Tracking Control for Robot Manipulators based on Radial Basis Function Networks

Min-Jung Lee*, Jin-Hyun Park**, Hyang-Sig Jun***, Myoung-Ho Gahng*, Young-Kiu Choi****

*Gyeongnam Hub Robot Center, **Jinju National University

***Korea Aerospace Research Institute, ****Pusan National University

E-mail : mnjlee@mk21.org, uabut@jinju.ac.kr, hsjun@kari.re.kr,

gahngmh@mk21.org, ykichoi@pusan.ac.kr

요 약

신경회로망은 지능제어알고리즘 중의 하나로 학습능력을 가지고 있다. 이러한 학습능력 때문에 많은 분야에서 널리 사용되고 있으나, 지능제어의 단점인 안정도 문제를 수학적으로 증명하기 어렵다는 문제점을 갖고 있다. 본 논문에서는 신경회로망의 한 종류인 RBFN과 적응제어기법을 이용하여 로봇 매니퓰레이터 궤적 제어기를 구성하고 자 한다. 본 논문에서는 RBFN의 파라메터들을 적응제어기법을 이용하여 수학적으로 구하였고, 시스템의 안정도를 수학적으로 UUB를 만족한다는 것을 증명하였다. 그리고 수평다관절로봇 매니퓰레이터 궤적제어기에 적용하였다.

ABSTRACT

Neural networks are known as kinds of intelligent strategies since they have learning capability. There are various their applications from intelligent control fields; however, their applications have limits from the point that the stability of the intelligent control systems is not usually guaranteed. In this paper we propose a neuro-adaptive controller for robot manipulators using the radial basis function network(RBFN) that is a kind of a neural network. Adaptation laws for parameters of the RBFN are developed based on the Lyapunov stability theory to guarantee the stability of the overall control scheme. Filtered tracking errors between the actual outputs and desired outputs are discussed in the sense of the uniformly ultimately boundedness(UUB). Additionally, it is also shown that the parameters of the RBFN are bounded. Experimental results for a SCARA-type robot manipulator show that the proposed neuro-adaptive controller is adaptable to the environment changes and is more robust than the conventional PID controller and the neuro-controller based on the multilayer perceptron.

Keywords

Neuro-adaptive controller, Radial basis function network, Lyapunov stability theory, Robot manipulators

I. Indtroduction

Robot manipulators used as industrial automatic elements are systems with highly nonlinear dynamics that are often unknown. Conventional feedback controllers such as PID controllers are commonly used in the field of industry since their control architectures are very simple and easy to implement; however, when these conventional feedback controllers are directly applied to the robot manipulators, they may suffer from poor performance and robustness due to unknown nonlinearities and external disturbances. For some decades, to deal with unknown nonlinearities and external disturbances, a lot of researches have been pursued to find various control strategies such as automatic tuning of PID control, variable structure control, feedback linearization, model reference adaptive control, direct adaptive control, intelligent control, etc. [1]-[4].

This paper proposes another neuro-adaptive controller based on the RBFN. It deals with tracking control problems for robot manipulators that are multi-input-multi-output systems. The proposed controller has a parallel structure that consists of a fixed gain PD controller and a RBFN controller. The PD controller is used to control the robot manipulator during the initial learning stage of the RBFN. The role of the PD controller is reduced after the learning stage of the RBFN. On-line adaptive laws are derived to adjust the weights and both of centers and widths of the RBFN and are constructed to guarantee the stability of the total control system on the basis of the Lyapunov stability theory.

Finally, the proposed neuro-adaptive control scheme is applied to a SCARA-type robot manipulator. We can find experimentally the validity of the neuro-adaptive control scheme by comparing with other control strategies.

II. Dynamics and Structural Properties

For control design purpose, it is necessary to have a mathematical model that reveals the dynamical behavior of a system.

Using the Euler-Lagrangian formulation, the equations of motion of an n-link manipulator can be written as [1]

$$D(q)\ddot{q}+C(q,\dot{q})+G(q)+\tau_d=\tau \qquad (1)$$

where $q\in R^n$ is the joint position, $D(q)\in R^{n\times m}$ is the inertial matrix, $C(q,\dot{q})\dot{q}\in R^n$ represents the Coriolis/centrifugal torque, $G(q)\in R^n$ is the gravitational torque, $\tau_d\in R^n$ is the disturbance torque, and $\tau\in R^n$ is the applied joint torque.

The dynamics of robot manipulator in the form of (1) is characterized by the following structural properties.

Property 1: The inertial matrix $D(q)$ is symmetric and positive definite, and there exist scalars, m_1 and m_2 such that $m_1 I\le D(q)\le m_2 I$.

Property 2: The Coriolis/centrifugal torque $C(q,\dot{q})\dot{q}$ is bounded by $c_b(q)\,\|\dot{q}\|^2$ with $c_b(q)\in C^1(S)$. S is a simply connected compact set of R^n

Property 3: The matrix $\dot{D}(q)-2C(q,\dot{q})$ is skew-symmetric. That is, the matrix is satisfied with $x^T\{\dot{D}(q)-2C(q,\dot{q})\}x=0$ for $\forall x\in R^n$.

Property 4: The unknown disturbance satisfies $\|\tau_d\|<b_d$ with a known positive constant b_d.

Property 5: There exists a vector $\theta\in R^n$ with components that depend on the manipulator parameters (masses, moments of inertia, etc.) such that

$$D(q)\ddot{q}+C(q,\dot{q})+G(q)=Y(q,\dot{q},\ddot{q})\theta=\tau \qquad (2)$$

where $Y(q,\dot{q},\ddot{q})\in R^{n\times r}$ is the regressor which is a function of time.

These properties are well known and generally used to design adaptive controllers.

III. Radial Basis Function Networks

The RBFN proposed by Moody, Darken, Powell, Broomhead and Lowe is used to approximate nonlinear functions and has faster convergence time than the multi-layer perceptron. RBFN also has similar feature to fuzzy inference system (FIS). That is, the output value is calculated using the weighted sum or weighted average method, the number of hidden layer's node of RBFN is the same as the number of if-then rules in FIS, and the radial basis functions are similar to the membership functions of FIS' premise part [4]. Fig. 1 shows the structure of RBFN with *M* hidden nodes. RBFN consists of hidden layer and output layer. The number of hidden layers are determined by designer. Gaussian function, triangular function and trapezoidal function are usually employed as basis functions of RBFN.

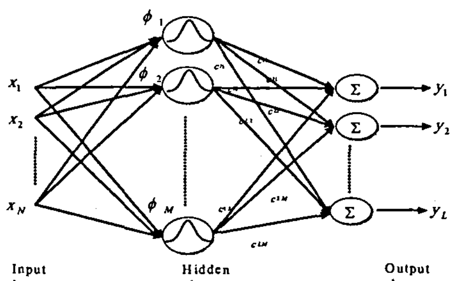

Fig. 1. Structure of the RBFN.

In this paper Gaussian function is selected as a basis function, and the output of RBFN is calculated by the weighted sum method. (3) shows the *i*-thoutput of RBFN.

$$y_i=\sum_{j=1}^{M}c_{ij}\phi_j \qquad i=1,2,\cdots,L \qquad (3)$$

$$\phi_i(x)=\exp\left(-\frac{\|x-u_j\|^2}{\sigma_j^2}\right) \qquad (4)$$

where **M** and **L** are the number of hidden nodes and output nodes, respectively. c_{ij} is the weight that connects the *j*-th hidden node to the *i*-th output node of RBFN, Φ_j is the *j*-th basis function, $u_j\in R^k$ is the *j*-th center vector, σ_j is the *j*-th standard deviation, and *k* denotes the number of input nodes.

In this paper, the RBFN is used to design a neuro-adaptive controller because its structure is simpler than the MLP. The RBFN is employed as a nonlinear function approximator. (5) shows the system model based on the RBFN with m hidden nodes.

$$y = c^T \Phi + \epsilon \quad (5)$$

where

$$y = [y_1 y_2 \cdots y_L]^T \quad \Phi = [\phi_1 \phi_2 \cdots \phi_M]^T$$

$$c^T = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1M} \\ \vdots & \vdots & \ddots & \vdots \\ c_{L1} & c_{L2} & \cdots & c_{LM} \end{bmatrix} \quad \epsilon = [\epsilon_1 \epsilon_2 \cdots \epsilon_L]^T$$

In (5), there are approximation error vector ϵ because we just consider the finite dimensional hidden nodes of RBNF. The approximation error vector ϵ can be made very small. Its norm is bounded by a known constant value according to the approximation theorem [6][10]. That is,

$$\| \epsilon \| \le \epsilon_N \quad (6)$$

where ϵ_N is the upper bound of $\| \epsilon \|$.

IV. Nero-Adaptive Controller

The proposed neuro-adaptive controller has a parallel structure that consists of the RBFN and PD controller. Fig. 2 shows the structure of the proposed controller.

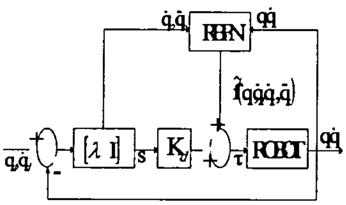

Fig. 2. Structure of the proposed controller.

Since the RBFN is trained in real-time, we can get good performance even under the existence of disturbances and change of parameters.

To update the parameters of the RBFN, adaptation laws are contructed using the Lyapunov stability theory. If the reference trajectory, $q_d \in R^n$, is given, the tracking error e is defined as $q_d - q$. And we propose a neuro-adaptive controller as follows:

$$\tau = \hat{c}^T \Phi(x) + Ks \quad (7)$$

where $x = [q_d \dot{q}_d \ddot{q}_r]$, $\dot{q}_r = \dot{q}_d + \Lambda q$, $\Lambda = \Lambda^T > 0$, $s = \dot{e} + \Lambda e$, and is diagonal and positive definite.

we have the following adaptation laws:

$$\dot{\hat{c}} = \Gamma_1 \Phi s^T \quad (8)$$

$$\dot{\hat{u}} = -\Gamma_2 \| s \| \hat{u} \quad (9)$$

$$\dot{\hat{\sigma}} = -\Gamma_3 \| s \| \hat{\sigma} \quad (10)$$

where Γ_1, Γ_2 and Γ_3 are diagonal, symmetric and positive definite matrices.

To prove the stability of the total control system, a Lyapunov function candidate is defined as

$$V = \frac{1}{2} s^T D(q) s + \frac{1}{2} tr(\tilde{c}^T \Gamma_1^{-1} \tilde{c}) + \frac{1}{2} tr(\tilde{u}^T \Gamma_3^{-1} \tilde{u}) + \frac{1}{2} tr(\tilde{\sigma}^T \Gamma_3^{-1} \tilde{\sigma}) \quad (11)$$

where $\tilde{c}(= c^* - \hat{c})$ is the weight matrix error between the optimal weight matrix c^* and estimated weight matrix $\hat{c}$ of the RBFN in(7). Also $\tilde{u}(= u^* - \hat{u})$ and $\tilde{\sigma}(= \sigma^* - \hat{\sigma})$ denote the center error and standard deviation error, respectively.

We have the following inequality:

$$\dot{V} \le - \| s \| \left\{ -K_{min} \| s \| + b_d + \epsilon_N + \frac{1}{4} u_{max}^2 + \frac{1}{4} \sigma_{max}^2 \right\} \quad (12)$$

where K_{min} is the minimum diagonal element of K, u_{max} is the maximum value of $\| u^* \|$, and σ is the maximum value of $\| \sigma^* \|$.

If we have the following condition:

$$\| s \| \ge \frac{b_d + \epsilon_N + \frac{1}{4} u_{max}^2 + \frac{1}{4} \sigma_{max}^2}{K_{min}}, \quad (13)$$

then

$$\dot{V} \le 0 \quad (14)$$

V. Experimental Results and discussion

A SCARA-type robot manipulator shows in Fig. 3 is employed as a testbed in this paper.

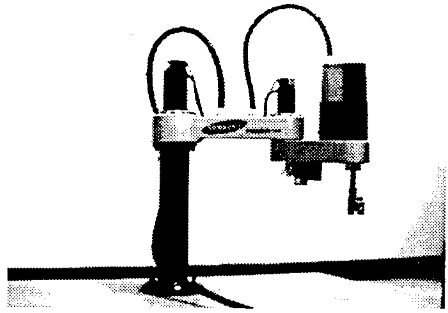

Fig. 3. SCARA type robot manipulator.

The proposed neuro-adaptive controller is compared with two other controllers: the PID controller and the neuro-controller using the MLP [5]. The first control scheme is a PID type conventional control method with constant gains such as K_{p1}=24.23, K_{i1}=7.27, K_{d1}=0.60, K_{p2}=39.9, K_{i2}=8.34, and K_{d2}=1.78 for the joints 1 and 2. In order to experiment the neuro-controller using the MLP, we have four and ten neurons in the input layer and hidden layer, respectively. The error-backpropagation algorithm is used as an updating algorithm for the neuro-controller using the MLP. The proposed control inputs (7), (8), (9), and (10) are used for the experiment. Control parameters are given in Table 2. The numbers of neuron of the RBFN is set to be ten. The sampling time is set to be 5[ms]. Three different cases of experiments are done to consider the performance under various environments. The first case is for a sinusoidal reference trajectory with frequency, ω =3.76[ms].

The reference input trajectories are defined as $q_1^d(t)=0.4\cos(3.76t)$[rad]and $q_2^d(t)=0.4\{\sin(3.76t)+1\}$ [rad] for the joints 1 and 2, respectively. Fig. 3 and 4 how the tracking errors of the PID controller, the neuro-controller using the MLP, and the proposed controller. The neuro-controller using the MLP and the proposed controller can make the tracking errors reduced during the learning process since both controllers have the learning ability. But the proposed controller has faster reduction rate in tracking errors than the neuro-controller using the MLP.

Experimental results mentioned above indicate that the proposed controller is very adaptable to the environmental changes and is more robust than PID controller and the neuro-controller using the MLP.

VI. Conclusions

This paper presents a neuro-adaptive controller using the RBFN to control robot manipulators. The proposed controller has a parallel structure that consists of the PD controller with fixed gains and the RBFN. The weights and both of centers and standard deviations of the RBFN are adjusted in real-time. The learning laws are constructed using the Lyapunov stability theory.

This paper shows that tracking errors are bounded uniformly and ultimately under the existence of disturbances and modeling errors. The SCARA type robot manipulator is employed as a testbed for the proposed neuro-adaptive controller. We compared the proposed controller with two different controllers: the PID controller and neuro-controller with the MLP. Experimental results show that the proposed neuro-adaptive controller is adaptable to environment changes and is more robust than the conventional PID controller and neuro-controller with the MLP.

Reference

[1] K.S. Fu, R.C.gonzalez, and C.S.G.Lee, *Robotics*, McGraw-Hill International Editions. 1987.

[2] M.W. Spong and M. Vidyasgar, *Robot Dynmamics and Control*, John Wiley&Sons, 1989.

[3] J.-J.E.Slotine and W. Li, *Applied Nonlinear Control*. Prentice-Hall, 1991.

[4] J.-S.R.Jang, C.-T.Sun, and E.Mizutani, *Neuro-Fuzzy and Soft Computing*, Prentice-Hall, 1997.

[5] Sigeru Omatu, Maruki Khalid, and Rubiyah Yusof, *Neuro-Control and Its Applications: Advaces in Industrial Control*, Springer-Verlag, 1996.

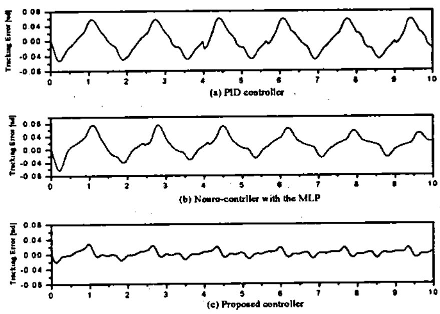

Fig. 3. Tracking errors of the joint 1.

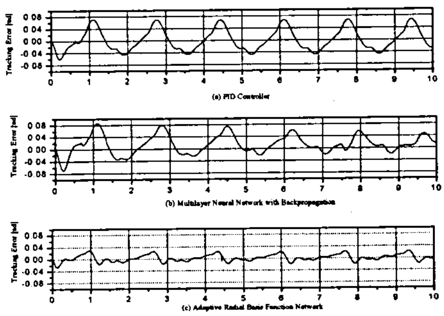

Fig. 4. Tracking erros of the joint 2.