الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Département : Génie électrique Et Informatique Industrielle

Mémoire de projet de fin d'études

pour l'obtention du diplôme de Master en

Filière : **Automatique**

Spécialité : **: Automatique et Informatique Industrielle**

Conception d'une commande robuste par mode glissant neuro-flou adaptative pour un robot manipulateur

Réalisé par

MAGRI EL OUADJERI Maissa

Présente et soutenue publiquement le (06/07/2022)

Composition du Jury :

Président	M. KHELOUAT Samir	M.C.B	ENST
Promotrice	Mme FELLAG Ratiba	M.R.B	CDTA
Co-promoteur	M. REZOUG Amar	M.C.A	ENST
Examinateur	Mme BOULAAM Karima	M.C.B	ENST
Examinateur	M. NAIT SLIMANI Boukhalfa	M.A.A	ENST

Année universitaire 2021 –2022

Conception d'une commande robuste par mode glissant neuro-flou adaptative pour un robot manipulateur

Magri El Ouadjeri Maissa
Génie électrique et Informatique industrielle
École Nationale Supérieure de Technologie
Alger, Algérie

Fellag Ratiba
Division Productique et Robotique
Centre de Développement des Technologies Avancées
Alger, Algérie

Rezoug Amar
Génie électrique et Informatique industrielle
École Nationale Supérieure de Technologie
Alger, Algérie

***Abstract*— Le but de cet article est de proposer un contrôleur robuste par mode glissant flou adaptatif pour améliorer les performances de suivi de trajectoire pour le robot PUMA 560 obtenues dans le cadre du projet de fin d'études. Ce contrôleur est basé sur les réseaux de neurones et la logique floue qui donne une technique neuro-floue par mode de glissement. L'objectif est de réduire le nombre des règles floues par une approximation floue adaptative. Ceci permettra de réduire la charge de calcul d'entrée de la commande du robot manipulateur PUMA 560 et ainsi rendre possible la mise en œuvre pratique de cette commande tout en conservant les propriétés de robustesses de la commande par mode de glissement.**

***Index Terms*—Adaptatif, Mode glissant, Logique floue, Réseaux de neurones, Neuro-floue, Robustesse.**

I. Introduction

Avec le développement rapide de la science et de la technologie, les commandes robotiques ont été largement utilisées dans de nombreuses applications industrielles et même dans notre vie quotidienne, et contrôler la position d'un robot manipulateur est le but le plus simple de contrôler un robot, et en même temps c'est l'un des problèmes les plus pertinents dans le fonctionnement des manipulateurs. Plusieurs approches ont été introduites pour traiter ce problème de contrôle, tels que le contrôle adaptatif et la commande par mode glissant et le contrôle floue [1].

La caractéristique principale de la commande par mode glissant est qu'elle utilise une loi de commande de commutation à grande vitesse pour conduire les états du système à partir de n'importe quel état initial sur une surface de commutation dans l'espace d'états, et pour maintenir les états sur la surface pour tout le temps ultérieur. Notez qu'un mouvement de glissement se produit dans lequel les états du système traversent la surface de commutation à plusieurs reprises. Lorsque le système d'état est dans la surface de glissement, la commande équivalente est capable de conserver l'état du système sur la surface de glissement [2].

L'avantage de la commande par mode glissant d'ordre un capture systématiquement les problèmes de stabilité, de suivi de trajectoire et robustesse vis-à-vis des incertitudes paramétriques et des perturbations externes. L'un des inconvénients du mode glissant est la difficulté de calcul de la commande équivalente et le problème de broutement. Ces inconvénients du contrôle par commande non linaire par mode glissant ont motivé les chercheurs à développer de nouvelles versions et stratégies de contrôle intelligent et adaptatif pour limiter leurs effets et retrouver l'efficacité de cette méthode [9].

Ces dernières années, des méthodes et des algorithmes intelligents, tels que les réseaux de neurones et la logique floue, ont été utilisés pour le contrôle de position/force des manipulateurs robotiques. En effet, un contrôleur adaptatif basé sur les réseaux de neurones et la logique floue est constitué de deux techniques de l'intelligence artificielle. Quelques exemples de l'application cette commande sur les robots manipulateurs peuvent être donnés [3]–[9]. Dans ce travail, nous allons exploiter ces deux techniques de l'intelligence artificielle ainsi que la commande par mode glissant pour proposer un contrôleur robuste adaptatif neuro-flou par mode glissant pour réaliser la tâche de suivi de trajectoire d'un robot manipulateur, le PUMA 560 . L'objectif est de calculer la commande équivalente par réseaux de neurones et le gain de la commande discontinue par l'adaptation avec la logique floue et éliminer le chattering de la commande mode glissant d'ordre un.

L'article est organisé comme suit : dans la section II, le modèle dynamique du robot manipulateur PUMA 560 est présenté. Puis, le contrôleur adaptatif par mode glissant neuro-flou est présenté dans la section III et les résultats de la simulation dans la section IV. Enfin, la section V conclut l'article.

II. Modèle dynamique du robot manipulateur

Le robot PUMA 560 est un robot à six degrés de liberté comme le montre dans la figure 1, son modèle dynamique s'écrit sous la forme de Lagrange suivante [10]:

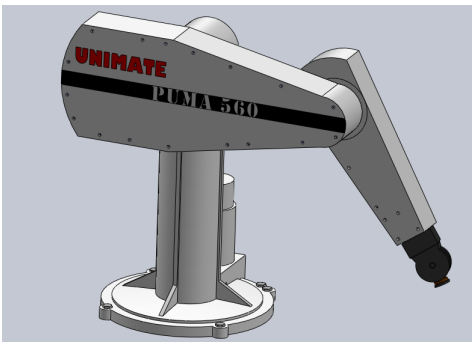


Fig. 1: Le Robot PUMA 560

$$M(q)\ddot{q} + B(q)[\dot{q}\dot{q}] + C(q)\dot{q}^2 + G(q) + F(\dot{q}) = \tau \quad (1)$$

Avec:

- $M(q)$ est une matrice d'inertie (symétrique définie positive)(6×6).
- $B(q)$ est une matrice des forces Coriolis (6×15).
- $C(q)$ est une matrice des forces centrifuges (6×6).
- $G(q)$ est un vecteur des forces de gravité (6×1).
- $F(\dot{q})$ est un vecteur de frottement (6×1).
- $\ddot{q}$ est un vecteur d'accélération (6×1).
- q est un vecteur des positions articulaires (6×1).
- $\dot{q}$ est un vecteur des vitesses articulaires (6×1).

Dans cet article, nous ne considérons que les trois premières articulations du robot PUMA 560 à savoir $q_1,\ q_2,\ q_3$ dans les simulations et les trois dernières sont considérées verrouillées. Le modèle considéré contient les incertitudes paramétriques et les forces de frottement dynamiques qui sont considérées comme perturbations. Les coefficients des matrices pour la constitution du modèle dynamique du robot PUMA 560 sont detaillés dans l'annexe A du mémoire du projet de fin d'études.

III. Conception du contrôleur adaptatif neuro-flou par mode glissant

L'objectif du contrôle est de réaliser la tâche de suivi de trajectoire par le robot PUMA 560. Nous allons concevoir des lois de commande par mode glissant en intégrant l'intelligence artificielle dans le but d'assurer le suivi de trajectoire et l'annulation de l'erreur de position par l'approximation de la commande équivalente ainsi que la suppression du chattering.

D'abord, on définit l'erreur de position comme suit:

$$e = q - q_d \quad (2)$$

- q la position de l'articulation.
- q_d la position désirée.

Puis, la surface de glissement de la commande par mode glissant définie pour la conception doit satisfaire aux spécifications de stabilité et de performance en boucle fermée, l'expression de la surface de glissement $s(e, \dot{e})$ peut-être écrite comme suit :

$$s = \lambda e + \dot{e} \quad (3)$$

avec λ constante positive.

La loi de commande adaptative basé sur les réseaux de neurones-flou par mode glissant proposée dans ce travail est constituée de deux parties: une partie de la commande équivalente qui est calculée par réseaux de neurones et l'autre partie de la commande discontinue où l'on remplace la fonction $sign$ par l'approche de la couche limite et le gain de cette commande est ajusté par l'approche de la logique floue.

La loi de commande mode glissant d'ordre un a été développé dans le cadre du projet de fin d'études, et est définie comme suit :

$$\tau = \tau_{eq} + \tau_{dis} \quad (4)$$

- τ_{eq} la loi de commande équivalente en fonction de la dynamique du système
- τ_{dis}la commande discontinue pour l'atteinte mouvement de phase.

A. La commande discontinue τ_{dis}

En générale, la commande discontinu est définie comme suit :

$$\tau_{dis} = -Ksign(s) \quad (5)$$

Mais la fonction $sign(s)$ produit des oscillations à haute fréquence, définies comme des vibrations donc sont à l'origine du chattering. Dans cet article, en remplace la fonction discontinue par la fonction continue sigmoïde (6) pour annuler problème du broutement:

$$g(s) = \frac{2}{1 + e^{-s_j}} - 1 \quad (6)$$

avec:

- j nombre d'articulation du robot PUMA 560 égale $j = 1..2..3$
- K est un gain positif adaptatif qui sera calculé par un contrôleur flou.

a) La commande par logique floue: la logique floue a connu un grand succès et a trouvé notamment une place importante dans le domaine de la commande pour une large gamme de systèmes. En ce sens, les contrôleurs basés sur la logique floue améliorent globalement les performances dynamiques et la robustesse du système commandé à partir de règles linguistiques. En fait, l'imitation des aspects qualitatifs et approximatifs du raisonnement humain a donné naissance à un autre type de système de contrôle en particulier pour des systèmes mal définis.

On propose, dans ce travail, un contrôleur flou pour estimer le gain de la commande discontinue. Ce dernier, a comme variable linguistique d'entrée la surface de glissement s et pour la sortie K. On définit chaque variable d'entrée et de sortie par sept sous-ensembles flous qui sont exprimées par NG (négatif grand), NM (négatif moyen), NP (négatif petit), Z (zéro), PG (positif grand), PM (positif moyen), PP (positif petit) avec des

fonctions d'appartenance gaussiennes (7) . On obtient alors la définition floue (fuzzification) des entrées-sorties.

$$\mu(x) = exp\left(\frac{(x-\alpha)^2}{\sigma^2}\right) \tag{7}$$

Grâce à ce contrôleur adaptatif par logique floue, le nombre de règles floues est réduit à 7 règles floues au lieu 49 règles comme définis dans le cadre de projet fin d'étude. Les règles floues sont extraites de manière à ce que la stabilité du système soit satisfaite et ces règles contiennent les relations entrée-sortie qui définissent la stratégie du contrôle. Ces règles floues linguistiques sont définies heuristiquement sous la forme suivante:

- si s_j est NG alors K_j est NG
- si s_j est NM alors K_j est NM
- si s_j est NP alors K_j est NP
- si s_j est ZO alors K_j est ZO
- si s_j est PP alors K_j est PP
- si s_j est PM alors K_j est PM
- si s_j est PG alors K_j est PG

avec $j = [1,\ 2,\ 3]$ définit numéro de l'articulation

Pour trouver la valeur réelle du gain K, la méthode de centre gravité du surface est utlisée dans l'étape de défuzzification.

B. Calcul la commande équivalente

La dynamique et les paramètres du système sont nécessaires pour pouvoir de calculer la commande équivalente. Dans ce travail, on calcule la commande équivalente par les réseaux de neurones en utilisant les outils de Matlab/Simulink.

L'approche des réseaux de neurones est une technique de l'intelligence artificielle définie par des neurones formels inter-connectés permettant la résolution de problèmes complexes tels que la reconnaissance des formes ou le traitement du langage naturel, grâce à l'ajustement des coefficients de pondération dans une phase d'apprentissage.

Les contrôleurs par réseaux de neurones sont des approximatives universels non linéaires généralement utilisés dans les schémas de commande adaptative pour estimer les incertitudes et les perturbation qui existent. Dans ce travail, la figure 2 montre l'architecture du réseau de neurone utilisé pour approximer la commande équivalente, qui est composé de trois couches : couche d'entrée, couche intermédiaire et couche de sortie. La couche d'entrée contient de la surface de glissement pour chaque articulation. La couche intermédiaire appelée couche cachée est composée de n neurones, qui sont dotés d'une fonction gaussienne pour chaque neurone. La couche de sortie donne l'estimation du contrôle équivalent.

La formule de la commande équivalente par réseaux de neurones est définie comme suit :

$$\tau_{eq} = \sum_{i=1}^{n} e\left(-\frac{\|s_j - c_i\|}{\sigma_j^2}\right) \tag{8}$$

$$\tau_{eq} = \sum_{i=1}^{n} W_i \sigma_j \tag{9}$$

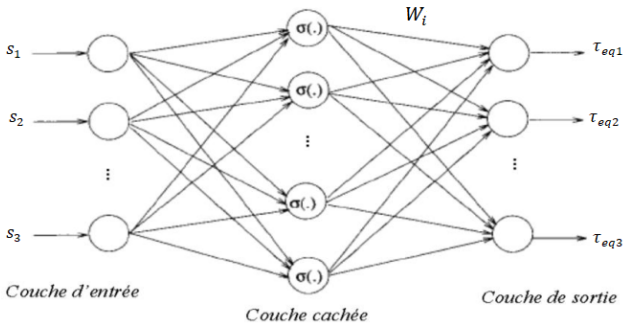


Fig. 2: Structure d'un réseau de neurones

Avec:

- le i nombre de neurones de la couche cachée
- le c_i est le neurone central position du neurone i.
- le σ est l'écart type de la fonction d'activation gaussienne.
- les j est le nombre de l'articulation.
- les $W_i$sont les poids du réseau de neurones, ce sont les paramètres à optimiser en vue de l'apprentissage.

Dans ce travail, on utilise le réseau de neurones pour calculer les poids et les paramètres de l'apprentissage pour approximer la commande équivalente.

IV. Résultats de simulation

Les simulations ont été effectuées pour l'application du contrôleur adaptatif neuro-flou par mode glissant sur le modèle dynamique d'un robot PUMA 560 $3ddl$ pour accomplir la tâche de suivi de trajectoire. La figure 3 montre le schéma fonctionnel de contrôleur proposé.

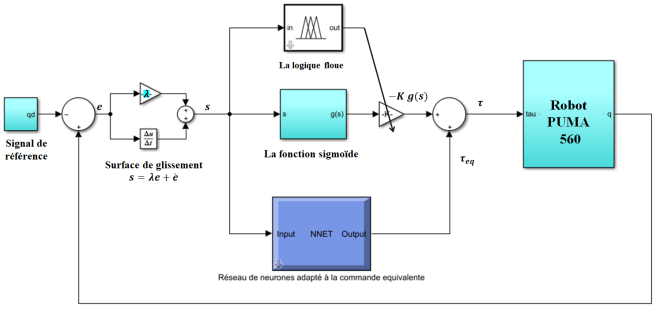


Fig. 3: Schéma fonctionnel du système de contrôle proposé

L'outil du réseau de neurones de Matlab/Simulink a été utilisé dans la simulation, pour construire un réseau de neurones constitué par trois neurones dans la couche d'entrée dont $[s_1, s_2, s_3]$ et deux couches cachées, qui sont composées de $n = 100$ neurones, et de trois neurones dans la couche de sortie ($\tau_{eq1}, \tau_{eq2}, \tau_{eq3}$). La fonction d'activation gaussienne est définie dans l'équation (8), et les poids $W_i = 0$ sont initialisés a zéros.

Dans l'objectif d'évaluer la commande adaptatif neuro-flou par mode glissant proposé dans ce travail, nous allons la

comparer avec la commande par mode glissant d'ordre un (MG-classique) conçue dans le cadre du projet de fin d'études. Nous vérifierons la précision de suivi de trajectoire et la suppression du chattering

Une trajectoire sinusoïdale désirée q_d a été considérée pour les trois articulations du robot PUMA 560 ayant la forme ci-dessous:

$$\begin{cases} q_{d1} = sin(t) \\ q_{d2} = sin(2t) \\ q_{d3} = sin(4t) + 0.5 \end{cases} \tag{10}$$

Les conditions initiales pour les positions et les vitesses articulaires sont les suivantes:

$$\begin{aligned} q_0 &= [q_1, q_2, q_3] = [0.5, 0.75, 0](rad) \\ \dot{q}_0 &= [\dot{q}_1, \dot{q}_2, \dot{q}_3] = [0, 0, 0](rad/s) \end{aligned} \tag{11}$$

Les résultats de simulation sont illustrées dans les figures suivantes:

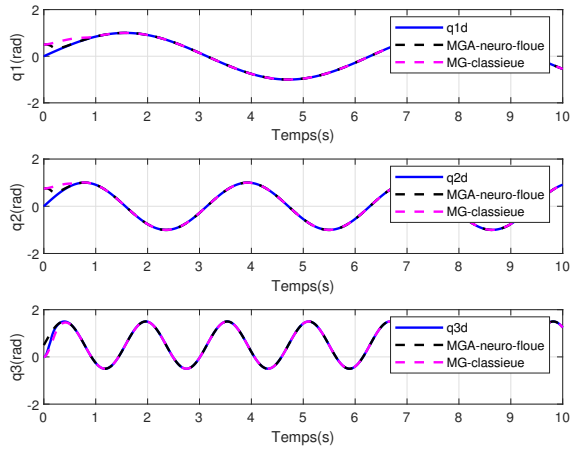


Fig. 4: La trajectoire de position pour les trois axes par les commandes robustes neuro-floue et MG-classique

A. *Interprétation des résultât*

La figure 4 montre le résultat de suivi de trajectoire, la figure 5 représente l'erreur de position et les figures 6a, 6b, 6c montrent les commandes appliquées pour les trois axes des articulations du robot PUMA 560 par les deux contrôleurs: le contrôleur adaptatif neuro-flou par mode glissant et le contrôleur mode glissant d'ordre un (MG-classique).

On remarque que le contrôleur adaptatif neuro-flou mode glissant donne un bon suivi de trajectoires désirées et que l'erreur de position converge vers le zéro avec une petite délai. Cette commande améliore la précision du suivi de trajectoire et diminue le temps de convergence. On peut remarquer aussi que le broutement a disparu dans les couples de commande et la robustesse vis-à-vis des incertitudes et des perturbations.

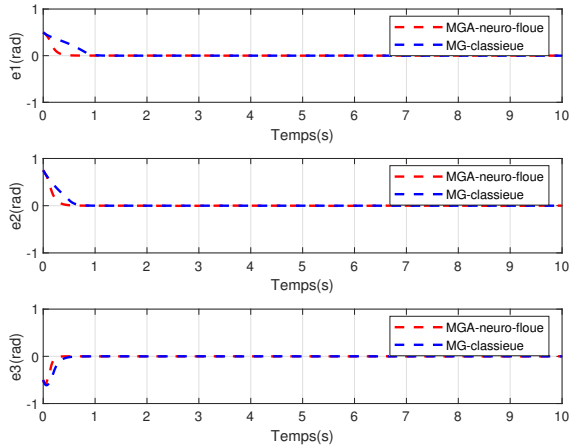


Fig. 5: L'erreur de position pour les trois axes par les commandes robustes neuro-floue et MG-classique

Temps de convergence (s)	Axe 1	Axe 2	Axe 3
La commande MG-classique	1	0.742	0.55
La commande neuro-flou MG	**0.315**	**0.413**	**0.278**

TABLE I: Tableau des temps convergence des erreurs de position vers zéro

B. *comparaison en termes des racines des erreurs quadratiques moyennes*

Le tableau I représente le temps convergence des erreurs de position vers zéro sur les trois axes pour la commande mode glissant d'ordre un et la commande neuro-flou adaptative par mode glissant d'ordre un au robot PUMA 560, on remarque que la commande proposée dans ce travail a réduit le temps de convergence vers la trajectoire désiré pour les trois axes.

C. *comparaison en termes des racines des erreurs quadratiques moyennes*

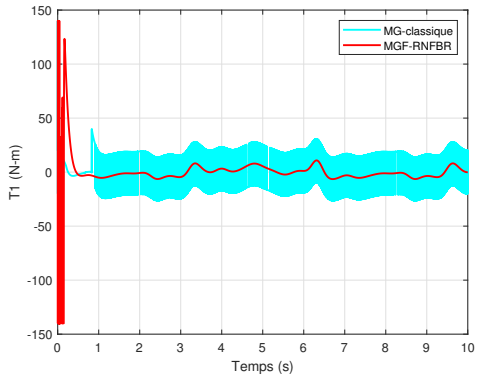
La racine des erreurs quadratiques moyennes (REQM) est calculée pour chaque articulation comme suit:

$$REQM = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(e)^2} \tag{12}$$

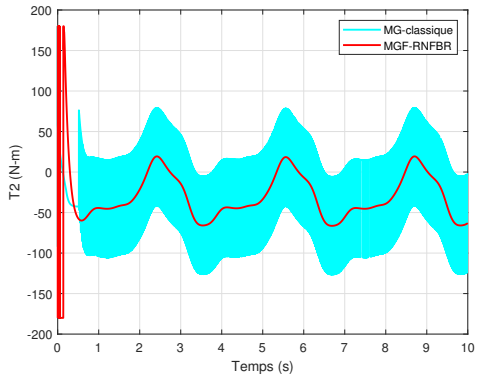
REQM (rad)	Axe 1	Axe 2	Axe 3
La commande MG-classique	0.0885	0.0986	0.0799
La commande neuro-flou MG	**0.0543**	**0.0766**	**0.0600**

TABLE II: La racine des erreurs quadratiques moyennes

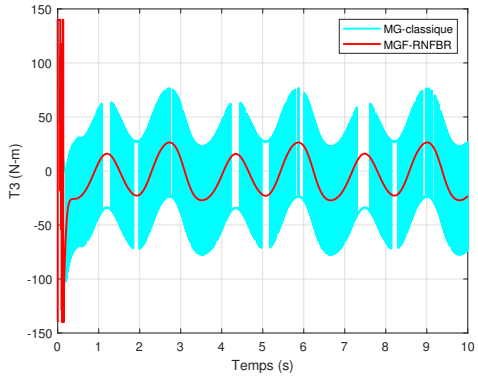
Le tableau II représente le résultat de calcul des racines des erreurs quadratiques moyennes sur les trois axes pour chaque contrôleur par mode glissant appliqué au robot PUMA 560. D'après ces résultats peut conclure que le contrôleur adaptatif neuro-flou par mode glissant proposé dans ce travail a donné les meilleurs résultats.



(a) Les commandes MG-classique et neuro-flou pour l'axe 1 du robot



(b) Les commandes MG-classique et neuro-flou pour l'axe 2 du robot



(c) Les commandes MG-classique et neuro-flou pour l'axe 3 du robot

Fig. 6: Les commandes appliquées pour trois axes du robot PUMA 560

V. Conclusion

Un contrôleur adaptatif mode glissant basé sur réseau de neurone et la logique flou est utilisée pour améliorer les performance et la précision du robot PUMA 560 pour réaliser le suivi de trajectoire désirées. Les règles de la logique floue sont utilisées pour l'adaptation du gain de la commande discontinue. L'outil de Matlab/Simulink des réseaux de neurones est utilisé pour calculer le contrôle équivalent. L'application du contrôleur adaptatif proposé sur robot PUMA 560 a permis de réduire le temps de convergence dans le mode de suivi le trajectoire et de minimiser les problèmes de broutement dans modes glissants d'ordre un classique. Les résultat de simulation démontrent que la commande proposée dans cet article est une commande robuste vis-à-vis des variations paramétriques et des perturbations.

References

[1] Yang, Chenguang, et al. "Adaptive RBFNN control of robot manipulators with finite-time convergence." IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society. IEEE, 2016.

[2] Ak, Ayca Gokhan, and Galip Cansever. "Three link robot control with fuzzy sliding mode controller based on RBF neural network." 2006 IEEE Conference on Computer Aided Control System Design, 2006 IEEE International Conference on Control Applications, 2006 IEEE International Symposium on Intelligent Control. IEEE, 2006.

[3] Belhachat, Faiza, et al. "Commande neuro-floue d'un hacheur MPPT." 4éme conférence internationale" Computer Integrated Manufacturing", CIP. Vol. 7. 2007.

[4] Chaudhary, Himanshu, et al. "Adaptive neuro fuzzy based hybrid force/position control for an industrial robot manipulator." Journal of Intelligent Manufacturing 27.6 (2016): 1299-1308.

[5] Bachir, Ouamri, and Ahmed-foitih Zoubir. "Adaptive neuro-fuzzy inference system based control of puma 600 robot manipulator." International Journal of Electrical and Computer Engineering 2.1 (2012): 90.

[6] Alavandar, Srinivasan, and M. J. Nigam. "Adaptive Neuro-Fuzzy Inference System based control of six DOF robot manipulator." Journal of Engineering Science and Technology Review 1.1 (2008).

[7] Alavandar, Srinivasan, and M. J. Nigam. "Adaptive Neuro-Fuzzy Inference System based control of six DOF robot manipulator." Journal of Engineering Science and Technology Review 1.1 (2008).

[8] Theodoridis, Dimitrios C., Yiannis S. Boutalis, and Manolis A. Christodoulou. "A new adaptive neuro-fuzzy controller for trajectory tracking of robot manipulators." International Journal of Robotics and Automation 26.1 (2011): 64.

[9] HALA Bezine, Nabil Derbel and Adel M. Alimi, "Fuzzy control of robot manipulators: some issues on design and rule base size reduction, " Engineering Applications of Artificial Intelligence, Sept 2002, vol. 15, Issue 5, pp. 401-416.

[10] Piltan, Farzin, et al. "PUMA-560 robot manipulator position sliding mode control methods using MATLAB/SIMULINK and their integration into graduate/undergraduate nonlinear control, robotics and MATLAB courses." International Journal of Robotics and Automation 3.3 (2012): 106-150.