

The Adaptive Tracking Control of Nonlinear System based on RBFN

Hyun-Seob Cho

*Dept. of Electronic Engineering, Chungwoon University,
Daehakgil-25, Hongseong, Chungnam, 350-701, South Korea
chohs@chungwoon.ac.kr*

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 an Adaptive Tracking Control of Nonlinear System the radial basis function network (RBFN) that is a kind of neural networks. The learning method involves structural adaptation and parameter adaptation. No prior knowledge of the plant is assumed, and the controller has to begin with exploration of the state space. The exploration versus exploitation dilemma of reinforcement learning is solved through smooth transitions between the two modes. The controller is capable of asymptotically approaching the desired reference trajectory, which is showed in simulation result.

Keywords: *Radial Basis Function Network, Adaptive Tracking Control, Nonlinear System*

1. Introduction

In this thesis, we propose a approach to NN-based self-tuning adaptive control, where the self-tuning principle and RL are combined, with special emphasis on resolving the exploration versus exploitation dilemma. The control scheme consists of a controller, a utility estimator, an exploration module, a learning module and a rewarding module [1]. The controller and the utility estimator are implemented together in a single Radial Basis Function neural network (RBFN). The learning method involves structural adaptation and parameter adaptation. The exploration versus exploitation dilemma of RL is resolved through smooth transitions between the two modes. This enables rapid exploration response to novel plant dynamics and stable operation in an occurrence of changes in plant dynamics.

2. Dynamics and Structural Properties of Neuro Adaptive Control System

There are some learning tasks arising in control problem require methods that are not accurately characterized as supervised learning [2]. Reinforcement learning controllers use less informative error signal than supervised learning controllers. The advantage is that the error signal is normally directly available from the environment with little or no processing. The importance of reinforcement learning controllers is that they can be trained on-line, and have a greater capacity for learning about and accommodating unexpected events in the environment than do other neuro-controllers [3]. In the reinforcement learning system, the dynamic model of the system is not known in advance, and must be estimated from data. Since the structure of the dynamic model is fixed, model estimation is a parameter estimation problem [4]. In the adaptive control algorithms, a common formulation of cost functions are quadratic penalties on deviation from desired state and action vectors.

3. Formulation of nonlinear control problem

Assume that a single-input-single-output (SISO) nonlinear discrete system is given in the form

$$\begin{aligned} y(k+1) &= f[q(k), u(k)] \quad (\text{state equation}) \\ y(k) &= g[q(k)] \quad (\text{output equation}) \end{aligned} \quad \text{Eq(3.1)}$$

Where $q \in R^n$ are the state variables, $u(k) \in R^1$ is the control input, $f[\cdot]$ and $g[\cdot]$ are the nonlinear maps on R^n , $f[\cdot]$ is bounded away from zero, and $y(k) \in R^1$ is the plant output. Finding a control signal $u(k)$ that will force the output $y(k)$ to track asymptotically the desired output $y_d(k)$, that is

$$\lim_{k \rightarrow \infty} [y_d(k) - y(k)] = 0 \quad \text{Eq(3.2)}$$

To achieve the above, the following assumptions about the nonlinear plant are required:

Assumption 1. for any $q \in R^n$

$$0 < k_1 \leq |f[\cdot]| \quad \text{Eq(3.3)}$$

Assumption 2. For any $k \in [0, \infty]$, the desired output $y_d(k)$ and its n -derivatives $y_d^{(1)}(k), y_d^{(2)}(k), \dots, y_d^{(n)}(k)$, are uniformly bounded, that is,

$$|y_d^{(i)}(k)| \leq m_i \quad i = 0, 1, 2, \dots, n \quad \text{Eq(3.4)}$$

Assumption 3. There exist coefficients a_{ff} and b_{fb} such that $\hat{f}[\cdot]$ and $\hat{g}[\cdot]$ approximate the nonlinear functions $f[\cdot]$ and $g[\cdot]$, respectively, with an accuracy ε on Σ , a compact subset of R^n , that is,

$$\begin{aligned} \max |f[\cdot] - \hat{f}[\cdot]| &\leq \varepsilon \\ \max |g[\cdot] - \hat{g}[\cdot]| &\leq \varepsilon \quad \forall q \in \text{on } \Sigma \end{aligned} \quad \text{Eq(3.5)}$$

A nonlinear system can be represented by one of the four discrete-time models as suggested in [2]. These models are described by the following nonlinear difference equations:

Model I

$$y(k+1) = \sum_{i=0}^{m-1} \alpha_i y(k-i) + g[u(k), u(k-1), \dots, u(k-m+1)] \quad \text{Eq(3.6)}$$

Model II

$$y(k+1) = f[y(k), y(k-1), \dots, y(k-n+1)] + \sum_{j=0}^{m-1} \beta_j u(k-j) \quad \text{Eq(3.7)}$$

Model III

$$y(k+1) = f[y(k), y(k-1), \dots, y(k-n+1)] + g[u(k), u(k-1), \dots, u(k-m+1)] \quad \text{Eq(3.8)}$$

Model IV

$$y(k+1) = f[y(k), y(k-1), \dots, y(k-n+1), u(k), u(k-1), \dots, u(k-m+1)] \quad \text{Eq(3.9)}$$

where $[u(k), y(k)]$ represents the input-output pair of a SISO plant at time k . In all four models, the output of the plant at time $(k+1)$ depends both upon on its past n values as well as the past m values of the input (output of the neural net). In the above equations, $f[.]$ and $g[.]$ are nonlinear functions that may take different forms. In model I, the plant output $y(k+1)$ is a linear function of the past values $y(k-i)$, while in Model II the relation between $y(k+1)$ and the past values of the control input $u(k-j)$ is assumed linear. In model III, the nonlinear relation of $y(k+1)$ with $y(k-i)$ and $u(k-j)$ is assumed to be separable, while Model IV, in which $y(k+1)$ is a nonlinear function of $y(k-i)$ and $u(k-j)$ subsumes Models I-III, is analytically least tractable.

The performance of the neural network with only synaptic adaptation has been discussed extensively for the above four models in [5]. A study of performance of the proposed neural network for a set of nonlinear systems represented by Model IV is reported in this thesis from the adaptive control point of view.

4. Radial Basis Function Neural Network (RBFN)

Radial basis function network is a type of ANN, whose locality property is more suitable to adaptive control application than other network architectures, such as the popular back-propagation (BP) networks. The reason is that RBFN allows to incrementally construct a function approximation without modifying the knowledge previously acquired when the weights are updated or new neurons are added to the network [6].

The structure of RBFN is a network with two layers: a hidden layer of radial basis neurons and an output layer of linear neurons. There are no weight parameters associated with any of the input interconnections. The activation function for a radial basis neuron is a Gaussian function given by the equation:

$$G_i(x) = \exp\left(-\frac{\|x - c_i\|^2}{2\sigma^2}\right), i = 1, 2, \dots, m \quad \text{Eq(4.1)}$$

5. Design of an Adaptive Tracking Controller

The learning of a neural network is an iterative process. A neural network learns its environment and adjust its weights in each learning step [7]. A learning or training algorithm may be classified into three categories: supervised learning, unsupervised learning, and reinforcement learning.

Supervised learning has an external teacher. The teacher can be regarded as a system which has knowledge of the environment that is represented by a set of input-output example. Examples of supervised learning algorithm are least-mean-square (LMS) or back-propagation (BP) algorithm. The LMS algorithm is concerned with a single neuron, while the BP algorithm is concerned with a multilayered neural network. Thus, the BP algorithm includes the LMS algorithm as a special case.

The network parameters are adjusted by the training vector and the error signal. The training vector is drawn from the environment, and the error signal is the difference between the desired output and the real output. Supervised learning can be performed off-line or on-line. In off-line learning, once the desired performance is achieved, the network design is

fixed and the neural network operates in a static way. Whereas, in on-line learning, learning is performed without separate computational unit in real time.

Drawback of supervised learning has not teaching process, so a neural network cannot learn new strategies. This may be overcome by using reinforcement learning.

Unsupervised or self-organized learning required no external teacher to learn, i.e., no target examples are needed to train the network. This is less computationally complex and less accurate than supervised learning. Thus, unsupervised learning is useful in case that we don't have enough time and information to employ supervised learning. Kohonen feature maps and Adaptive Resonance Theory (ART) networks are examples of unsupervised learning networks.

6. Simulation Examples

List Simulation studies discussed in this section, a SISO nonlinear dynamic system of form:

$$Y(k+1)=f[y(k),y(k-1),\dots,y(k-n+1),u(k),u(k-1),\dots,u(k-m+1)] \quad \text{Eq(6.1)}$$

If the error is defined as $e(k) = y(k) - \hat{y}(k)$, where $y(k)$ is the desired output and $\hat{y}(k)$ is the actual output of the plant under control, the aim of control is to determine a bounded control input $u(k)$ such that

$$\lim_{k \rightarrow \infty} [y(k) - \hat{y}(k)] = \lim_{k \rightarrow \infty} e(k) = 0 \quad \text{Eq(6.2)}$$

In our further discussion, the input signal $r(k)$ is considered the desired response for the unknown nonlinear plant to follow, thereby avoiding the necessity for obtaining the desired model.

In the section, we present three simulation examples, the study of robustness of the dynamic neural network is carried out by making the plant undergo different nonlinear functions at different instants of the control process. These examples demonstrate the ability of the dynamic neural network that makes the plant adaptively follow the varying input signal.

Consider the system is described by the equation:

$$y(t+1) = \begin{cases} 1.5y(t)/(1+y^2(t)) + 0.3\cos(y(t)) + 1.2u(t) & t \leq 4000 \\ 1.5y(t)/(1+y^2(t)) + 0.3\sin(y(t)) + 0.6u(t) & t > 4000 \end{cases} \quad \text{Eq(6.3)}$$

The objective is to control the time-dependent nonlinear system according to the reference trajectory. The reference signal r is $r(t) = \sin(2\pi t/20)$, The reference is changed to a lower frequency at time $t = 8000$, is $r(t) = \sin(2\pi t/40)$.

The controller and the utility estimator are implemented as a 2-input 2-output growing RBFN:

$$[u(t), \hat{U}(t)] = f_{RBFN} [r(t), y(t)] \quad \text{Eq(6.4)}$$

Here we used the threshold value $\theta=0.95$, which guarantees proper overlapping of the basis functions. We applied the learning rate $\alpha = 0.1$, which was found to be convenient during the experiment. The reinforcement signal U , was generated in the interval $[0, 1]$ with value 1 representing perfect tracking. A random exploration component of the control action $\delta u(t)$,

was generated in the interval $[-(1-\hat{U}(t))\delta u_{\max}, (1-\hat{U}(t))\delta u_{\max}]$, where δu_{\max} denotes the maximal value of random perturbation δu .

Figure 1 and Figure 2 shows results of control using the proposed control scheme. System output y , reference signal r , and control u , are shown in the following time intervals: [3500 : 4500] and [7500 : 8500]. The intensive exploration phase is initiated at time $t=0$. The controller is investigating the effect of various control actions on the behaviour of the system. After several hundred time steps, tracking accuracy improves considerably and the exploration component δu is automatically reduced. After several hundred subsequent time steps, a stable control law with accurate tracking is established.

The system is changed at time $t= 4000$ according to Eq(6.3) and the operation of the controller is again switched to intensified exploration. Soon, output error is reduced and the changed control law stabilizes. The last change was introduced at time $t = 8000$, when the reference r , was modified. The dynamics of the controlled system were not changed and therefore no additional exploration was necessary for accurate control.

Figure 3 shows the number of neurons m , for the RBFN implementing the controller and the utility estimator. A rapid increase in the number of neurons is observed at the beginning of operation at time $t= 0$, and at time $t= 4000$ when the system is changed.

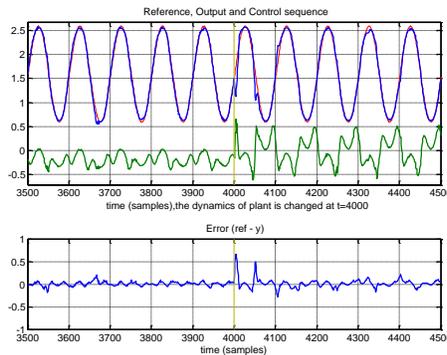


Figure 1. Input $r(t)$ (dashed line), plant output $y(t)$ and control signal $u(t)$ are plotted for three time intervals: [3500:4500]

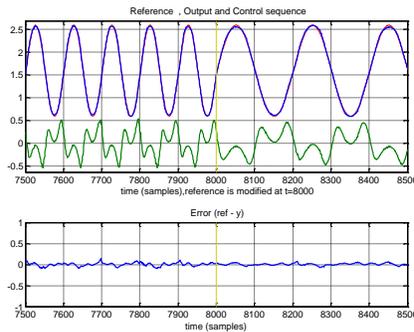


Figure 2. Input $r(t)$ (dashed line), plant output $y(t)$ and control signal $u(t)$ are plotted for three time intervals: [7500:8500]

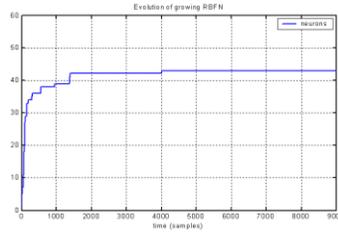


Figure 3. The number of neurons m is plotted during the evolution of growing RBFN

7. Conclusion

Direct adaptive self-tuning neuro-control for time-dependent nonlinear plants has been proposed in this thesis. The method is model-free and uses an adaptive exploration/exploitation strategy through reinforcement learning to investigate the properties of the system. The controller and the utility estimator are implemented together in a single growing RBFN. Learning of the RBFN involves structural adaptation and parameter adaptation. The exploration-exploitation dilemma is resolved through smooth transitions between the two modes. This method has been shown to successfully control the simulated nonlinear system.

Of course, there are still some works need to improve in the future. In the study, a simple exploration strategy was used, where only maximal amplitude of the perturbation was adapted on-line. A more elaborate exploration strategy could be utilized to improve the convergence properties. For real-world applications, pruning capacity of growing RBFN would have to be added in order to limit the size of ANN. A strategy should be devised to determine and eliminate obsolete neurons generated during the exploration phase.

References

- [1] K. Ogata, "Modern Control Engineering", 2nd Ed., Prentice Hall, (1990).
- [2] K. S. Narendra and K. Parthasarthy, "Identification and Control of Dynamical Systems Using Neural Networks", IEEE Trans. Neural Networks, vol. 1, no. 4, (1990).
- [3] M. M. Gupta and D. H. Rao, "Dynamic Neural Units in the Control of Linear and Nonlinear Systems", In Proceedings of the International Joint Conf. On Neural Networks, (1992) June, pp. 100-105
- [4] J. Moody and C. Darken, "Learning with localized receptive fields", In D. Touretzky, G. Hinton, and T. Sejnowski, (Eds.), Proceedings of the 1988 Connectionist Models Summer School, San Mateo, (1989), pp. 133-143.
- [5] J. C. Platt, "A resource-allocating network for function interpolation," Neural Computation, vol. 3, no. 2, (1991), pp. 213-225.
- [6] B. Fritzke, "Fast Learning with incremental RBF Networks", Neural Proc. Lett., vol. 1, no. 1, (1994), pp. 2-5.
- [7] S. Haykin, "Neural Networks", Macmillan, (1994).

Author



Hyun-Seob Cho received his B.S. degree Electrical Engineering from Wonkwang University, Korea, in 1990, and M.S. and Ph.D. degrees in Electrical Engineering from Wonkwang University, Korea, in 1992 and 1996 respectively. He is currently an Professor in Chungwoon University, Chungnam, Korea. His research interests include Electric and Electronics Application and Automatic Control.