



HAL
open science

Conception d'un contrôleur flou-adaptatif: application à la commande d'un robot de traite de vaches laitières

Christophe Collewet, G. Rault, S. Quellec, P. Marchal

► To cite this version:

Christophe Collewet, G. Rault, S. Quellec, P. Marchal. Conception d'un contrôleur flou-adaptatif: application à la commande d'un robot de traite de vaches laitières. Journées automatique agricole et agro-alimentaire, Oct 1997, Clermont-Ferrand, France. pp.207-214. hal-02577374

HAL Id: hal-02577374

<https://hal.inrae.fr/hal-02577374v1>

Submitted on 30 Sep 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Conception d'un contrôleur flou auto-adaptatif : application à la commande d'un robot de traite de vaches laitières

Christophe Collewet, Guylaine Rault, Stéphane Quéllec, Philippe Marchal

*Cemagref Rennes, 17 avenue de Cucillé,
35044 Rennes Cedex*

Résumé : Les contraintes propres à la robotique agricole nécessitent l'utilisation de lois de commande performantes tout en restant implantables sur des architectures à faibles moyens de calcul. Un contrôleur flou auto-adaptatif a été conçu pour la commande articulaire d'un robot de traite de vaches laitières. Des simulations montrent le comportement d'un des axes lorsqu'il est soumis à de fortes modifications en ligne du comportement de son actionneur. De plus, cet algorithme a été porté sur une carte standard et les temps de calcul obtenus montrent qu'il peut parfaitement appréhender des applications nécessitant des fréquences d'échantillonnage élevées.

Mots clés : contrôleur flou, apprentissage spécialisé, modèle de référence, commande articulaire, métarègle.

Introduction

Parallèlement aux robots dits industriels d'autres secteurs d'activités ont vu apparaître la robotisation. L'agriculture au sens large en est un exemple. Depuis l'avènement et le développement de l'électronique dans les années 70, les machines agricoles sont devenues de plus en plus intelligentes et la robotique agricole en est aujourd'hui un aboutissement [10]. Beaucoup de travaux relatent de tels robots et ce dans divers domaines de l'agriculture [4], [14], [19]. La robotisation de la traite des vaches laitières est un exemple type. C'est une évolution très attendue par les exploitants, elle est vue comme le moyen de diminuer la pénibilité de leurs travaux et d'éviter leur présence biquotidienne toute l'année. En effet, dans la plupart des exploitations, c'est la seule opération manuelle restant à effectuer. Depuis une dizaine d'année maintenant, elle fait l'objet de nombreuses recherches principalement en Europe mais aussi aux États-Unis et au Japon. Une quarantaine de prototypes sont installés en test sur site [13].

Dans le cadre du projet Eureka CIMIS (Complete Integrated Milking System), le *Cemagref* travaille sur deux points clés de la robotisation de la traite : la localisation des trayons dans l'espace et l'élaboration d'une commande adaptée à ce type de robot. En effet, ces robots se distinguent de leurs homologues rencontrés dans d'autres domaines par les caractéristiques suivantes :

- La précision demandée est faible tout comme les vitesses et accélérations requises. Ils ont aussi la particularité de concerner de faibles volumes de marché. Il en résulte que, pour

proposer un équipement à un prix abordable et pour tenir compte des conditions de fonctionnement décrites ci-dessus, la fabrication de ces robots sera peu soignée, les tolérances importantes favorisant ainsi jeux de fonctionnement et usure des pièces en mouvement, les organes de commande et de transmission seront choisis bon marché et enfin l'interchangeabilité des pièces sujettes à caution.

Ce premier aspect fait apparaître qu'une modélisation rigoureuse est illusoire.

- Par ailleurs, ils sont soumis à des contraintes d'utilisation très sévères de part l'ambiance dans laquelle ils sont amenés à évoluer : travaux en milieux corrosifs imposés par des nettoyages stricts, taux d'humidité élevé, grande amplitude de variation des températures. Ils peuvent être éventuellement objets de chocs physiques qui conduisent alors à des modifications de la structure même du robot.

De ce deuxième aspect il résulte que ces robots doivent être considérés non seulement comme des systèmes difficilement modélisables mais aussi variables dans le temps.

D'après les caractéristiques énumérées ci-dessus, seul l'emploi de techniques de commande adaptative peut aboutir à la bonne réalisation des tâches à effectuer. Plus précisément, si on utilise le modèle dit géométrique pour la commande du robot, le problème se pose alors en terme de conception d'une loi de commande pour chaque axe du robot.

Le présent papier s'articulera comme suit. Tout d'abord, nous évoquerons les différents type de correcteur et nous nous focaliserons ensuite sur celui retenu. L'architecture d'apprentissage et l'algorithme associé seront ensuite étudiés. Après avoir modélisé le système à commander des résultats de simulation seront présentés. Une dernière partie conclura par les principaux résultats obtenus.

Choix du contrôleur

L'automatique classique a montré que de très bons résultats peuvent être obtenus pourvu que la connaissance du modèle du système à commander soit suffisamment juste [5], [6], [23]. Le cas échéant, des algorithmes d'adaptation peuvent, dans une certaine mesure, prendre en compte des variations du système grâce notamment à des techniques d'estimation paramétrique en ligne [15]. L'utilisation de ces algorithmes nécessite cependant des développements mathématiques importants imposant des temps de calcul incompatibles avec une implantation matérielle à faibles moyens de calcul [8], [24], typiquement des cartes à microprocesseur 8 bits en robotique agricole.

Les techniques de commande basées sur la théorie des ensembles flous [22] ont elles aussi largement montré et ce dans de nombreux domaines qu'elles pouvaient appréhender avec succès des applications complexes où les systèmes à commander étaient en particulier mal connus. Citons par exemple la commande de robots [20] ou la synthèse de correcteur [2], [9] pour ce qui nous concerne. Pour un recueil d'applications industrielles on pourra se reporter aussi à [16], [17], [21] et partiellement à [7]. Ces applications conduites avec succès nous ont convaincu d'aborder ce type d'approche, *a priori* mieux adaptée à la commande des robots agricoles.

Les applications à base de logique floue que l'on rencontre habituellement sont souvent menées à bien grâce à la connaissance *a priori* dont un expert peut disposer de tel ou tel problème. Cependant, pour atteindre les performances désirées, le processus d'ajustement des fonctions d'appartenance ou des conclusions des règles peut s'avérer être une tâche longue et

difficile, parce que souvent réalisée par essais et erreurs. C'est pourquoi, très tôt des recherches concernant des structures auto-adaptatives ont été conduites [11]. Le système flou est alors conçu non plus grâce à la connaissance *a priori* mais grâce à des mesures prises sur le système lui même, on parle alors d'apprentissage. Ce type de système, très intéressant, fait actuellement l'objet de nombreux travaux que nous avons rappelés dans [3]. Parmi les types de contrôleur flou possible celui de type Takagi-Sugeno [18] est un de ceux le plus à même d'être implanté sur une architecture à faible moyen de calcul. Le chapitre suivant décrit un tel correcteur.

Description du correcteur flou étudié

Soit un système flou de type Takagi-Sugeno à n entrées x_i pour $i = 1 \dots n$ et une sortie $y(x_1, \dots, x_n)$. On suppose les x_i bornés de bornes x_{i_inf} , x_{i_sup} .

Les différentes valeurs linguistiques que pourront prendre les variables x_i sont représentées par p_i sous-ensembles ou partitions flous sur leurs univers de discours respectifs.

On régit ce système flou par K règles $R^{(k)}$ du type :

$$R^{(k)} : \text{si } (x_1 \text{ est } A_{1,k}) \text{ et } \dots \text{ et } (x_n \text{ est } A_{n,k}) \text{ alors } y(X) = W_k \quad (1)$$

où :

- $A_{i,k}$ est le sous-ensemble flou intervenant dans la règle k pour l'entrée x_i ,
- W_k est la sortie non floue du système quand seule la règle k est active,
- $K = \prod_i p_i$.

Soit α_k la valeur de vérité de la règle k et $\mu_{A_{i,j}}$ la fonction d'appartenance de la variable x_i au sous-ensemble flou $A_{i,j}$. On impose aux $\mu_{A_{i,j}}$ des formes triangulaires centrées en $x_{i,j}$ et se coupant toutes en 1/2, on adopte comme processus de défuzzification le calcul du centre de gravité relatif aux conclusions des règles et on choisit comme T-norm la fonction produit. On a alors :

$$y(X) = \sum_k \alpha_k W_k \quad (2)$$

Nous allons, dans le chapitre suivant, associer à ce correcteur une structure d'apprentissage afin d'obtenir les conclusions des règles.

Structure d'apprentissage

La contrainte d'un apprentissage en ligne est de devoir utiliser directement les consignes générées pour le déplacement du robot et non des signaux particuliers propres à exciter tous les modes du système. Pour ce faire, un apprentissage de type spécialisé est utilisé [12]. Une architecture permettant un tel apprentissage est représentée sur la figure 1.

On note :

- r la consigne,
- s la sortie du système,
- v la vitesse de la sortie du système,
- s_m la sortie du modèle de référence,
- e le signal d'erreur,
- de la variation du signal d'erreur,
- m la commande envoyée au système,

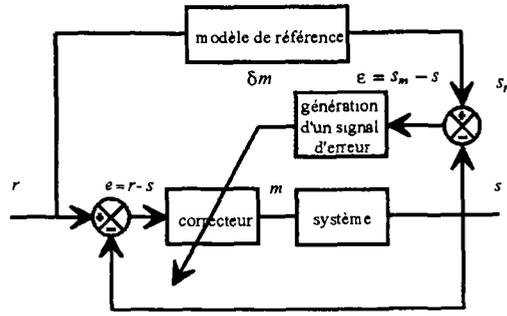


Figure 1: structure d'apprentissage spécialisé avec modèle de référence.

- δm le signal d'erreur sur la commande.

On impose au système bouclé de se comporter comme un modèle de référence choisi du second ordre : $\ddot{s}_m + 2\xi\omega_0\dot{s}_m + \omega_0^2 s_m = \omega_0^2 r$. On peut ainsi facilement imposer le taux de dépassement et le temps de montée en ajustant respectivement ξ et ω_0 . On a choisi pour la suite $\xi = 0.9$ et $\omega_0 = 20$ rad/s.

Grâce au signal ε , un signal d'erreur δm sur la commande est reconstruit pour la mise à jour des conclusions des règles. Dans ce but, on cherche, à chaque itération d'apprentissage, à minimiser le critère J' :

$$J' = \frac{1}{2}(s_m - s)^2 \quad (3)$$

Par descente du gradient on reconstruit un signal d'erreur sur la commande $\delta m = m^* - m$ où m^* est la commande désirée mais inconnue :

$$\delta m = \lambda'(s_m - s) \frac{\partial s}{\partial m} \quad (4)$$

Par contre, on connaît qualitativement les conséquences d'une action sur le système, plus précisément on sait que $\delta s / \delta m > 0$. On supposera de plus que ce terme est constant. Cette approche a déjà été testée avec succès dans [1]. On applique ensuite une nouvelle fois l'algorithme du gradient sur $J = \frac{1}{2}\delta m^2$ dans le but de minimiser (4). En utilisant (2), et, tout calcul fait il vient : $\Delta W_k = \lambda \delta m \alpha_k$ où on a posé $\lambda = \lambda' \frac{\partial s}{\partial m}$. Pratiquement, on rajoute un terme de moment η pour lisser les variations trop rapides du gradient :

$$\Delta W_k = \lambda \delta m \alpha_k + \eta \Delta W_{k-1} \quad (5)$$

Sauf spécifications contraires, les paramètres utilisés seront $\lambda = 0.3$ et $\eta = 0.8$.

Modélisation du système à commander

Les rotations étant obtenues par translation de vérins de type électrique la fonction de transfert liant la position s de la tige du vérin à la commande m peut s'écrire :

$$G(p) = \frac{K}{p(p+a)} \quad (6)$$

où a est l'inverse de la constante de temps du système et K le gain.

Pour simuler les conséquences d'un système variable dans le temps les paramètres a et K seront supposés non constants. De plus, des termes de frottements seront ajoutés. Soient m_p et m_n les composantes du frottements respectivement dans les sens croissants et décroissants. Dans un soucis de simplification, ils seront supposés indépendants de la position s de la tige du vérin. D'autre part, toute référence au temps d'apprentissage doit être comprise comme le

temps « réellement » pris par le robot pour apprendre y compris la durée des mouvements et non seulement les temps de calcul.

Résultats de simulation

Utilisation de métarègles

Lors des déplacements habituels du robot, des consignes articulaires de chaque axe sont calculées pour suivre des profils de vitesse trapézoïdal. Ce sont ces signaux qui sont utilisés pour l'apprentissage. Cependant, d'autres types de déplacement peuvent être imposés. Des simulations préalables ont mis en évidence que lors de ces mouvements le comportement du système bouclé s'est montré insuffisant. Le correcteur avait pour caractéristique principale : 3 entrées e , de et v avec respectivement 5 sous-ensembles flous pour e et de et 3 pour v . Il s'est avéré que les consignes trapézoïdales étaient en fait des signaux beaucoup trop pauvres pour activer toutes les règles du correcteur. Pour répondre à ce problème, une démarche consistant à utiliser des métarègles regroupant toutes les règles devant conduire aux mêmes conclusions est employée.

D'après la connaissance *a priori* dont on peut disposer du système à commander on peut réaliser la base de règles fixes suivante où un même comportement vis à vis des frottements est désiré :

$$\begin{array}{ll} \text{si } (v > 0 \text{ ou } (v = 0 \text{ et } e > 0) \text{ ou } (v = 0 \text{ et } e = 0 \text{ et } de > 0)) & \text{alors } x_3 = x_{3_sup} \\ \text{si } (v < 0 \text{ ou } (v = 0 \text{ et } e < 0) \text{ ou } (v = 0 \text{ et } e = 0 \text{ et } de < 0)) & \text{alors } x_3 = x_{3_inf} \\ \text{si } (v = 0 \text{ et } e = 0 \text{ et } de = 0) & \text{alors } x_3 = \frac{1}{2}(x_{3_sup} + x_{3_inf}) \end{array}$$

A l'aide de la variable x_3 fournie par une métarègle et en rajoutant les variables $x_1 = e$ et $x_2 = de$ on réalise ensuite un correcteur flou à 3 entrées. On impose $p_1 = p_2 = 2$ et $p_3 = 3$ et des fonctions d'appartenances triangulaires centrées en x_{1_inf} , x_{1_sup} , x_{2_inf} , x_{2_sup} et x_{3_inf} , $\frac{1}{2}(x_{3_sup} + x_{3_inf})$, x_{3_sup} .

Comportement du système sans perturbation

La figure 2 rapporte le comportement du système bouclé en phase d'apprentissage, tous les poids W_k étant initialisés à zéro. Les figures 3 et 4 montrent que, cette fois, toutes les règles ont bien été apprises. De plus, dans [3] nous avons pu montrer que d'une part à chaque métarègles lui correspondait un correcteur et que d'autre part le correcteur total était stable. Par ailleurs, les temps de calcul de l'algorithme complet ont été mesurés sur les bases d'un microprocesseur 8 bits bas de gamme (80c51). Suivant la fréquence d'horloge de ce dernier, des fréquences d'échantillonnage de 700 Hz peuvent être atteinte et 200 Hz en standard.

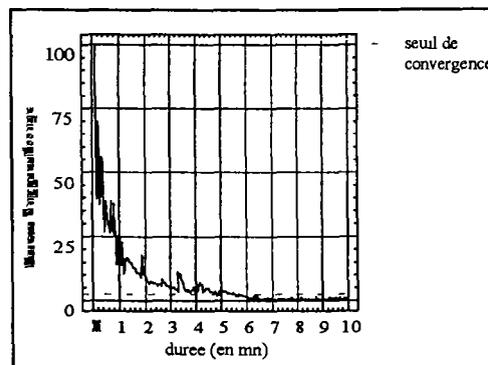


Figure 2 : courbe d'apprentissage.

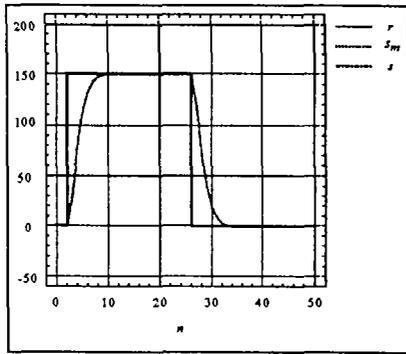


Figure 3 : réponse à un signal de consigne carrée après un apprentissage de 20 mn

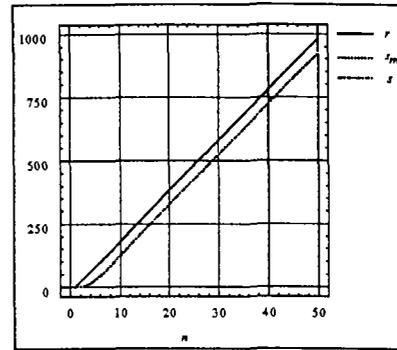


Figure 4 : réponse à un signal de consigne rampe après un apprentissage de 20 mn.

Comportement du système avec perturbations en ligne

On se propose de présenter maintenant le comportement en ligne du système face à une perturbation sinusoïdale d'un des paramètres a , K ou m_p au cours d'une trajectoire de durée d_r :

$$\begin{cases} \text{pour } nT \leq d_r : x(nT) = x(0) \left(1 + a_p \cdot \sin \left(2 \cdot \pi \cdot n \cdot \frac{T}{T_p} \right) \right) \\ \text{sinon } x(nT) = x(d_r) \end{cases}$$

avec x le paramètre perturbé, a_p et T_p respectivement l'amplitude et la période de la perturbation. T_p est fixé à d_r , pour que la perturbation soit maximum en fin de trajectoire, a_p est fonction de x et vaut respectivement 0.25, 0.25 et 1 pour les paramètres a , K et m_p .

Les figures 5, 6 et 7 représentent le comportement en ligne du système perturbé respectivement par les variations de a , K et m_p . On peut noter que dans tous les cas l'apprentissage en ligne permet de minimiser considérablement l'effet des perturbations en maintenant ε à une valeur faible, valeur à comparer à ε lorsque l'apprentissage est inhibé.

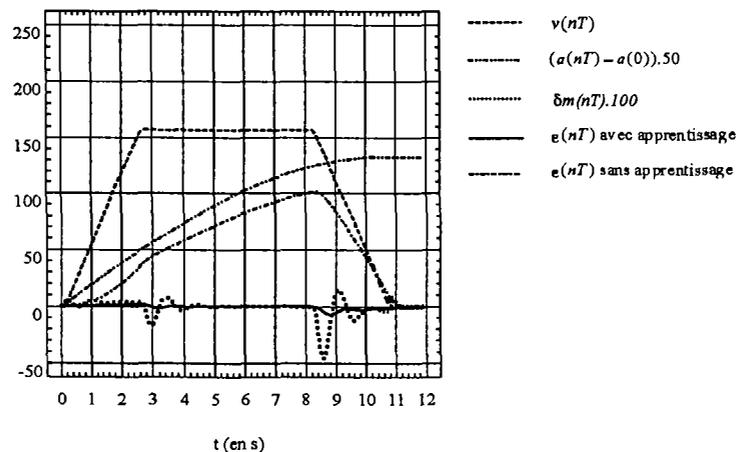


Figure 5 : influence d'une variation du paramètre a au cours d'une trajectoire.

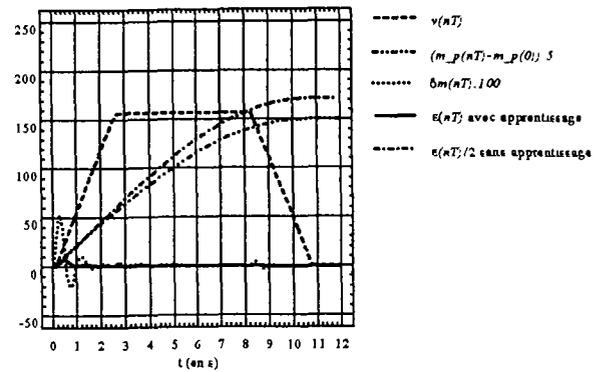
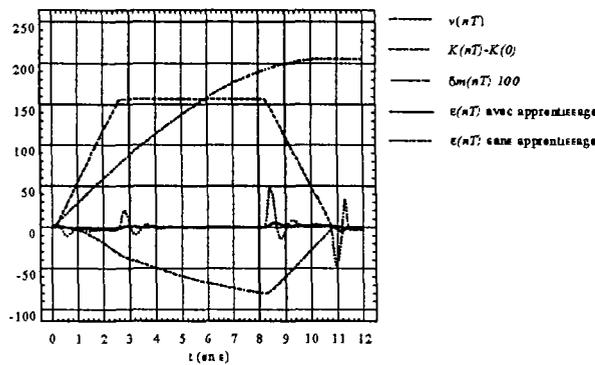


Figure 6 : influence d'une variation du paramètre K au cours d'une trajectoire. Figure 7 : influence d'une variation du paramètre m_p au cours d'une trajectoire.

Conclusion

Dans cette étude nous avons conçu un correcteur flou auto-adaptatif répondant aux contraintes de la robotique agricole. Les résultats obtenus doivent se généraliser à d'autres systèmes difficiles à modéliser et variables dans le temps ayant grossièrement un comportement analogue au système étudié. En effet, seule une connaissance qualitative a été nécessaire. Les résultats de simulations sont très encourageants puisque de fortes variations des paramètres du modèles ont pu être prises en compte (de 25 à 100% suivant les cas).

L'utilisation de métarègles a permis de contourner le problème de signaux d'apprentissage relativement pauvres imposés par l'application et a permis de résoudre convenablement le problème des frottements.

Par ailleurs, la portabilité temps réel de l'algorithme entier a été vérifié sur une architecture à faible ressource et des fréquences en standard de 200 Hz ont pu être obtenues répondant ainsi aux préoccupations rencontrées en robotique agricole.

Bibliographie

- [1] Bersini H., Nordvik J.P., Bonarini A., A Simple Direct Adaptive Fuzzy Controller Derived from its Neural Equivalent, Proceedings of the 2nd IEEE international conference on fuzzy systems, pp. 345-350, 1993.
- [2] Chen C.L., Chen P.C., Chen C.O.K., A Pneumatic Model-Following Control System Using a Fuzzy Adaptive Controller, Automatica, vol. 29, pp; 1101-1105, 1993.
- [3] Collewet C., Rault G., Quellec S., Marchal P., Fuzzy adaptive controller design for the joint space control of an agricultural robot, à paraître dans Fuzzy Sets and Systems.
- [4] Ito N., Current topics in agricultural robotics and food industry, in the second IARP Workshop on Robotics in Agriculture and the food Industry, P. Gandolfo and G. Sandini, Eds. Ansaldo Spa, DIST - Univeristy of Genova, June 1991, pp. 19-29.
- [5] Foulloy L., Titli A., Logique Floue, Observatoire français des techniques avancées, Eds. Masson, Paris, 1994.
- [6] He S.Z., Tan S., Xu F.L., Wang P.Z., Fuzzy self-tuning of pid controllers, Fuzzy Sets and Systems, vol. 56, pp. 37-46, 1993.

- [7] Lee C.C., Fuzzy logic in control systems : Fuzzy logic controller- Part II, *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 20, n°2, pp. 404-435, 1990.
- [8] Li Y.F., Lau C.C., Application of Fuzzy Control for Servo Systems, *Proceedings of the IEEE International conference on Robotics and Automation*, Pennsylvanie, Philadelphie, 1988.
- [9] Lim C.C., Hiyama T., Experimental Implementation of a fuzzy logic control scheme for a servomotor, *Mechatronics*, vol. 3, n°1, pp. 39-47, 1993.
- [10] Marchant J.A., *Mechatronics in agricultural engineering*, *Mechatronics*, vol. 1, n°1, pp. 11-18, 1991.
- [11] Procyk T.J., Mamdani E.H., A linguistic self organising process controller, *Automatica*, vol. 15, pp. 15-30, 1979.
- [12] Psaltis D., Sideris A., Yamamura A., Neural Controllers, *Proceedings of the first International Conference on Neural Networks*, vol. 4, San-Diego, pp. 551-558, 1987.
- [13] Rossing W., Hogewerf P.H., State of the Art of Automatic Milking Systems, *Computers and electronic in Agriculture*, vol. 17, pp. 1-17, 1997.
- [14] Sarig Y., Robotics of fruits harvesting: a state-of-the-art review, *Journal of Agricultural Engineering Research*, vol. 54, n°4, pp. 265-280, April 1993.
- [15] Slotine J.-J.E., Li W., *Applied Nonlinear Control*, Englewood Cliffs, NJ Prentice-Hall, 1991.
- [16] Sugeno M., Eds., *Industrial applications of Fuzzy control*, Amsterdam , North-Holland, 1985.
- [17] Sugeno M. (Guest Editor), Special issue on Fuzzy control, *Fuzzy Sets and Systems*, vol. 26, n°2, 1988.
- [18] Takagi T., Sugeno M., Fuzzy identification of systems and its applications to modeling and control, *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 15, pp. 116-132, 1985.
- [19] Tillet N.D., Robotic manipulators in horticulture: A review, *Journal of Agricultural Engineering Research*, vol. 55, n°2, pp. 89-105, June 1993.
- [20] Wu A.T., Ju M.S., Tsuei Y.G., Comparison of fuzzy Logic and self-tuning adaptive control of single-link Flexible Arm, *Mechatronics*, vol. 3, n°4, pp. 451-464, 1993.
- [21] Yamakawa T., Hirota K., Special issue of fuzzy logic control to industry, *Fuzzy Sets and Systems*, vol. 32, pp. 117-237, 1989.
- [22] Zadeh L., Fuzzy sets, *Information and Control*, vol. 8, pp. 338-353, 1965.
- [23] Zhou J., Coiffet P.: Fuzzy control of robots, *IEEE international conference on Fuzzy systems*, March 8-12, San-Diego, CA, USA, pp. 1357-1364, 1992.
- [24] Zomaya A., Reinforcement Learning for the Adaptive Control of Nonlinear Systems. *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 24, n°2, pp. 357-363, 1994.