

Commande neuro-flou inverse des systèmes non linéaires
Dr. Fatima Zohra Daikh Dr. amadouche mohamed amine
Université de mascara

RÉSUMÉ :

Dans cet article, ont traite l'exploitation des propriétés de l'Intelligence Artificielle dans le domaine d'automatique, notre travail est consacré a l'utilisation des réseaux neuro-flou et précisément ANFIS (Adaptive Neuro-Fuzzy Inference System) pour l'identification des modèle inverse nécessaires a la réalisation des lois de commandes d'un système dynamique non-linéaire. Le modèle est utilisé dans une première approche, comme contrôleur en boucle ouverte avec le système, (Direct inverse control) pour un objectif de régulation. Dans la deuxième approche le modèle inverse est utilisé conjointement avec un contrôleur PID classique dans une structure de commande hybride pour améliorer ses performances. Il contribue au rejet des perturbations. La troisième approche la commande par modèle interne (Internal model control) pour connaître la robustesse du modèle neuro-flou, même que le système est soumis à une constante perturbation. La dernière section présente une application de ces structures sur un système non-linéaire. Les travaux sont validés par des simulations sous l'environnement MATLAB.

Mots clés: Réseaux neuro-flou, ANFIS, Apprentissage, Commande neuro-flou, Système non-linéaire, Intelligence artificielle

ABSTRACT:

In this article, have dealt with the exploitation of the properties of Artificial Intelligence in the field of automatic, our work is dedicated to the use of neuro-fuzzy networks and specifically ANFIS (Adaptive Neuro-Fuzzy Inference System) for the identification of the inverse model necessary for the realization of the control laws of a nonlinear dynamic system. The model is used in a first approach, as an open loop controller with the system, (Direct inverse control) for a control purpose. In the second approach the inverse model is used in conjunction with a conventional PID controller in a hybrid control structure to improve its performance. It contributes to the rejection of disturbances. The third approaches the internal model control to know the robustness of the neuro-fuzzy model, even if the system is subjected to a constant perturbation. The last section presents an application of these structures on a non-linear system. The work is validated by simulations under the MATLAB environment.

Key words: Neuro-fuzzy networks, ANFIS, Learning, Neuro-fuzzy control, Non-linear system, Artificial intelligence

I INTRODUCTION

Les principales difficultés dans la théorie de la commande des systèmes dynamiques réels sont les non-linéarités et les incertitudes. Or la commande passe par l'élaboration d'un modèle mathématique du système en trouvant une relation entre les entrées et les sorties, ce qui suppose une bonne connaissance de la dynamique du système et ses propriétés. Dans le cas des systèmes non-linéaires, les techniques conventionnelles ont montré souvent leur insuffisance surtout quand les systèmes à étudier présentent de fortes non-linéarités. Le manque de connaissances a priori nécessaire pour l'élaboration du modèle mathématique était en quelque sorte dans cet échec.

Face à ce problème, le recours aux méthodes de commandes par apprentissage est devenu une nécessité car les système de commande obtenus ainsi procèdent par collecter des données empiriques, stocker et extraire les connaissance contenues dans celle-ci et utiliser ces connaissances pour réagir à de nouvelles situations : on est passé à la commande intelligente (intelligent control) [11].

Parmi les techniques d'apprentissage, les réseaux neuro-flous, où la force du raisonnement global et la capacité d'adaptation de la logique floue et la puissance des capacités d'apprentissage et de généralisation des réseaux de neurones sont combiné. Diverses combinaisons de ces deux méthodes (logique flou, réseaux neurones) ont été développées depuis 1988. Elles ont donné naissance aux systèmes neuro-flou qui sont le plus souvent orientées vers la commande de système complexes et des systèmes multi-variables.

Plusieurs chercheurs ont essayé d'exploiter les avantages des réseaux neuro-flou pour commander un système dynamique, et plus précisément, dans le domaine de la robotique [17] [19] et pour la commande d'un moteur asynchrone [2] [21]. On peut trouver plus de détails sur les structures de commandes avec modèle neuro-flou dans [13] [16].

Le présent travail est une mise en exergue des capacités des réseaux neuro-flou à reproduire la dynamique inverse des systèmes non-linéaires et l'utilisation du modèle neuro-flou (ANFIS) inverse ainsi reproduit dans la conception des structures de commande neuro-flou d'un système non- linéaire.

La première section est consacrée à l'élaboration du modèle neuro-flou (ANFIS) inverse et la présentation des architectures utilisées pour l'apprentissage de ce modèle. Ensuite, en section 2, trois structures de commande sont étayées : la première basée sur l'utilisation du modèle inverse comme contrôleur (Direct inverse control), la seconde, le modèle inverse intervient à côté d'un contrôleur PID classique dans une structure de commande hybride et la troisième, le modèle inverse avec la modèle interne (modèle direct) dans une commande par le modèle interne (IMC) .

La dernière section présente une application de ces structures sur un système non-linéaire les résultats sont validées par des simulations MATLAB

II. LES RÉSEAUX NEURO-FLOUS

Les réseaux neuronaux et la logique floue présentent séparément, des avantages et des inconvénients. Le point de départ des travaux sur la fusion de ces deux approches, le neuro-flou, fut la volonté de réduire leurs inconvénients en alliant les caractéristiques sémantiques du flou et les capacités d'apprentissage des systèmes neuronaux. Un système basé sur des règles linguistiques, le connexionnisme a apporté des méthodes systématiques pour l'extraction des connaissances et l'optimisation des fonctions d'appartenance, alors que le flou a conduit à une diminution du temps d'apprentissage, par la possibilité d'introduire de la connaissance à priori dans les poids synaptiques.

Il s'agit souvent d'une présentation neuronal du flou dans laquelle le neuronal joue le rôle d'hardware pour implémenter un système d'inférence flou, avec la connaissance exprimée à partir des règles (comme pour le flou) et un mécanisme d'apprentissage (comme pour le neuronal).

II.1. Structure d'un réseau neuro-flou

La structure d'un réseau neuro-flou est inspirée de la similitude d'organisation entre un système d'inférence flou et un réseau de neurones multicouches. Cette analogie peut se résumer comme suit:

- Le raisonnement utilisé dans un système d'inférence flou peut être effectué en trois étapes (fuzzification, inférence des règles et défuzzification), quel que soit le nombre de règles utilisées (caractéristique du parallélisme massif)
- Ce raisonnement présente une structure que l'on peut qualifier de pré neuronale, i.e. une structure en couches où chaque couche est interprétée comme une partie du système d'inférence. Les paramètres du contrôleur correspondent aux poids du réseau, tandis que les règles sont constituées à partir du réseau.

Il existe quatre grandes catégories de combinaisons des réseaux de neurones avec la Logique floue : réseau flou neuronal, système neuronal/flou simultanément, modèles neuro-flous coopératifs et modèles neuro-flous hybrides, les approches neuro-flous modernes sont de cette forme. Un réseau neuronal et un système flou sont combinés dans une architecture homogène. Le système peut être interprété comme un réseau neuronal spécial avec des paramètres flous ou comme un système flou mise en application sous une forme distribuée parallèle.

II.2. ANFIS et ses variantes

II.2.1. Adaptive Neuro- Fuzzy Inference System (ANFIS)

La méthode ANFIS est une technique d'optimisation des systèmes d'inférence flous de type Takagi Sugeno propose par Jang[6]. Celle-ci utilise pour ajuster les paramètres du système, la méthode des moindres carrés combinée à la méthode de descente de gradient. Cette méthode est basée sur l'utilisation de réseaux multicouches. On assume que le système d'inférence flou a deux entrées x et y et une seule sortie Z . Supposons que la règle de base contient deux règles floues "si-alors" de type Takagi Sugeno :

Règle i : si x est A_i et y est B_i alors $f_i = p_i x + q_i y + r_i$

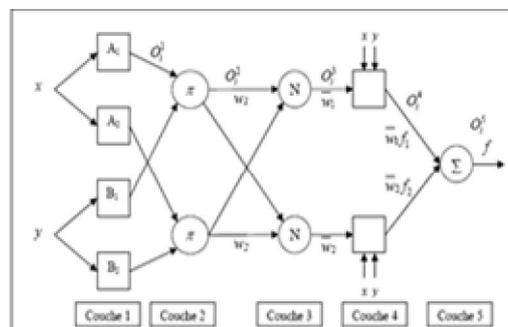


Fig.1: Architecture d'ANFIS

Couche 1 :

Chaque nœud i dans cette couche est un nœud carré avec une fonction d'appartenance

$$O_i^1 = U_{A_i}(x) \quad (1)$$

Avec x : L'entée du nœud i , A_i : Label linguistique associé au nœud de fonction, en d'autre terme est la fonction d'appartenance de A_i et il spécifie le degré d'appartenance avec lequel x satisfait. est choisi de la forme :

-Forme de cloche

$$U_{A_i}(x) = \frac{1}{1 + \left[\left(\frac{x - c_i}{\alpha_i} \right)^2 \right]^b} \quad (2)$$

Où $\left[\left(\frac{x - c_i}{\alpha_i} \right)^2 \right]^b$ Forme gaussienne

$$U_{A_i}(x) = \exp \left[- \left(\frac{x - c_i}{\alpha_i} \right)^2 \right] \quad (3)$$

Où sont des paramètres qui font référence aux paramètres prémisses. Les valeurs de ces derniers changes selon diverse exhibition de la fonction d'appartenance.

Couche 2 :

Les sorties de cette couche sont les poids des règles, ils sont obtenus par une simple multiplication des entrés dans chaque cellule.

$$w_i = U_{A_i}(x) \times U_{B_i}(x) \quad (4)$$

Couche 3 :

Cette couche correspond à la normalisation des poids des règles. Elle calcule le ratio entre le i Poids de la règle et la somme de tout les poids des règles.

$$\bar{w} = \frac{w_i}{w_1 + w_2}, i = 1, 2 \quad (5)$$

Couche 4 :

Chaque nœud i dans cette couche est un noeud qui se calcule de la façon suivante:

$$O_i^4 = \bar{w}_i f_i = \bar{w}_i (p_i x + q_i y + r_i) \quad (6)$$

sont les sorties de la couche 3.

sont les paramètres appelés paramètres conséquents de la fonction de sortie.

Couche 5 :

Dans cette couche la cellule fait la somme de tous les signaux d'entrés et retourne donc, en sortie, la valeur approximative de la fonction désirée [6].

$$O_i^5 = \sum_i \bar{w}_i f_i \quad (7)$$

III. LE MODELE INVERSE

Bien que le modèle inverse de système joue un rôle important dans la théorie de la commande, l'accomplissement de sa forme analytique est assez laborieuse. Plusieurs méthodes de modélisation des systèmes ont été présentées dans la littérature [9]. Un système dynamique peut être décrit par l'équation (8) reliant ces entrées aux sorties :

$$y(k+1) = f(y(k), \dots, y(k-n+1), u(k), \dots, u(k-m+1)) \tag{8}$$

où la sortie du système $y(k+1)$ dépend des n valeurs précédentes de la sortie et les m valeurs passées de l'entrée. En général, le modèle inverse de ce système peut être présenté sous la forme suivante (9):

$$u(k) = f^{-1}(y(k+1), y(k), \dots, y(k-n+1), u(k-1), \dots, u(k-m)) \tag{9}$$

Les réseaux neuro-flou peuvent être utilisés pour élaborer le modèle inverse du système. Cependant, d'autres types de réseaux ont été utilisés [18], toutefois la représentation de l'aspect dynamique du système reste une problématique. L'application des retards vers la couche d'entrée à ce type de réseau peut présenter la solution pour palier ce manque.

Le réseau correspondant à (9) est :

$$\hat{u}(k) = g(x(k), w) . \tag{10}$$

La fonction g va être approximée par un ANFIS en modifiant ses poids w . Le réseau a un vecteur d'entrée $x(k)$ composé de la sortie $k+1$ et des valeurs des sorties et entrées passées et ayant pour sortie le signal \hat{u} qui va servir à commander le système (Fig. 2)

L'identification du modèle inverse commence par la détermination du vecteur d'entrée, à savoir le nombre des retards en sorties et entrées, ceci est lié à l'ordre du système.

A. Architecture Générale d'Apprentissage

Dans la première architecture (Fig.3), le signal u est appliqué à l'entrée de système, produisant une sortie y qui est fournie au réseau. La différence entre le signal d'entrée u et la sortie, l'erreur rétro propagée à travers le réseau et qui va servir pour l'apprentissage hors ligne du réseau [10].

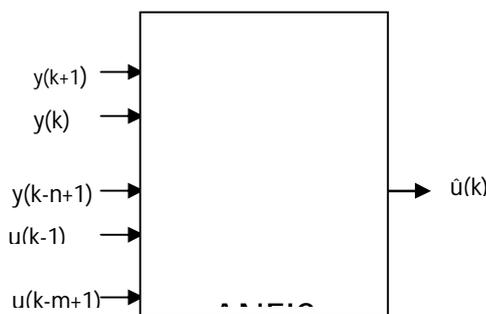


Fig. 2. Réseau ANFIS pour identifier le modèle inverse

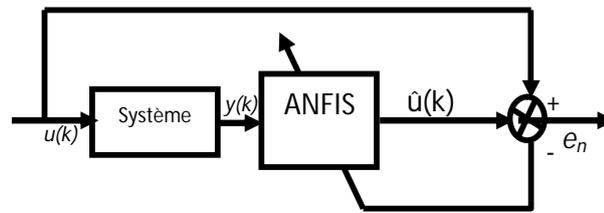


Fig. 3. Architecture générale d'apprentissage

Cette méthode présente plusieurs inconvénients [8] [20] :

Les sorties y du système utilisées dans l'apprentissage ne garantissent pas que les sorties du modèle neuronal vont être dans des régions voulues pour le succès de son utilisation dans la commande.

Si le système à commander est multi- variable, le modèle ainsi retenu peut ne pas imiter le système réel.

B. Architecture Indirecte d'Apprentissage

Cette approche est une mise en œuvre particulière de l'architecture précédente dans laquelle, le modèle inverse en cours d'apprentissage sert aussi à commander le système. La consigne r est fournie au premier réseau qui produit une commande \hat{u} au système, la sortie de celui-ci est passée comme consigne à la seconde copie du modèle inverse, qui produit alors une commande \hat{u} . La différence entre u et \hat{u} sert de signal d'erreur afin d'effectuer l'apprentissage des paramètres du modèle. Cette architecture, proposée par [12] est présentée sur la (Fig.4).

L'idée derrière cette architecture est que la minimisation de l'erreur commise sur la commande entraînera une minimisation de l'erreur en sortie du système. Cependant, une erreur nulle sur la commande ne provoque pas nécessairement l'annulation de l'erreur totale en sortie du système

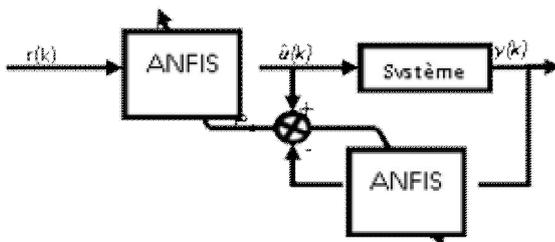


Fig. 4. Architecture indirecte d'apprentissage

IV.COMMANDE NEURO-FLOU

Parmi les structures de commande neuro-flou utilisant le modèle inverse, trois sont présentées dans les paragraphes qui suivent :

A. Commande neuro-flou directe par modèle inverse :

Comme son nom l'indique, le modèle neuro-flou inverse placé en avant du système, est utilisé comme contrôleur pour commander le système en boucle ouverte (Fig.5).

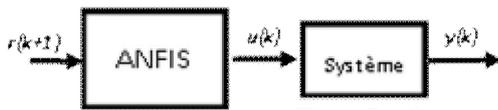


Fig. 5. Commande directe par modèle inverse

La valeur de $y(k+1)$ dans (9) est substituée par la sortie désirée $r(k+1)$ et on alimente le réseau par les valeurs retardées de $u(k)$ et $y(k)$. Si le modèle ANFIS est exactement l'inverse du système alors il conduit la sortie à suivre la consigne.

B. Commande neuro-flou hybride

La deuxième approche de commande est celle qui fait intervenir le réseau ANFIS au sein d'une structure de commande à laquelle participe aussi un contrôleur classique existant. Elle est présentée dans [4] dans le cadre de la manipulation d'un bras de robot. Elles consistent globalement à faire fonctionner simultanément un contrôleur feedback conventionnel et un modèle connexionniste. Cette approche ne vise plus à apprendre la dynamique inverse du système, mais à effectuer une régulation consistante de celui-ci. On fournit au régulateur feedback classique ainsi qu'au réseau ANFIS l'erreur e_c commise en sortie du système. Le réseau dispose de plus de la consigne $r(k)$, le signal de commande total est constitué de la somme des sorties des deux dispositifs. L'apprentissage ANFIS est réalisé par rétropropagation de la sortie du régulateur classique. Cette structure est représentée sur Fig.6

L'approche est justifiée par le fait que lorsque la régulation opérée est consistante, la sortie du régulateur classique est nulle, confiant ainsi la tâche de régulation au réseau neuro-flou (ANFIS). Il s'agit d'une optimisation du fonctionnement du modèle ANFIS par d'autres techniques. Citons à ce titre. [1], où un régulateur PID est utilisé pour réaliser la commande neuro-flou d'un procédé non linéaire simple. Jorgensen et al [7] utilisent cette méthode pour réaliser la commande d'un modèle d'avion en phase d'atterrissage.

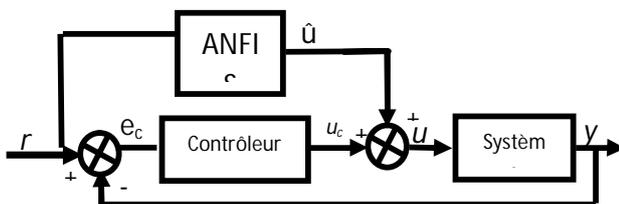


Fig. 6. Structure de commande neuro-flou hybride

L'apprentissage de la dynamique d'un bras articulé par une méthode de ce type est évoqué dans [5]. Pour la commande d'une colonne à distiller, Steck et al [15] proposent de copier un régulateur existant.

En ce qui concerne la coopération des modèles feedback classiques avec des modèles neuro-flou, Gomi et al [4] procèdent à des tests sur le problème classique du pendule inversé. Enfin, Psaltis et al [14] présentent un autre exemple de coopération dans laquelle le modèle neuro-flou utilise des informations issues d'un PID lors de son initialisation.

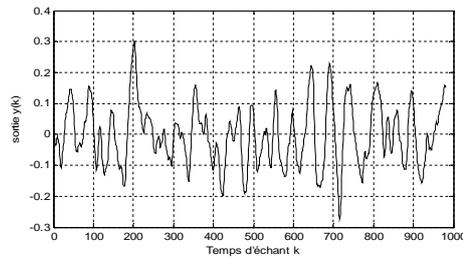


Fig. 9. Signal Sortie y(k)

Après plusieurs essais, réseaux ANFIS est comme entrée $\{y(k), y(k-1), y(k-2)\}$ avec deux fonction d'appartenance (type cloche) et 800 itération.

L'erreur entre les deux signaux entrée de système $u(k)$ et sortie d'ANFIS inverse $\hat{u}(k)$ est présentée dans Fig.10 avec un coût de

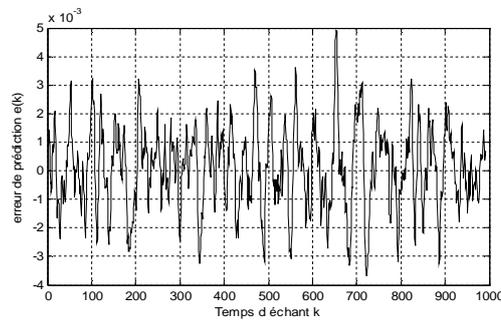


Fig. 10. Erreur de prédiction du modèle inverse

Pour l'application du modèle inverse pour la commande directe (§ 4.1) on a testé le modèle pour deux consignes, l'une

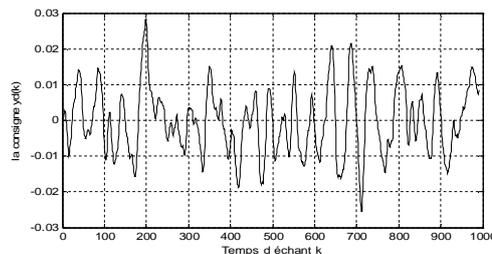


Fig. 11. Consigne yd

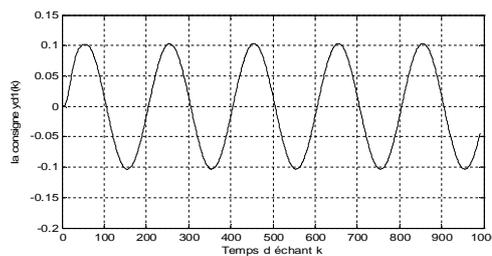


Fig. 12. Consigne yd1

On présente l’erreur entre la consigne y_d et la sortie du système commandé par modèle inverse dans (Fig.13) et celle entre la consigne y_{d1} et la sortie du système (Fig.14)

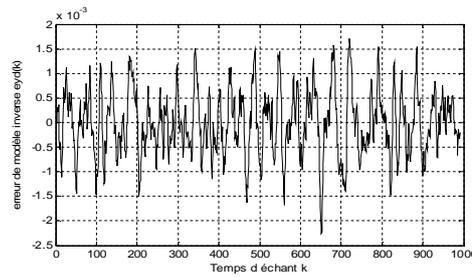


Fig. 13. Erreur commise lors de la commande directe (Consigne y_d)

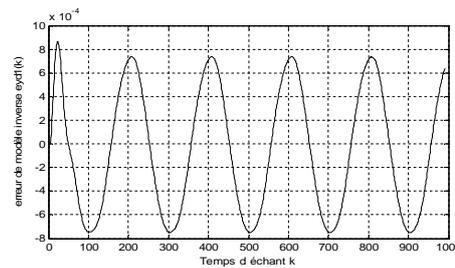
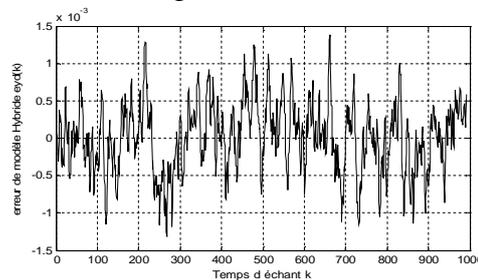


Fig. 14. Erreur commise lors de la commande directe (Consigne y_{d1})

Quant à la commande hybride directe (§ 4.2) on a obtenu des résultats satisfaisants dont l’erreur entre la consigne y_d et la sortie y est présentée en Fig.15 et celle entre la consigne y_{d1} et la



sortie réelle est en Fig.16

Fig. 15. Erreur commise lors de la commande hybride (Consigne y_d)

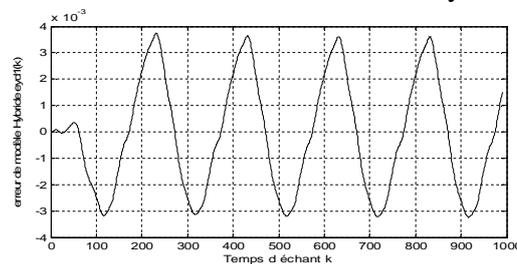


Fig. 16. Erreur commise lors de la commande hybride (Consigne y_{d1})

En dernier, on introduit un bruit (Fig.17) dans le système et on applique la commande hybride utilisée dans la simulation précédente. Les erreurs pour les consignes y_d et y_{d1} sont en Fig.18 et Fig.19

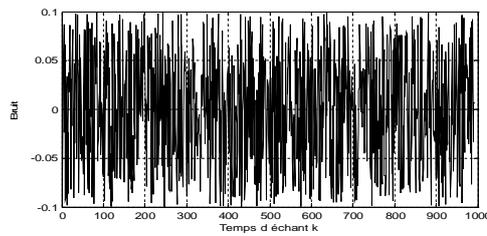


Fig. 17. Le Bruit

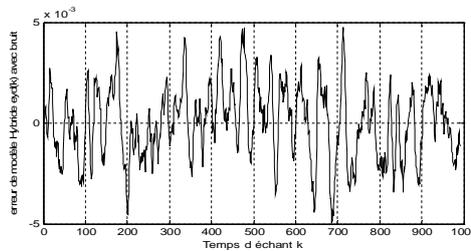


Fig. 18. Erreur commise lors de la commande hybride avec bruit (Consigne y_d)

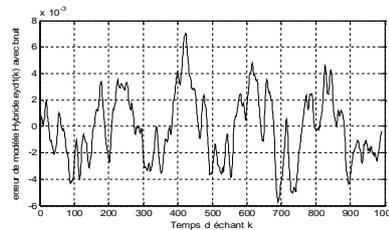


Fig. 19. Erreur commise lors de la commande hybride avec bruit (Consigne y_{d1})

Pour la commande par modèle inverse (IMC) (§ 4.3), l'erreur entre la sortie de système (11) et la sortie de modèle ANFIS est présenté dans la Fig. 20.

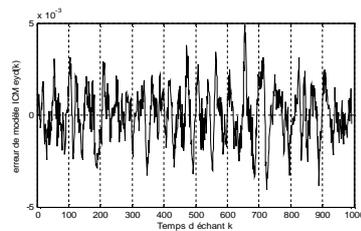


Fig. 20. Erreur commise lors de la commande IMC

Même chose comme la commande hybride en injecte le bruit (Fig. 17) a la sortie du système, la fig. 21 présente l'erreur de la commande modèle interne avec bruit.

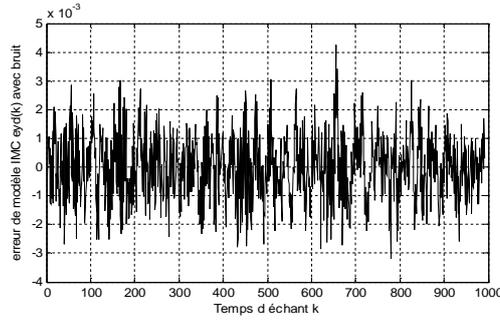


Fig. 21. Erreur commise lors de la commande IMC avec bruit

VI.CONCLUSION

Nous avons présenté, dans ce papier, trois structures de commande par réseaux neuro-flou basées sur le modèle ANFIS inverse, l'apprentissage de ce modèle a été fait après plusieurs tentatives pour arriver à une architecture optimale et minimiser le nombre de paramètres du modèle. L'architecture retenue d'ANFIS avec trois entrées et deux fonctions d'appartenance de type cloche, L'erreur de prédiction obtenue est très réduite. Le modèle retenu a été utilisé ensuite dans une structure de commande directe en boucle ouverte mais ceci suppose que le modèle inverse est presque parfait et le système est non bruité, ce qui est loin de la réalité. Pour essayer d'y remédier, nous avons utilisé le modèle inverse dans deux structure : commande hybride avec un PID classique et commande par modèle interne et nous sommes arrivés à des résultats très concluants. Pour valider les deux structures, hybride et IMC, nous avons injecté un bruit dans le modèle. Ceci n'a pas trop affecté le système commandé.

Les résultats obtenus montrent qu'un modèle ANFIS avec un bon apprentissage, s'adapte parfaitement à la dynamique d'un système avec rejet des perturbations.

REFERENCES

- [1]. H. Bleuler, D. Diez, G. Lauber, U. Meyer and D. Zlatnik, "Nonlinear neural network control with application example", In International Neural Networks Conference, vol.1, Paris, 1990, pp.201-204
- [2]. P. Branšetter and M. Skotnica, "Application of artificial neural network for speed control of asynchronous motor with vector control", Proceedings of International Conference of Košice, EPE-PEMC, 2000, pp. 6-157-6-15
- [3]. M.A. Denai F. Palis, A. Zeghib "Modeling and control of non-linear Systems using soft computing techniques" Applied Soft Computing 728– 738, 2007
- [4]. H. Gomi and M. Kawato, "Neural network control for a closed loop system using feedback error learning", Neural Networks, vol.5, 1993, pp.933-946.
- [5]. A. Guez and I. Bar-Kana, "Two degree of freedom robot adaptative controller", In American Conference on Control, San Diego, 1990, pp. 3001-3006.
- [6]. R. Jang, "ANFIS: Adaptative Neuro fuzzy inference system", Université de Californie, Berkley, CA94720, 1993.
- [7]. C.C. Jorgensen and C.Schley, "A neural network baseline problem for control of aircraft flare and touchdown", Neural networks for Control, 1990, pp.423-425
- [8]. G. Karsai, "learning to control, some practical experiments with neural networks", In International Joint Conference on Neural Networks, vol. II, Seattle, 1991, pp. 701-707
- [9]. L. Ljung, "System Identification"; Theory for the User, Prentice Hall, 1987.
- [10]. G.W. Ng, "Application of neural networks to adaptive control of nonlinear systems", Research Studies Press, Taunton, Somerset, England, (1997.
- [11]. J.Panos K.Antsaklis, and M.Passino," Introduction to Intelligent and Autonomous Control", Kluwer Academic Publishers, ISBN: 0-7923-9267-1, 1993.

- [12]. D. Psaltis, A.Sideris and A. Yamamura, "Neural controllers", In International Neural Networks Conference, vol.IV, 1987, pp.551-558
- [13]. E. Ronco and, P.J. Gawthrop, "Neural networks for modelling and control", Technal Report CSC-97008, Center for Systems and Control, Glasgow, 1997.
- [14]. G.M. Scott, J.W. Shavlik and J.W.Harmon Ray, "Refining PID controllers using neural networks", NIPS 4, Advances in Neural Information Processing Systems, 1991, pp. 555-562
- [15]. J. Steck, B. Krishnamurthy, B. McMillin and G. Leininger, "Neural modeling and control of a distillation column", In International Joint Conference on Neural Networks, vol.II, Seattle, 1991, pp. 771-773.
- [16]. P. Tai, H.A. Ryaciotaki-Boussalis and T. Kim, "the application of neural networks to control systems: a survey", Signals, systems and Computers, Record Twenty-Fourth Asilomar Conference on Vol.1, 1990.
- [17]. T. Yabuta and T. Yamada, "Possibility of neural networks controller for robot manipulators, Robotics and Automation", Proceedings, IEEE International Conference, vol.3, 1990, pp. 1686-1691
- [18]. L. Yan and C.J. Li, "Robot Learning Control Based on Recurrent Neural Network Inverse Model", Journal. of Robotic Systems, Vol. 14, 1997, pp.199-212.
- [19]. S. Yildirim, "New neural networks for adaptive control of robot manipulators", Neural Networks, International Conference, vol.3, 1997, pp.1727 - 1731
- [20]. B. Widrow, "Adaptive inverse control", In Proceedings of the IFAC Adaptive Systems in Control and Signal Processing Conference, Lund, 1988, pp.1-5
- [21]. M.T. Wishart and R.G. Harley, "Identification and Control of Induction Machines Using Artificial Neural Networks". IEEE Transaction on Industry Applications, Vol. 31, No 3, 1995.