

*Contrôle Robuste des
Système non Linéaire par
des Approches du Type
Neuro-Flou*

Fatima Zohra DAIKH *

*Université de Mascara
(Algérie)*

Fatima_daikh@yahoo.fr

=====

Hamadouche mohamed
el amine

*Université de Mascara
(Algérie)*

ham1879@yahoo.fr

Abstract:

In this article, we have dealt with the exploitation of the properties of Artificial Intelligence in the field of automation. Our work is devoted to the use of neuro-fuzzy networks and precisely STFIS (Self Tuning Fuzzy Inference System) and sliding mode control (SMC). We proposed a hybridization between the STFIS controller and the sliding mode to improve the performance of nonlinear system control. We will use an STFIS network to approximate the dynamics assumed to be totally unknown. After having introduced the principle of the proposed approaches, we will present their implementations for inverted pendulum control. The work is validated by simulations under the MATLAB environment.

Keywords: Neural fuzzy networks, STFIS, Sliding mode control, Inverted pendulum

Résumé:

Dans cet article, nous avons traité l'exploitation des propriétés de l'Intelligence Artificielle dans le domaine de l'automatique. Notre travail est consacré sur l'utilisation des réseaux neuro-flous et précisément STFIS (Self Tuning Fuzzy Inference System) et la commande par mode glissant(SMC). Nous avons proposé une hybridation entre le contrôleur STFIS et le mode glissant pour améliorer les performances de contrôle de système non linéaire. On utilisera un réseau STFIS pour approximer la dynamique supposée totalement inconnue. Après avoir introduit le principe des approches proposées, nous présenterons leurs mises en œuvre pour la commande de pendule inversé. Les travaux sont validés par des simulations sous l'environnement MATLAB.

Mots Clés : Réseaux neuro-flous, STFIS, La commande mode glissant, Pendule

1. INTRODUCTION

Les méthodes classiques de l'automatique ont été largement appliquées à de nombreux problèmes de régulation industrielle. Cependant, la plupart des systèmes physiques présentent des non linéarités et leurs paramètres sont souvent mal connus et/ou variables dans le temps. Avec le développement considérable des calculateurs numériques, les automaticiens ont de plus en plus adopté de nouvelles approches telles que la commande prédictive, la commande robuste, ainsi que des techniques basées sur l'intelligence artificielle.

Parmi les techniques de l'intelligence artificielle, nous allons nous intéresser à des systèmes neuro-flous. Cette méthodologie utilise une combinaison des réseaux de neurones artificiels avec les approches de la logique floue dans des structures de contrôle des systèmes non linéaires pour résoudre le problème de suivi de trajectoire, tout en assurant la stabilité et robustesse et améliorer les performances de contrôle.

Le choix de système neuro-flou comme une technique de commande est motivé par le fait, qu'elle offre un grand potentiel d'adaptabilité aux systèmes non linéaires.

2. LES NEURO-FLOU

Les systèmes neuro-flous sont des systèmes flous formés par un algorithme d'apprentissage inspiré des réseaux de neurones. La technique d'apprentissage opère en fonction de l'information locale et produit uniquement des changements locaux dans le système flou d'origine [Nauck, 1997].

Les règles floues codées dans le système neuro-flou représentent les échantillons imprécis et peuvent être vues en tant que prototypes imprécis des données d'apprentissage. On peut aussi noter que les systèmes neuro-flous peuvent être utilisés comme des approximateurs universels [Lee, 2001].

2.1. La 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:

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 dernière 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 [Zemalache, 2006].

2.2. STFIS (Self Tuning Fuzzy Inference System)

La méthode présente une complète structurelle avec un système d'inférence flou de type Takagi-Sugeno d'ordre zéro. Ce SIF peut être schématisé sous la forme d'un réseau de quatre couches voir figure.1 :

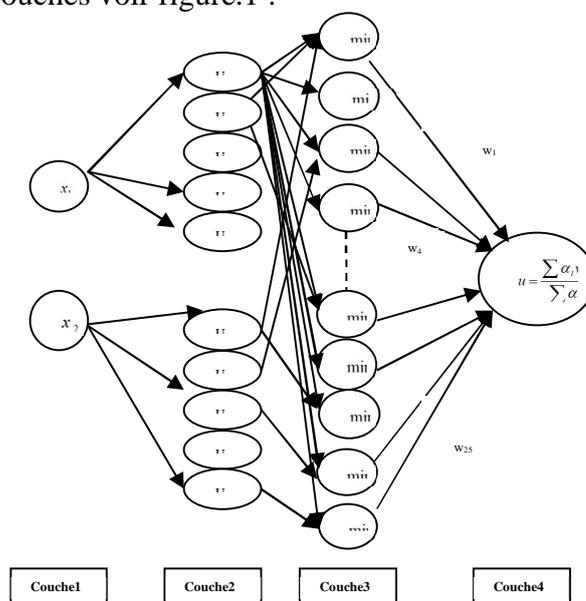


Figure1 : Structure STFIS

2.3. Algorithme d'optimisation

Plusieurs algorithmes sont utilisables pour optimiser les paramètres ajustables du réseau. Dans notre application, nous nous contenterons d'utiliser la méthode de rétropropagation de gradient pour ajuster les poids de la dernière couche du réseau. Le principe général de cette méthode peut être résumé comme : à chaque itération; on modifie les poids de la couche de sortie. Cette modification des poids se fait dans le sens opposé du gradient de la fonction coût. On répète le processus jusqu'à ce que les poids de la couche de sortie aient convergé, c'est-à-dire que l'écart entre la sortie de réseau et la sortie désirée devienne acceptable.

conclusion des règles et les ajuster. L'algorithme de descente de gradient avec la régression des paramètres optimisant seulement la partie conclusion des règles a été adoptée pour satisfaire les objectifs fixes.

$$J = E + \lambda \sum w_i^2 \quad (1)$$

$$E = \frac{1}{2} \varepsilon^2 \quad (2)$$

λ : constante de contrôle d'augmentation des paramètres.

ε : terme d'erreur.

w_i : poids ou paramètre.

A l'aide de l'algorithme de rétropropagation du gradient, les paramètres sont modifiés suivant la formule suivante :

$$w_{ij}^n(k) = w_{ij}^n(k-1) + \Delta w_{ij}^n(k) \quad (3)$$

Aussi nous avons

$$\Delta w_{ij}^n(k) = -\eta \delta_i^n \alpha_j^{n-1} + b \Delta w_{ij}^n(k-1) \quad (4)$$

k : représente les itérations.

w_{ij}^n : poids entre le $i^{\text{ème}}$ neurone de la couche $j^{\text{ème}}$ neurone de la couche $n-1$.

η : gain d'optimisation mesurant la vitesse avec laquelle les poids vont converger vers leur valeur finale.

b : moment, ce paramètre est compris entre 0 et 1 ; l'introduction du terme $b \Delta w_{ij}^n(k-1)$ permet d'éviter les minima locaux en faisant intervenir les variations des poids.

α_j^{n-1} : sortie du neurone de la couche $n-1$.

δ_i^n : terme de dérivée de la fonction coût ($i^{\text{ème}}$ neurone de la couche n).

A l'aide de l'algorithme classique de rétropropagation du gradient, les paramètres sont modifiés suivant la formule suivante :

$$w(k+1) = w(k) + \eta \left(\frac{-\partial J}{\partial w} \right) \quad (5)$$

Cet algorithme intègre facilement l'effet du terme de régression dans la fonction coût. En effet, il suffit de dériver la fonction coût J par rapport à chaque paramètre w_i et en posant $\beta = 2\lambda\eta$ on obtient :

Avec :

β : Coefficient du terme de weight decay ou de régression.

En se limitant dans cette étude pour l'optimisation des conclusions des sorties, ainsi on obtient [Zemalache, 2008] :

$$\Delta w_{ij}^4(k) = -\eta \delta_i^4 \alpha_j^3 + b \Delta w_{ij}^4(k-1) - \alpha_j^3 \beta w_{ij}^4(k-1) / \sum \alpha_j^3 \quad (6)$$

$$\delta_i^4 = y_e - y_d / \sum \alpha_j^3 \quad (7)$$

Où:

y_e : valeur effective de sortie.

y_d : sortie désirée.

3. LA COMMANDE PAR MODE GLISSANT

Les avantages de la commande par mode de glissant sont importants et multiples, comme la haute précision, la bonne stabilité, la simplicité, l'invariance et la robustesse. Ceci lui permet d'être particulièrement adaptée pour les systèmes ayant un modèle imprécis.

La conception de cette commande peut être divisée en trois étapes principales très dépendantes, ces étapes concernent :

- Choix de la surface ;
- Etablissement des conditions d'existence de convergence ;
- Détermination de la loi de commande.

3.1. Choix de la surface

Slotine [Slotine, 1991] a proposé une forme générale qui consiste à définir une fonction scalaire des surfaces de glissement dans le plan de phase dans le but d'assurer la convergence d'une variable d'état x vers sa valeur de consigne x_d cette fonction est donnée par l'équation:

$$S(x) = \left(\frac{\partial}{\partial t} + \lambda_x \right)^{r-1} e(x) \quad (8)$$

Avec :

$e(x)$: l'écart sur la variable à régler

λ_x : une constante positive

r : le degré relatif, qui représente le nombre de fois qu'il faut dériver la surface pour faire apparaître la commande.

L'objectif de la commande est de garder la surface $S(x)$ à zéro. Cette dernière est une équation de la commande différentielle linéaire dont l'unique solution est $e(x)=0$, pour un choix convenable du paramètre λ_x . Ceci revient à un problème de poursuite de trajectoire, ce qui est équivalent à une linéarisation exacte de l'écart, tout en respectant la condition de convergence.

3.2. Condition de convergence

Pour réaliser la loi de la commutation, on peut appliquer la théorie de l'hyper stabilité de Popov pour la synthèse des systèmes à structures variables, ou bien le principe d'optimisation de Pontriagin ou comme dans notre cas la méthode de Lyapunov. Cette dernière est basée sur une fonction dite « fonction de Lyapunov » [Benaïssa, 2006].

La condition de convergence permet au système de converger vers la surface de glissement. Il s'agit alors de formuler une fonction scalaire positive $V(x) > 0$ pour les variables d'états du système qui est définie par la fonction de Lyapunov suivante:

$$V(x) = \frac{1}{2} \cdot S^2(x) \quad (9)$$

Pour que la fonction de Lyapunov décroisse, il suffit de s'assurer que sa dérivée soit négative. Ceci est vérifié par la relation suivante [Slotine, 1991]:

$$\dot{V}(x) = \frac{1}{2} \frac{\partial}{\partial t} (S^2) \leq -\eta \cdot |S| \quad (10)$$

Où $\eta > 0$

3.3. Détermination de la loi de commande

La structure d'un contrôleur comporte deux composantes, une première concernant la linéarisation exacte (u_e) et une deuxième stabilisante (u_c). Cette dernière est très importante dans la technique de commande par mode de glissement, car elle est utilisée pour

éliminer les effets d'imprécision du modèle et de rejeter les perturbations extérieures. La loi de commande par mode glissant est donnée par la formule suivante [Benaïssa, 2006]:

$$u = u_e + u_c \quad (11)$$

La commande équivalente

Si le régime glissant est parfait, le point de fonctionnement évolue sur la surface de glissement $S(x) = 0$ et satisfait $\dot{S}(x) = 0$; puisque la fonction S est alors constante. La commande continue, déduite de $dS/dt = 0$, est appelée commande équivalente u_e .

La commande discontinue

La commande discontinue correspond à la composante discontinue de la forme $-k \cdot \text{sign}(S)$ qui permet au point de fonctionnement d'osciller autour de la surface de glissement avec une amplitude d'autant plus petite et une fréquence d'autant plus élevée que le correcteur est bien calculé (les paramètres de réglage sont alors optimaux) [Lopez, 2000].

La commande de la forme :

$$u = u_e + u_c = u_e - k \cdot \text{sign}(s) \quad (12)$$

$$\text{sign}(S) = \begin{cases} 1 & \text{si } S > 0 \\ -1 & \text{si } S < 0 \end{cases} \quad (13)$$

sign : est la fonction signe.

k : est une constante positive qui représente le gain de la commande discontinue.

La loi de commande glissant est certes robuste vis-à-vis des perturbations paramétriques et externes mais présente quelques inconvénients majeurs :

L'utilisation du terme $\text{sign}(S)$ dans le signal de commutation provoque le phénomène de broutement qui peut exciter les hautes fréquences et détériorer le système commandé. Une méthode qui permet de réduire l'effet du broutement est de remplacer la fonction discontinue par une fonction de saturation, qui consiste à déterminer une bande limite autour de la surface de glissement ainsi assurant le lissage de la commande et le maintient de l'état du système dans cette bande. La loi de commande devient alors :

$$u = u_e - k \cdot \text{sat} \left(\frac{s}{\varepsilon} \right) \quad (14)$$

$$\text{sat} (s / \varepsilon) = \begin{cases} \text{sign} (s) & \text{si } |s| > \phi \\ s / \varepsilon & \text{si } |s| \leq \phi \end{cases} \quad (15)$$

Où la largeur de la bande limite est égale à 2ϕ , et peut être définie selon la précision (ε) désirée, sachant que :

$$\varepsilon = \frac{\phi}{\lambda^{n-1}}$$

4. LA COMMANDE HYBRIDE STFIS ET MODE GLISSANT

Dans cette section, on propose une commande permettant de résoudre les problèmes du mode glissant classique cités précédemment c'est à dire en cas où la dynamique est inconnue et garantir de bonnes performances de poursuite même en présence de perturbations externes. On utilisera un réseau STFIS pour approximer la dynamique supposée totalement inconnue. La commande hybride présentée dans la figure 2 contient deux éléments, le régulateur STFIS à deux entrées caractérisant la surface et sa variation et une sortie qui caractérise la commande équivalente, le deuxième est la commande discontinue. La somme des deux commandes, forme la commande globale qui stabilise le système.

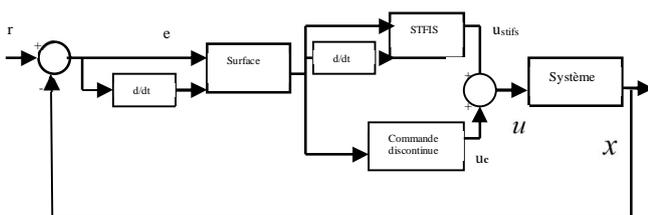


Figure 2 : la commande hybride.

La loi de commande proposée est :

$$u = u_{STFIS} + u_c = u_{STFIS} - k \cdot \text{sat}(s) \quad (16)$$

u_{STFIS} : la commande générée par le contrôleur STFIS;

u_c : la commande discontinue.

La surface de glissement utilisée est $s = \dot{e} + \lambda e$ avec λ est un scalaire qui représente la pente de la surface

4.1. RESULTAT DE SIMULATION

Les simulations ont été réalisées sur le modèle dynamique du pendule inversé (17).

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = \frac{(m_c + m)g \sin x_1 - mlx_2^2 \cos x_1 \sin x_1}{l(4/3(m_c + m) - m \cos^2 x_1)} \\ + \frac{\cos x_1}{l(4/3(m_c + m) - m \cos^2 x_1)} u(t) + d(t) \end{cases} \quad (17)$$

$$\begin{cases} \dot{x}_3 = x_4 \\ \dot{x}_4 = \frac{4/3 mlx_2^2 \sin x_1 + mg \cos x_1 \sin x_1}{4/3(m_c + m) - m \cos^2 x_1} \\ + \frac{4}{3 \cdot (4/3(m_c + m) - m \cos^2 x_1)} u(t) + d(t) \end{cases}$$

Où x_1 position angulaire, x_2 vitesse angulaire, x_3 position de chariot, x_4 vitesse de chariot, $g = 9.8m/s^2$, m_c est masse de la chariot et $m_c = 1kg$, m est la masse de pendule $m = 0.1kg$, $l = 0.5m$ est longueur du pendule centre de masse et $d(t) = 0.2$ est un bruit.

L'architecture de la commande appliquée est présentée dans la figure 2, utilisés pour commander l'angle du pendule (θ) de système pendule inversé.

On définit cinq fonctions d'appartenance pour les entrées de STFIS $s(t)$, $\dot{s}(t)$, grand négative (GN), petit négative (PN), zéro (Z), petit positive (PP), grand positive (GP) normalisé dans l'univers [-1 1] voir la figure 3.

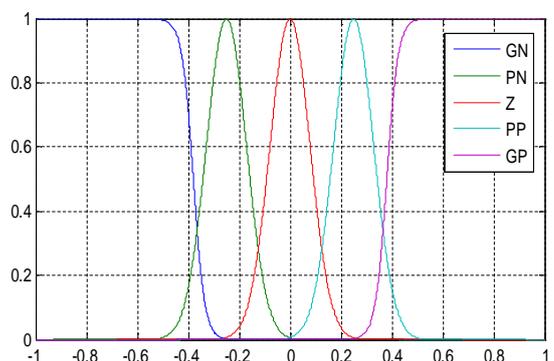


Figure 3 : les fonctions d'appartenance.

On a injecté le bruit dans le système pendule inversé dès le départ. Après des différents essais de simulation, les valeurs des gains ont donné de bons résultats sont présentées dans le tableau qui suivra:

λ	k	β	η	b
10	50	0.0006	0.3	0.9

Table 1 : les paramètres optimisés.

L'erreur entre la référence et la position angulaire de pendule inversé est présentée dans figure4.

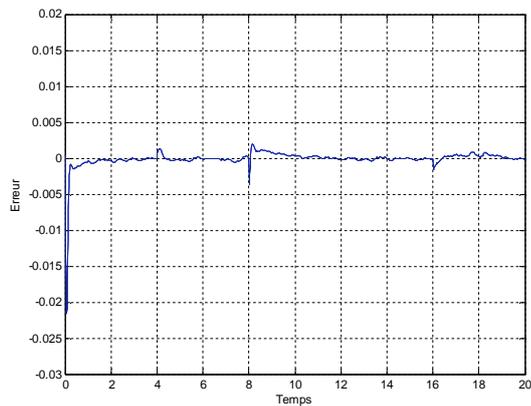


Figure4 : Erreur entre la référence et position angulaire.

Pour évaluer des teste de robustesse, nous avons effectué des variations paramétriques. Les figures5, 6 présentent l'erreur entre la référence et la position angulaire de la variation de masse $m = 0.5kg$ et variation de la longueur de pendule $l = 1m$.

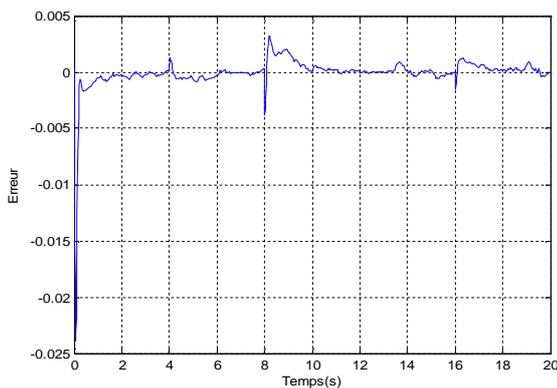


Figure 5 : Erreur de variation de masse de pendule.

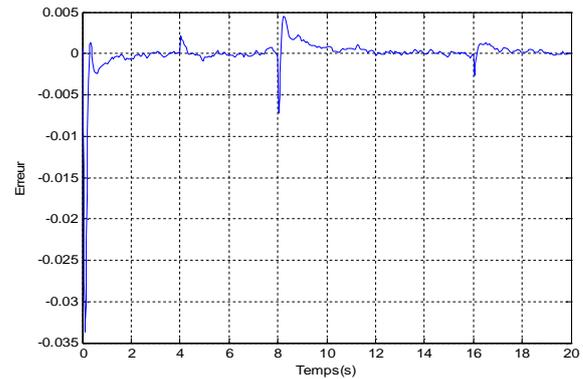


Figure 6 : Erreur de variation de longueur de pendule.

A travers les résultats de simulation effectuée sur le modèle non linéaire du système, nous pouvons constater que le régulateur STFIS synthétisé a généré la composante équivalente et la commande hybride a permis de stabiliser le système aussi bien en régulation qu'on poursuite de trajectoire. Malgré les différentes perturbations (variations paramétriques), la stabilisation reste toujours robuste.

5. CONCLUSION

Dans ce papier, nous avons essayé de résoudre le problème du contrôle des systèmes non linéaire, faisant intervenir les techniques d'intelligence artificielle tel que les systèmes neuro-flous avec la commande mode glissant. Nous avons présenté l'apport principal de cette étude qui résidait dans le développement de la technique neuro-flou. En donnant les notions de base de système neuro-flou ainsi l'approche STFIS. La deuxième partie de ce travail a été réservée à la présentation de la commande par mode glissant. Dans la troisième, nous avons présenté une commande hybride (mode glissant et STFIS) pour le contrôle du pendule inversé. Les dynamiques inconnues sont approximées par un réseau STFIS. Nous avons étudié la robustesse de la commande hybride mode glissant et neuro-flou en présence de perturbation et le changement de paramétrique. Les résultats de simulation ont montré un comportement satisfaisant du système.

Bibliographie

- [Benaissa, 2006] Benaissa, A., Fellah, M.K. "Commande par mode de glissement d'un convertisseur AC/DC avec correction du facteur de puissance". ACTA Electrotehnica, Volume 47, Numéro 2, Université de Djilali Liabes Sidi Bel Abbes, Algérie (2006).
- [Lee, 2001] Lee, G., Wang, J.S.: "Efficient Neuro-Fuzzy Control Systems for Autonomous Underwater Vehicle Control". IEEE International Conference on Robotics and Automation, Seoul, Coree (2001).
- [Lopez, 2000] Lopez, P., Saïd Nouri, A.: "Théorie élémentaire et pratique de la commande par les régimes glissants". Library of Congress Control Number 2005938497 (2000).
- [Nauck, 1997] Nauck, D., Kruse, R.: "What are Neuro-Fuzzy Classifiers?". Seventh International Fuzzy Systems Association World Congress IFSA'97, Vol. IV, pp. 228-233, Académie de Prague (1997).
- [Slotine, 1991] Slotine, J., Li, W. "Applied Nonlinear Control". Englewood Cliffs, NJ: Prentice-Hall, (1991)
- [Zemalache, 2006] Zemalache Meguenni, K.: "Commande d'un système sous- actionne : Application a un drone a Quatre Helices". Thèse de doctorat en Génie Informatique, Automatique, Université d'Evry Val d'Essonne, Décembre (2006).
- [Zemalache, 2008] Zemalache, K., Beji, L., Maaref, H.: "Control of Drone: Study and Analysis of the Robostness". Journal of Automation, Mobile Robotics and Intelligent Systems (2008).