

# CONTROLEUR FLOU DYNAMIQUE

P.Goureau, D.Sirier

***E.N.S.E.A. - E.C.S.***

6, avenue du Ponceau  
95014 Cergy-Pontoise FRANCE  
Tél : (1) 30 73 66 66  
Fax : (1) 30 73 66 67  
Mail : goureau@ensea.fr

A.Bigand

***Université du Littoral - L.I.S.I.R.***

B.P. 649, 6228 CALAIS Cedex

## 1. Introduction.

Dans le domaine de l'asservissement, les contrôleurs flous peuvent concurrencer les correcteurs classiques. En effet, ils offrent de bonnes caractéristiques en termes de temps de réponse, dépassement et surtout robustesse dans certaines applications. De plus, ils n'ont besoin que d'une connaissance sommaire du système à contrôler, qui s'exprime sous forme d'un tableau (moteur d'inférence) où l'on enregistre le comportement du correcteur. Le temps de mise au point est donc beaucoup plus court bien qu'il soit nécessaire d'utiliser soit un micro-contrôleur, soit un composant dédié. Dans un souci d'optimiser encore la qualité de l'asservissement, on se propose de trouver un moyen de rendre le contrôleur flou dynamique. Ainsi, la base de connaissances du contrôleur devrait se modifier petit à petit en suivant les variations des paramètres du système et assurer un asservissement optimal. La figure 1 représente le schéma du système asservi utilisé.

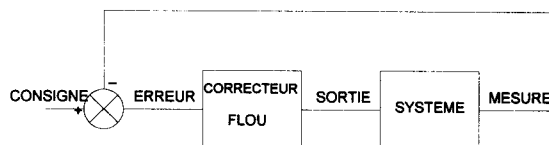


figure 1

## 2. Contrôleur de TAKAGI et SUGENO.

Il se différencie du contrôleur flou de type MAMDANI par l'expression des valeurs de la sortie de chaque règle. En effet, au lieu d'attribuer à l'aide de chaque règle une valeur floue à la sortie, on calcule directement une valeur analogique à l'aide d'un polynôme fonction des entrées, comme par exemple :

$$\text{SI } A \text{ est } E^1 \text{ et } B \text{ est } F^1 \text{ alors } Y^1 = C_0^1 + C_A^1 * A + C_B^1 * B$$

où A et B sont les entrées,  $Y^1$  la sortie de la règle l,  $E^1$  et  $F^1$  deux ensembles flous.

On peut résumer l'ensemble de ces règles sous forme d'un tableau représentant le moteur d'inférence où l'on place les valeurs des différents coefficients. Dans le cas où les entrées sont l'erreur et sa variation, les coefficients de ce polynôme représentent alors le gain proportionnel et le gain dérivé auxquels s'ajoute une constante  $C_0^1$  appelée parfois talon. Pour calculer la valeur de la sortie du correcteur, il faut faire le barycentre des valeurs de tous les polynômes  $Y^1$  pondérés par le degré d'activation de la condition associée  $W^1$ . Dans le cas

classique où les ensembles flous sont choisis tels que  $\sum_i \mu_i(A) = 1$ , où  $i$  représente tous les ensembles flous relatifs à la variable  $A$ , et que les mêmes conditions sont respectées pour  $B$ , on démontre alors que la sortie du contrôleur peut s'écrire :  $S = \sum_l W^l \cdot Y^l$ .

Ce contrôleur a l'avantage de donner directement une valeur analogique pour la sortie et nous affranchit donc du problème du choix de la méthode de déffuzification. Cependant, il est beaucoup moins facile de déterminer les coefficients de ces polynômes que de déterminer le contenu d'un moteur d'inférence (c'est d'ailleurs l'avantage du contrôleur flou type MAMDANI).

On peut en déduire un schéma mettant en évidence la structure neuronale de ce contrôleur (figure 2):

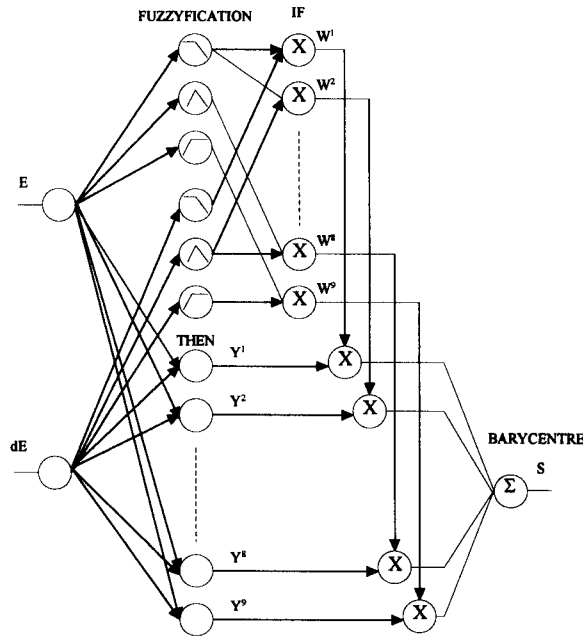


figure 2

### 3. Mise en équation.

Chaque couche de ce réseau de neurones peut se mettre en équations. Les entrées forment un vecteur colonne noté  $X = [e ; de]$  où «  $e$  » représente l'erreur et «  $de$  » sa dérivée

Pour la Fuzzyfication, deux fonctions  $\mu_e(e)$  et  $\mu_{de}(de)$  vont donner respectivement les degrés d'appartenance de l'entrée et de la dérivée de l'entrée sous forme d'un vecteur ligne. Il faudra donc définir les ensembles flous de chaque variable. On choisira de façon classique des ensembles de forme trapézoïdale :

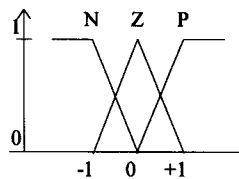


figure 3

Dans ce cas, le vecteur s'écrit par exemple :

$$\mu_e(e) = [\mu_N(e) \quad \mu_Z(e) \quad \mu_P(e)]$$

La Couche « IF » est celle où vont être calculés tous les  $W^l$ . Ces vecteurs  $W^l$  correspondent aux produits de toutes les combinaisons entre un élément de  $\mu_e(e)$  et  $\mu_{de}(de)$ . On remarque que le produit matriciel de la transposée du vecteur ligne  $\mu_e(e)$  par le vecteur ligne  $\mu_{de}(de)$  donne la matrice  $\Lambda$  suivante :

$$\Lambda = \begin{bmatrix} \mu_N(e) \cdot \mu_M(de) & \mu_N(e) \cdot \mu_Z(de) & \mu_N(e) \cdot \mu_P(de) \\ \mu_Z(e) \cdot \mu_M(de) & \mu_Z(e) \cdot \mu_Z(de) & \mu_Z(e) \cdot \mu_P(de) \\ \mu_P(e) \cdot \mu_M(de) & \mu_P(e) \cdot \mu_Z(de) & \mu_P(e) \cdot \mu_P(de) \end{bmatrix}$$

Le nombre de règles est donc égal au produit du nombre d'ensembles flous définissant l'erreur par le nombre d'ensembles flous définissant la dérivée de l'erreur. Le vecteur  $W^1$  sera ensuite divisé par le terme  $\sum_i W^1$  pour donner le vecteur  $W^1$ .

La couche « THEN » effectue le calcul de tous les  $Y^1$ . Pour cela on a besoin de trois vecteurs colonnes contenant les coefficients  $C_0$ ,  $C_e$  et  $C_{de}$ . Ces vecteurs auront donc autant de composantes qu'il y aura de règles.

La couche « Barycentre » permet d'évaluer la valeur de la sortie du contrôleur. D'après l'équation permettant le calcul de  $S$ , on en déduit que la sortie s'exprimera simplement par le produit du vecteur colonne  $Y^1$  par le vecteur ligne  $W^1$  donc :  $S = Y^1 * W^1$ .

#### 4. Apprentissage par rétropropagation.

On se propose de rendre le contrôleur adaptatif afin que celui-ci règle de lui-même ses coefficients. Compte tenu de la structure neuronale de ce type de contrôleur, on utilise une méthode classique d'apprentissage des réseaux de neurones : la rétropropagation. Cette méthode permet de trouver un minimum d'une fonction de plusieurs paramètres. Elle utilise le gradient de cette fonction afin de modifier les paramètres et de minimiser celle-ci. Ainsi si  $F(x)$  est une fonction qui dépend d'un paramètre  $A$ , on modifiera ce paramètre en utilisant

l'équation suivante donnant l'incrément de  $A$  : 
$$dA = -Lr \cdot \frac{\partial F(x)}{\partial A}$$

Le coefficient  $Lr$  (Learning Rate ou taux d'apprentissage) permet de gérer la vitesse de convergence. Une valeur trop petite rend la convergence trop lente, une valeur trop grande risque de la rendre instable. Il faudra donc trouver une valeur intermédiaire. Cette valeur sera déterminée par simulation. Dans notre cas, on se propose de rendre le contenu des matrices  $C_0$ ,  $C_e$  et  $C_{de}$  adaptatif. Etant donné que le but d'un asservissement est de faire en sorte que la sortie du système colle au mieux avec la consigne, on choisit l'erreur quadratique ( $e^2$ ) comme critère à minimiser.

Dans un cas générique, on considère que la sortie est fonction d'un coefficient  $A$  :

$$\frac{\partial (\text{erreur}^2)}{\partial A} = \frac{\partial (\text{erreur}^2)}{\text{erreur}} \cdot \frac{\partial \text{erreur}}{\partial \text{mesure}} \cdot \frac{\partial \text{mesure}}{\partial \text{sortie}} \cdot \frac{\partial \text{sortie}}{\partial A}$$

Dans cette expression, seule  $\frac{\partial \text{mesure}}{\partial \text{sortie}}$  est inconnue. Cependant, on sait que ce terme est toujours positif ou nul et donc qu'il ne modifiera pas le sens de la rétropropagation. De plus le coefficient  $Lr$  va multiplier tout ceci et va donc absorber en moyenne ce terme. On en déduit alors :

$$[dC_0] = 2 \cdot Lr \cdot \text{erreur} \cdot W^t \quad (W^t = \text{transposé de } W^1)$$

$$[dC_e] = e \cdot [dC_0]$$

$$[dC_{de}] = de \cdot [dC_0]$$

#### 5. Premières simulations.

On utilise pour cela le logiciel SIMULINK permettant de mettre en place l'environnement dans lequel nous allons faire travailler le correcteur. Les figures suivante précise les schémas d'asservissement utilisés :

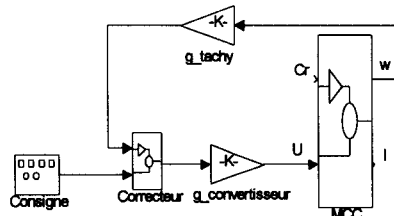


figure 4 : Schéma global de l'asservissement

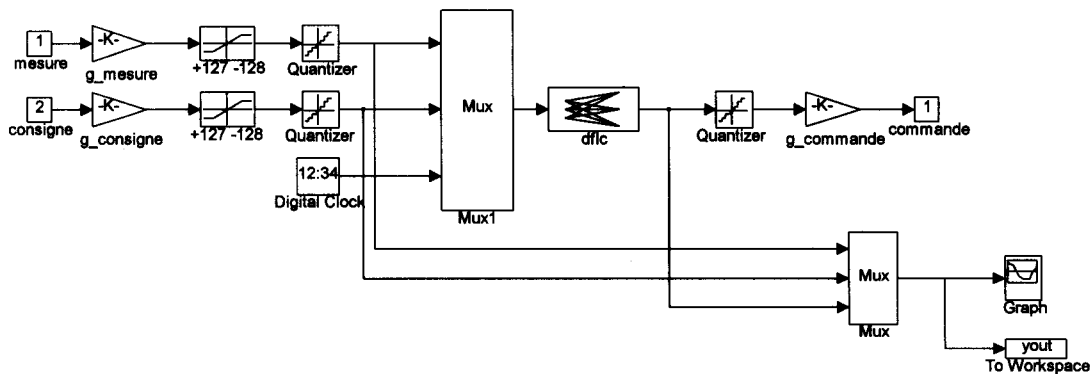


figure 5 : Schéma du correcteur flou

La simulation prend en compte la quantification due au CAN ainsi que les saturations réelles que l'on peut rencontrer dans un tel asservissement numérique. De plus les amplitudes des grandeurs analogiques et numériques sont respectées en tout point. Le correcteur commence par bien réagir, il se sature tant que l'erreur est grande puis diminue sa sortie jusqu'à ce que le régime permanent soit atteint. Par la suite, le temps pendant lequel le correcteur reste saturé augmente petit à petit et rend la suite de la réponse de plus en plus oscillante. Ceci est dû à l'augmentation des gains qui ne peut être contrôlé par la rétropropagation étant donné qu'il y a une saturation. La rétropropagation tant alors à sursaturer le correcteur, les gains augmentent et l'asservissement devient de plus en plus instable.

Ceci se voit très bien sur les plans de contrôle calculés à  $T=2$  sec (figure 6) et  $T=4$  sec (figure 7) et donnant la sortie du correcteur en fonction de l'erreur et de sa dérivée. On remarque que la zone saturée est de plus en plus importante et que le gain au voisinage de l'erreur nulle est de plus en plus important.

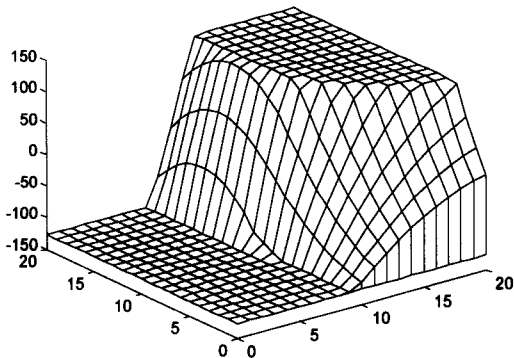


figure 6

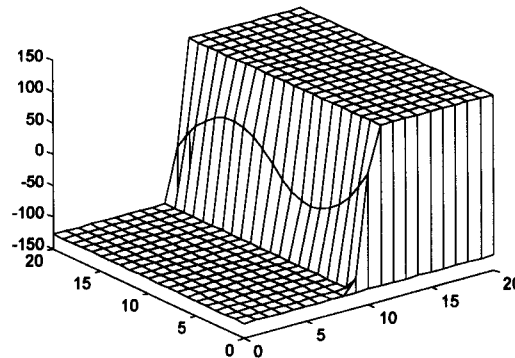


figure 7

## 6. Modification de la structure.

Pour améliorer les résultats précédents, on se propose de mettre en place un modèle de réponse du système asservi. En effet, il est absurde de prendre en compte une consigne de forme carrée puisque que l'on sait que l'asservissement ne pourra pas la suivre. Ce modèle de réponse sera le résultat du passage de la consigne à travers un filtre du deuxième ordre qui permettra d'approcher la réponse réelle que l'on peut attendre d'un tel asservissement. Ce sera donc l'erreur entre ce modèle de réponse et la réponse réelle de l'asservissement qui sera rétropropagée. Ceci ne limitera en rien le phénomène de sursaturation progressive qui entraîne l'instabilité de l'ensemble. Pour résoudre ce problème, on va mettre en place un identificateur qui va modéliser le système. Cet identificateur va être capable de faire fonctionner le correcteur en tache de fond, puisque l'on pourra prédire la sortie du système grâce à l'estimateur et effectuer plusieurs rétropropagations en utilisant comme critère la différence entre la sortie de l'estimateur et du modèle de la réponse.

On s'attend donc à une convergence plus rapide de la sortie du système avec la nouvelle consigne sortant du modèle de la réponse. De plus, on supervisera l'action de la rétropropagation. Lorsque celle-ci tendra à faire sursaturer le correcteur, le cycle d'apprentissage sera interrompu. Le schéma d'ensemble de la nouvelle structure est alors le suivant (figure 8):

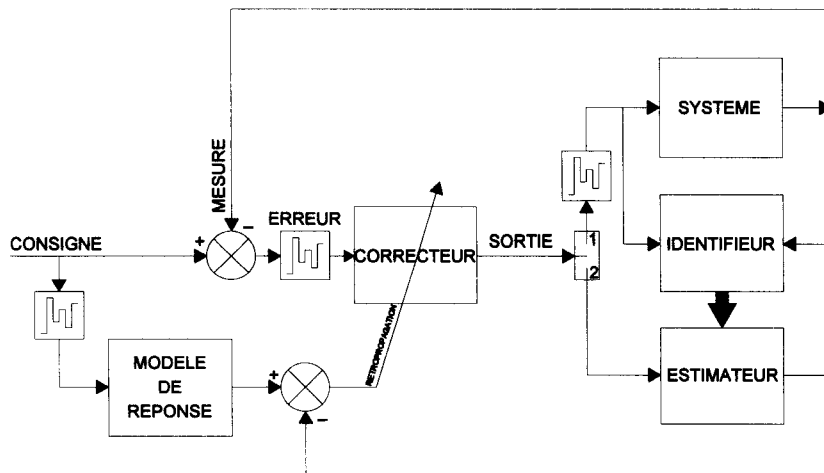


figure 8

L'identificateur a pour but d'approcher le comportement du système. Dans le cas d'une machine à courant continu (M.C.C.), on choisit un modèle du deuxième ordre du type : 
$$\frac{Y}{X} = \frac{az^{-1} + bz^{-2}}{1 + cz^{-1} + dz^{-2}}$$

On doit rechercher les valeurs de a, b, c et d qui permettent d'approcher au mieux le comportement réel du système. On utilise pour cela un certain nombre d'échantillons k de l'entrée X et de la sortie Y qui permettront de calculer une estimée de la sortie du système Y' et l'on cherchera à minimiser  $\sum_{k=2} (Y - Y')^2$ . On peut écrire le problème sous forme matricielle :  $F \cdot A - Y = \epsilon$  où  $\epsilon$  est la matrice colonne à minimiser. On en déduit, après calcul, l'écriture de A minimisant  $\epsilon$  :

$$A = (F^T \cdot F)^{-1} \cdot F^T \cdot Y$$
 où  $F^T \cdot F$  est une matrice 3x3 inversible. Ceci ne sera possible d'un point de vue physique que lorsque le système est dans une phase transitoire apportant beaucoup d'information sur son comportement dynamique. Lorsque le système approche du régime permanent, les échantillons n'apportent plus d'information et doivent être éliminés. On peut s'intéresser à la dérivée de la sortie du système, qui lorsqu'elle sera inférieure en valeur absolue à un certain seuil, arrêtera l'accumulation des échantillons. De plus, le nombre d'échantillons doit être limité au nombre nécessaire uniquement pour enregistrer le régime transitoire. On rajoute pour cela une condition sur le déterminant de la matrice  $F^T \cdot F$  qui, lorsqu'il est trop petit, ne permet plus de calculer correctement le modèle. Les valeurs limites de la dérivée et du déterminant seront pré-déterminées par simulation. Pour cela, on étudie le comportement de l'identificateur lorsqu'il travaille avec une MCC et l'on enregistre les valeurs limites lorsque la réponse du modèle trouvée commence à s'écarter de la réponse réelle de la MCC.

## **7. Simulations du système complet.**

Voici les résultats de la simulation ainsi que le plan de contrôle, les paramètres de la rétropropagation étant les suivants :  $L_r = 2 \cdot 10^{-5}$  et NBEIT = 20 (nombre d'itérations).

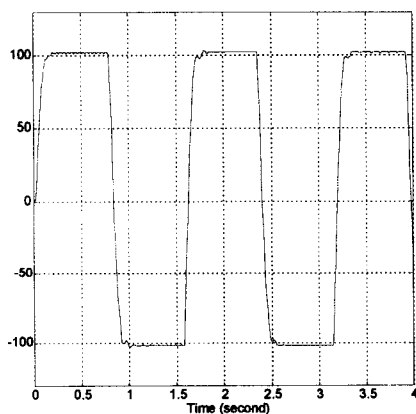


figure 9 : Sortie du système

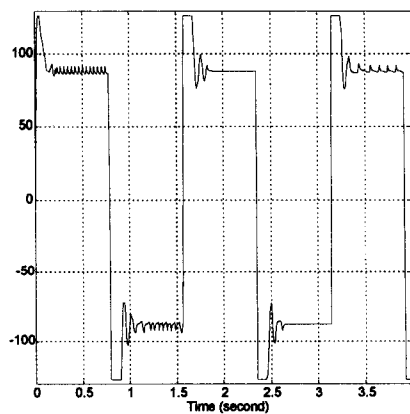


figure 10 : Sortie du contrôleur

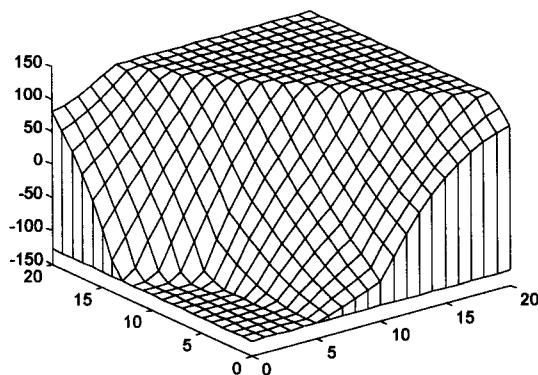


figure 11 : Plan de contrôle

On remarque que l'asservissement est de bonne qualité. La réponse suit bien la consigne imposée par le modèle de réponse. La sortie du contrôleur est peu oscillante (oscillation due uniquement à la discrétisation). On note un dépassement inférieur à 2% et une vitesse de réponse quasi optimale car la réponse à 5% est obtenue pendant la phase de saturation. On voit un certain déséquilibre sur le plan de contrôle. Celui-ci devrait posséder une symétrie centrale compte tenu du fait que le correcteur doit réagir de la même manière au signe près à un échelon positif ou à un échelon négatif. Ceci est le résultat d'une rétropropagation qui a tendance à localiser son action sur une partie du plan de contrôle suivant le signe de l'échelon.

Le contrôleur étant un équivalent proportionnel dérivé, on devrait alors avoir une erreur statique non-nulle. Cependant dans le contrôleur flou de TAKAGI et SUGENO, on voit apparaître des coefficients  $C_0$  qui s'ajustent pour permettre à cette erreur statique de diminuer progressivement.

## **8. Conclusion.**

De nombreux autres essais où les paramètres du système ont été rendus variables (inertie, couple de charge,...) ont permis de tirer des conclusions identiques en termes de performances et de stabilité. Il apparaît donc que cette technique de correction, appliquée ici à un système relativement simple, outre le fait qu'elle se prête bien à la régulation de systèmes qui pourraient être plus complexes (et en particulier non linéaires), offre l'avantage d'une mise en oeuvre très facile avec une auto-adaptation à la variation des paramètres du système. Ce type de correcteur est donc très robuste tout en gardant une excellente stabilité quel que soit le point de fonctionnement. Ce dernier point fait l'objet de nos études actuelles où, par l'utilisation de la méthode de Ljapunov, on arrive à trouver un domaine de stabilité important.

L'implantation matérielle de ces algorithmes sur un ordinateur de type D.S.P. permettra de corroborer ces résultats de simulation. De plus, ce type de régulateur sera mis en oeuvre sur des régulations de vitesse de machines à courant alternatif où les besoins en identificateur sont encore plus importants.

**9. Bibliographic.**

- [1] « Fuzzy identification of systems and its application to modelling and control »  
T. Takagi and M. Sugeno
- [2] « Industrial Applications of Fuzzy Control »  
M.Sugeno, 1985
- [3] « Design of neuromorphic fuzzy controllers »  
D.T. Pham, D. Karaboga
- [4] « Fuzzy logic in control system : fuzzy logic controller »  
Chuen Chien LEE
- [5] « A direct adaptive method for faster backpropagation learning : the RPROP algorithm »  
M. Riedmiller, H. Braun
- [6] « Development of Fuzzy Algorithms for Servo System », IEEE Control System Magazine, pp65-72,  
April 1989
- [7] « Robust Design of rule-based fuzzy controllers »  
A.Ollero, J.Arakil, A.Garcia, Fuzzy sets and systems 70, 1995
- [8] « Stability of fuzzy control : a Ljapunov approach »  
Y.Y.Chen, IEE Proc. of Annual Conference on Systems Man and Cybernetics, vol 3, 1978
- [9] « Stability analysis of non linear fuzzy PI control system »  
G.Chen, H.Ying, IFIS'93, pp128-133
- [10] « A fuzzy controller improving a linear model following controller for motor drives »  
Y.S.Kung, C.M.Liaw, IEEE transactions on fuzzy systems, vol 2, n°3, August 1994