

Etude de performance des réseaux de neurones à 3 couches pour le pilotage des robots industriels apprentissage non supervisé

Rakotondrainibe F. E.¹, Andriamanohisoa H. Z.²

Laboratoire de Recherche – Sciences Cognitives (LR-SCA)

Equipe d’Accueil Doctorale en Sciences Cognitives et Application (EAD-SCA)
Ecole Doctorale en Sciences et Techniques de l’Ingénierie et de l’Innovation (ED-STII)

Université d’Antananarivo BP 1500, Ankatso – Antananarivo 101 – Madagascar

¹barhzsys@hotmail.fr,²aherizo@yahoo.com

Résumé :

L’objectif de cette article est de démontrer que le pilotage des robots industriels 6 axes est possible, même si ces robots sont mathématiquement solvables.

Les simulations montrent qu’un réseau de neurone à 3 couches est capable de résoudre les problèmes.

La fonction d’activation sigmoïde est la plus performante

Mots clés : Réseau de neurones, MLP, Robot 6 axes, Fonction d’activation

Abstract

The objective of this article is to demonstrate that the control of 6-axis industrial robots is possible, even if these robots are mathematically solvable.

The simulations show that a 3-layer neural network is capable of solving problems.

The sigmoid activation function is the most efficient

Keywords: Neural network, MLP, 6-axis robot, Activation function

1. INTRODUCTION

En robotique industriel le pilotage reste une des problèmes majeurs à cause des équations non linéaire et trigonométriques.

Les solutions mathématiques existent mais l’implémentation réel restent difficiles gourmand en puissances de calcul et requière donc des machines très performantes, des logiciels des simulations et de commandes très sophistiqué, d ou le prix exorbitant.

Cette étude essaye de résoudre ce problème en utilisant les réseaux de neurones ainsi que le choix technologique adéquat.

2. Le perceptron simple

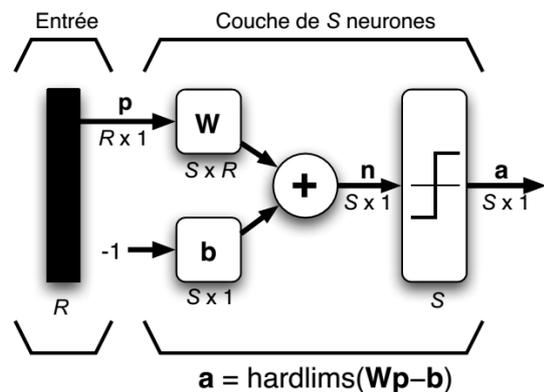


Figure 1 Le perceptron simple

il s’agit d’une seule couche de S neurones totalement connectée figure 1 [8] sur un

vecteur \mathbf{p} de R entrées. La matrice $\mathbf{W} = [1\mathbf{w} \ 2\mathbf{w} \ \dots \ S\mathbf{w}]T$ de dimension $S \times R$ représente l'ensemble des poids de la couche, avec les vecteur-rangées $i\mathbf{w}$ (dimension $R \times 1$) représentant les R poids des connexions reliant le neurone i avec ses entrées. Le vecteur \mathbf{b} (dimension $S \times 1$) désigne l'ensemble des S biais de la couche.

Les niveaux d'activations

$$n = \sum W_p - b \quad (01)$$

la couche servent d'argument a la fonction d'activation qui applique un seuil au niveau 0 pour produire le vecteur des sorties $\mathbf{a} = [a_1 \ a_2 \ \dots \ a_S]T$, ou :

$$a_i = f(n) \quad (02)$$

Dans notre études comparatives comparant les différentes fonctions d'activations les plus couramment utilisées tels que :

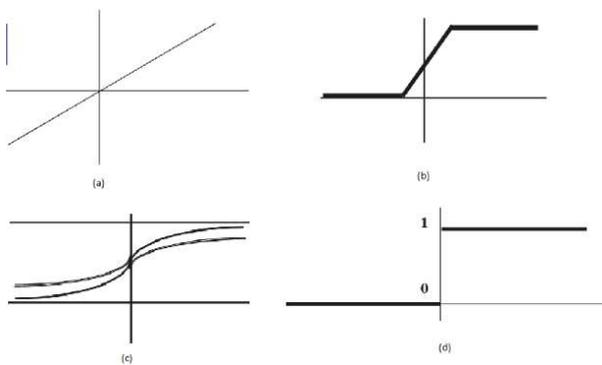


Figure 2: Fonction d'activation

- a) Linéaire $f(x)=x$
- b) Par morceau $F(x) = \begin{cases} 0 & \text{si } x \in]-\infty, a[\\ cx + d & \text{si } x \in [a, b] \\ 1 & \text{ailleurs} \end{cases}$
- c) Sigmoïde $F(x) = \frac{1}{1+e^{x+T}}$
- d) Signe $f(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases}$

3. Réseau multicouche

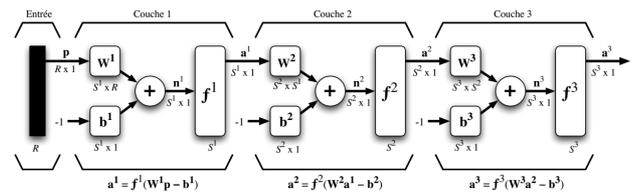


Figure 3: Réseaux multicouches

Les réseaux multicouches figure 3 [8] ne pouvaient résoudre que des problèmes de classification linéairement séparables. Les réseaux multicouches permettent de lever cette limitation en robotique l'utilisation fréquente des fonctions trigonométrique engendre des fonctions non linéaire. On peut même démontrer qu'avec un réseau de trois couches (deux couches cachées + une couche de sortie), on peut construire des frontières de décision de complexité quelconque, ouvertes ou fermées, concaves ou convexes, à condition d'employer une fonction de transfert non linéaire et de disposer de suffisamment de neurones sur les couches cachées.

4. Pilotage des robots industriels

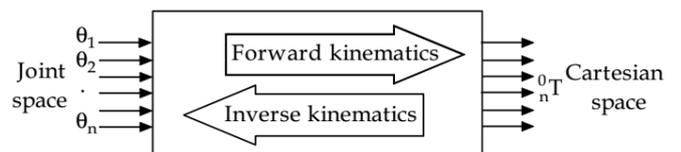


Figure 4: Modèle géométrique d'un robot industriel

Dans le cas pratiques pour le pilotage automatique les angles (θ_1 à θ_n) sont préprogrammés dans le calculateur des robots, mais dans notre recherche d'autonomies décisionnelles les coordonnées dans une espace cartésienne sont fournies par des capteurs, caméras ou autres organes sensoriel.

Comme les solutions mathématiques (cas des robots 6 axes groupe 1.1) de ces

problèmes géométriques existe de nos jours même si le du pilotage est toujours difficiles à maitre en pratique (mathématiquement mais surtout techniquement) à cause des autres facteurs tels que l'inertie, les tolérances de fabrication. Il est donc faciles d'évaluer l'erreur de calcul de nos réseaux ainsi que de mettre en œuvre un apprentissage non supervisé.

Architecture de pilotage d'un robot 6 axes (groupe 1.1) avec des réseaux de neurones

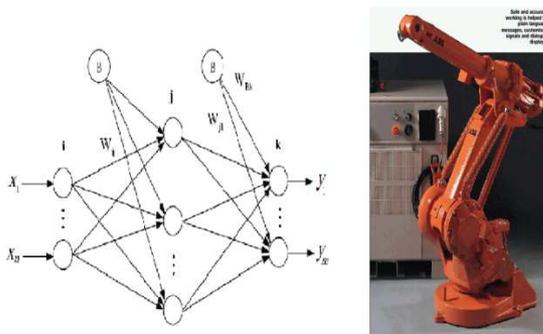


Figure 5: Schémas synoptique de l'installation

Dans notre études considérons un réseau a 6 entrées X (3 vecteur position « x y z » et 3 vecteur d'orientation « Vx Vy Vz »), et 6 sorties Y (les valeurs des 6 axes du robot).

5. MLP 6axis

Vue le cout des logiciels professionnels, la performance et la complexité d'utilisation des logiciel libre on a décidé de crée MLP 6axis. C'est un logiciel spécialement conçu pour la simulation et le pilotage des robots 6 axes groupe 1.1 développé dans notre laboratoire de recherche.

Il intègre à l'intérieur des algorithmes standard utilisé en réseau de neurones et de robotique tels que :

Solution mathématique des MGD (Forward kinematics)

Calcul des sorties de chaque réseau

Calcul des erreurs

Mise à jour des poids en utilisant l'algorithme de retro-propagation

a) Algorithme de MLP 6axis

Début

Etape 1 :

Etape 2 : répéter n fois

Etape 3 : Choisir au hasard les 6 entrées

Etape 4 : Calculer les sorties

Etape 5 : Calcul du MGD

Etape 6 : Calcul des erreurs

Etape 7 : Mise à jour des poids

Etape 8 : Mise à jour des poids en utilisant l'algorithme de retro-propagation

Fin

b) Voici son interface graphique

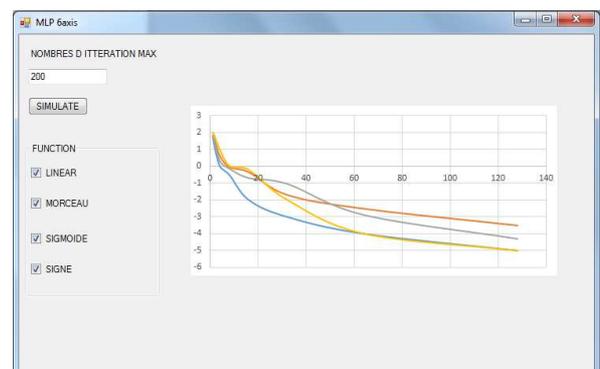


Figure 6: GUI du MLP6axis

Legende : Rouge signe

Bleu linéaire

Morceau

Sigmoïde

6. Interprétation

Toutes les fonctions convergent vers 0

Les résultats montrent que l'erreur diminue de façon exponentielle mais les fonctions sigmoïde et en morceau s'avère plus performant que les deux autres.

La fonction sigmoïde est plus rapide que la fonction en morceau et attein plus rapidement une valeur acceptable.

7. CONCLUSION

Vue la complexité du pilotage d'un robot 6 axes groupes 1.1, les réseaux de neurones est une des meilleures solutions proposé car il est capable de résoudre des problèmes non linéaire et dans le cas des robots des fonctions trigonométrique, ainsi que l'existence de l'inertie dans le cas pratiques

La simulation nous montre que l'utilisation de la fonction d'activation sigmoïde s'avère plus approprié.

Une autre étude fera l'objet de l'amélioration de la fonction sigmoïde en modifiant ces paramètres principaux.

8. References:

[1] Hasan, A.T.; Hamouda, A.M.S.; Ismail, N. & Al-Assadi, H.M.A.A.(2006). An adaptive learning algorithm to solve the inverse kinematics problem of a 6 D.O.F serial robot manipulator. *Journal of Advances in Engineering Software*, Vol.37, pp. 432-438.
 [2] Haykin S. (1994). *Neural Networks*. A Comprehensive Foundation. New York: Macmillan.
 Hornik, K. (1991). Approximation capabilities of multi-layer feed forward networks. *IEEE*

Trans. Neural Networks, Vol.4, No.2, pp. 251-257.

[3] Karilk, B. & Aydin, S. (2000). An improved approach to the solution of inverse kinematics problems for robot manipulators. *Journal of Engineering applications of artificial intelligence*, Vol.13, pp.159-164.
 [4] Köker, R. (2005). Reliability-based approach to the inverse kinematics solution of robots using Elman's networks. *Engineering Applications of Artificial Intelligence*, Vol.18, pp. 685-693.
 [5] Köker, R.; Öz, C.; Çakar, T. & Ekiz, H. (2004). A study of neural network based inverse kinematics solution for a three-joint robot. *Journal of Robotics and Autonomous Systems*, Vol.49, pp. 227-234.
 [6] Kuroe, Y.; Nakai, Y. & Mori, T. (1994). A new Neural Network Learning on Inverse Kinematics of Robot Manipulators, *International Conference on Neural Networks*, IEEE world congress on computational Intelligence. Vol.5, pp. 2819-2824.
 [7] Ogawa, T.; Matsuura, H. & Kanada, H. (2005). A Solution of Inverse Kinematics of Robot Arm Using Network Inversion. *Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation*.
 [8] Zurda, M. J. (1992). *Introduction to Artificial Neural System Network*. West Publishing Companies, ISBN 0-314-93397-3, St. Paul, MN, USA.