contrôler.

REMERCIEMENTS

Les auteurs remercie vivement la DGRSDT-MESRS-Algérie pour leur soutien.

Reçue le 30 décembre 2020

RÉFÉRENCES

1. R. Marino, P. Tomei, CM. Verrelli, *Induction Motor Control Design*, Springer, 2010, pp. 229 – 286.

2. N. Saad, M. Arrofiq, *PLC-based Fuzzy Logic Controller for Induction-motor Drive with Constant V/Hz Ratio*, Robotics, and Computer-Integrated Manufacturing, **28**, 1, pp. 95 – 112 (2012)
3. S. Ganesh M.J. Rao Malla, *Direct torque control of induction motor with fuzzy controller, a review*, International Journal of Engineering Trend in Electrical and electronics, **10**, 3, pp. 2320 – 9569 (2014).
4. E.B. Agamloh, *Power and Efficiency Measurement of Motor-Variable Frequency Drive Systems*, IEEE Transactions on Industry Applications, **53**, 1, pp. 766 – 773 (2017)
5. C.L. Su, W.L. Chung, K.T. Yu, *An energy-savings evaluation method for variable-frequency-drive applications on ship central cooling systems*, IEEE Transactions on Industry Applications, **50**, 2, pp. 1286–1294 (2014).
6. V. Ganesan, A. Unruh, A. Rudolph, *Simulation of variable frequency drive systems in oil gas applications*, Petroleum and Chemical Industry Conference Europe (PCIC Europe), Antwerp, pp. 1–8 (2018).
7. L. Reznik, O. Ghanayem, A. Bourmistrov, *PID plus fuzzy controller structures as a design base for industrial applications*, Engineering Applications of Artificial Intelligence, **13**, 4, pp. 419 – 430(2000).
8. J. Xing, Z. Liu, Y. Yang, Z. Zheng, *Fuzzy self-adaptive PI control method for permanent magnet synchronous motor drive*, International Journal of Simulation -- Systems, Science & Technology, **17**, 17, pp. 2.5–2.8 (2016).
9. A.N. Seghir, T. Henni, M. Azira, *Fuzzy and adaptive fuzzy PI controller-based Vector control for permanent magnet synchronous motor*, 10th IEEE International Conference On Networking, Sensing and Control (ICNSC), pp. 491 – 496 (2013).
10. A. Derbane, B Tabbache, A. Ahrich, *A fuzzy logic approach based direct torque control and five-leg voltage source inverter for electric vehicle powertrains*, Rev. Roum. Sci. Techn.– Électrotechn. et Énerg., **66**, 1, pp. 15–20 (2021).
11. H.M. Shi, Z.J. Yu, *Design of PMSM Position servo system based on fuzzy adaptive PI control*, Advanced Materials Research, **706 - 708**, pp. 737-741 (2013).
12. A. Asri, Y. Mihoub, S. Hassaine, P. O. Logerais, A. Amiar, T. Allaoui, *An adaptive fuzzy proportional integral method for maximum power point tracking control of permanent magnet synchronous generator wind energy conversion system*, Rev. Roum. Sci. Techn.–Électrotechn. et Énerg., **63**, 3, pp. 320– 325 (2018).
13. M. El Azzaoui, H. Mahmoudi, *Fuzzy-PI control of a doubly fed induction generator-based wind power system*, International Journal of Automation and Control, **11**, 1, pp. 54–66 (2017).

IMPLEMENTATION OF AN ADAPTIVE FUZZY PI REGULATOR ON AN INDUSTRIAL PROGRAMMABLE PLC S7-300 FOR THE SPEED CONTROL OF AN ASYNCHRONOUS MACHINE

Keywords: Programmable logic controller; Asynchronous machine; Proportional–integral controller; Adaptive fuzzy proportional-integral; Fuzzy logic; Variable-frequency drive.

This article presents the design and implementation of an adaptive-fuzzy PI regulator in a SIEMENS S7-300 Industrial Programmable Logic Controller to improve the performance of the internal PID regulator of the PLC. The inputs of the adaptive-fuzzy PI regulator are the adjustment error $e(t)$ and its variation $\Delta e(t)$. They are chosen to represent the system's dynamics as faithfully as possible. The outputs of the fuzzy regulator are connected to the PID regulator of the PLC to adapt its gains K_p and T_i using the techniques of fuzzy logic theory. This adaptive-fuzzy PI regulator is used for closed-loop control of the speed of a three-phase asynchronous machine through a VFD frequency converter.

On the other hand, scalar control is chosen at the level of the VFD to achieve this regulation. A validation test is carried out to compare the results of the designed regulator, obtained using the PLCSIM simulator, with those obtained by the fuzzy module designed in MATLAB. The experimental results show the difference between the performance of PI and PI-fuzzy regulators. By using the fuzzy PI regulator, the response time was reduced to 50% compared to that obtained by the classic PI regulator.