다중신경회로망과 관리제어를 이용한 안정한 동정화와 제어

엄태덕, 김성우, 홍선기, 이주장
한국과학기술원 전기 및 전자공학과

Stable Identification and Control Using Multiple Neural Networks and Supervisory Control

Tae-Duck Ohm, Sung-Woo Kim, Sun-Gi Hong and Ju-Jang Lee
Department of Electrical Engineering, KAIST

Abstact - **In many applications of the neural networks to system identification and control,** we often ignore the problem of stability. This paper suggests successive identification and control scheme using multiple neural networks to handle this critical problem of stbility. The main theme of the scheme is that we always assure the identification process in the basin of attraction of one unique stable equilibrium point obtained by nonlinearity canceling. The starting point of identification is the small reigon we know in the whole state space of system and this region can be obtained through the concept of supervisory control. We cannot assure the stability one hundred percent but it is a reasonable and powerful scheme as we can see from the simulation results.

I. INTRODUCTION

The main concept can be split in two. First, we always identify the system with the state trajectory converging to the stable equilibrium point and start with the region identified under stability using supervisory control so we can always assure the stability until task is completed. Second, identification on the domain of stable equilibrium point avoids nonuniform data sampling which occurs due to unstable equilibrium points.

A. Supervisory Control

The target system for which we want to apply discrete-type supervisory control must have an input as an additive term, that is,

$$x(k+n) = f(x(k), x(k+1), \cdots, x(k+n-1)) + bu \qquad (1)$$

In such a system, if function f is bounded, $|f| < f^U$, by adding a supervisory control input u_s activated when the Lyapunov function V of error dynamics is larger than V_M, small reigon is identified with state trajectory being confined in the assigned boundary. So to speak, supervisory control does the role of pushing when state trajectory shows a tendancy to exit out of the boundary.

When the supervisory control input is not activated, in other words, the current state is inside the pre-assigned region, we assume that the classical PD control is enough to roughly track the desired trajectory for getting uniformly distributed training samples. But most of cases concerned does not have this property, so it takes much time to make the neural network finely approximate the nonlinear function. Therefore, it's better to choose the supervisory controlled region as small as possible.

B. Identification and Control Using Multiple Neural Networks

After small region is identified by the supervisory controller, we start the new procedure of successive identification and control. First, we make a stable equilibrium point at the center of the identified region. Then, checking the relative stability by a certain criterion function, the neural network is trained until the approximation error is smaller than we want. This process assures stability cause we get the training samples from the state trajectory converging to the equilibrium point. The feasibility of this procedure is based on the fact that there always exists more or less stable region due to the generalization property of the neural networks.

The significant problem in combining supervisory control and successive identification and control algorithm is that successive identification and control algorithm we developed is applied to a system which is different from (1). But when the order of a system is the first, two forms are same. The efforts to match two forms are now proceding.

II. DESIGN OF THE SUPERVISORY CONTROLLER

The system equation is given by the equation of motion (1) where f is an unknown nonlinear function, b is a known positive constant and $u \in R$ is the input. Define the state vector $x(k)=[x(k),x(k+1),\cdots,x(k+n-1)]'$ and it is available by measurement. Our control objective is to design a neural network identifier and controller to track the given reference signal x_d by using the pre-designed supervisory controller. The nonlinear funtion $f(x)$ is unknown but assumed to be bounded such that

$$|f(x)| \le f^U(x). \tag{2}$$

Let the error vector $x_e = x - x_d$, then we can control the system with the following

$$u^* = \frac{1}{b}[-f(x) + x_d(k+n) - k'x_e] \tag{3}$$

where the gain $k=[k_1,k_2,\cdots,k_n]'$ is chosen for the polynomial $h(z)=z^n+k_n z^{(n-1)}+\cdots+k_1$ to have eigenvalues inside the unit circle. Applying (3) to (1) makes the system asymptotically stable, but this control input can not be implemented since $f(x)$ is not known. Our purpose is to design a neural network controller instead of using (3). Before designing the neural network controller, we design a supervisory controller that is used to train the networks within the region of interest.

From now on, the supervisory controller is designed using Lyapunov theory to guarantee the boundedness of the system trajectory. Suppose that the control u is the addition of the neural network based controller, u_c, which will be designed later, and the supervisory control, u_s, therefore,

$$u = u_c + u_s \tag{4}$$

Substituting (4) to (1), we have

$$x(k+n) = f(x(k)) + b(u_c + u_s). \tag{5}$$

By adding and substracting bu^* in (1), we obtain the equation of error:

$$x_e(k+n) = -k'x_e(k) + b(u_c + u_s - u^*), \tag{6}$$

or equivalently

$$x_e(k+1) = \Lambda x_e(k) + b(u_c + u_s - u^*) \tag{7}$$

where

$$\Lambda = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -k_1 & -k_2 & -k_3 & \cdots & -k_n \end{bmatrix}, \qquad b = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ b \end{bmatrix}. \tag{8}$$

Let's define $V = \frac{1}{2} x_e' P x_e$, where P is a symmetric positive definite matrix satisfying the following Lyapunov equation

$$\Lambda' P \Lambda - P = -Q, \tag{9}$$

where Q is positive definite. Then the forward difference of V along the system trajectory becomes

$$\begin{aligned} \Delta V &= V(k+1) - V(k) \\ &= -\frac{1}{2} x_e' Q x_e + x_e' \Lambda' P b (u_c + u_s - u^*) \\ &\le -\frac{1}{2} x_e' Q x_e \\ &\quad + |x_e' \Lambda' P b| (|u^*| + |u_c|) \\ &\quad + x_e' \Lambda' P b u_s \end{aligned} \tag{10}$$

We build the supervisory control u_s as follows

$$\begin{aligned} u_s = & -I sgn(x_e' \Lambda' P b)[\,|u_c| \\ & + \frac{1}{b}(f^U + |x_d(k+n)| + |k' x_e|)] \end{aligned} \tag{11}$$

where $I=1$ if $V>V_M$ and $I=0$ otherwise and V_M is a constant specified by the designer according to the training region of the neural networks. Substituting (11) and (3) into (10) for the case of $I=1$, we have

$$\dot{V} \le -\frac{1}{2}x_eQx_e + |x_e'\Lambda'Pb|\frac{1}{b}(|f|-f^U) \le -\frac{1}{2}x_eQx_e < 0 \qquad (12)$$

Therefore using the supervisory control u_s of (11), we always have $V<V_M$. Further, since P is positive definite, the boundedness of V implies the boundedness of x_e, which implies the boundedness of x.

SIMULATION STUDY:
We consider the inverted pendulum of which the dynamic equation is governed by

$$ml^2\ddot{x}+B\dot{x}+mglcos(x)=u \qquad (15)$$

where x denote the joint angle. For the simplicity of simulation, assume all the parameters of (15) are 1. Then the equation (15) results in

$$\ddot{x}=-\dot{x}-\cos(x)+u \qquad (16)$$

We select V_M such that $|x(t)| \le \frac{\pi}{24}\sin(\frac{t}{2})$. Fig.1 shows the tracking control of the desired trajectory given by $x_{d(t)}=\frac{\pi}{24}\sin(\frac{t}{24})$.

III. SUCCESSIVE IDENTIFICATION AND CONTROL

We used a supervisory control input to get the identification of small region under stability and set the base for extention to the outer space. In this chapter the concept of successive identification and control will be explained.

When the system is the first order, we will use graphical analysis to show the reasonablity of the control scheme.

We focuses on the system which can be expressed as the following nonlinear difference equation:

$$x(k+1)=f[x(k)]+u(k) \qquad (17)$$

where x(k) and u(k) are the state and control input at time k, $u(k),x(k)\in R^n$, and $f\,R^n \rightarrow R^n$ is an unknown mapping. If we know the function in some domain we can cancel out the nonlinearity and make a point asymptotically stable. Such a input $u(k)$ is $-f[x(k)]+Ax(k)+(I-A)x^*$.Here matrix A has eigenvalues inside the unit circle. So, if we know a little part of system, we can make a stable equilibrium point in this region. Then we find the basin of attraction of this point is stretching outside due to the generalization property. Here the word generalization looks like somewhat vague. But we can understand that if we handle the problem using graphical analysis for the first order case.

The concept of successive identification and control was first proposed by Kumpati S. Narendra and Asriel U. Levin [3]. I think there are two major problems. First, they use a theorem(proof in [2]) below to assure that the identification is performed in the basin of attraction of a stable equilibrium point. But, that is a purely mathematical theorm so in practical applications it is difficult to find input bound u that makes the state trajectory inside the basin of attraction. Also, they use the error between real function and approximated neural network as another neural network initiation condition. For this reason, though they got a successful result about the study of uniform sampling, they did not find any method to maintain stability.

Theorem: Let p *be the unique asymptotically stable equilibrium point of* f *on the interval* I, *then starting at any point* $x(0)\in I$ $\exists\varepsilon$ *s.t if* $\|u(k)\|<\varepsilon$ *then* $\forall k$ $x(k)\in I$.

To solve these problems we suggest a supervisory control mentioned earlier and a new neural network initiation condition. I will show the validaty of this criterion using graphical analysis. In figure (a) below if the initial condition is min(L1,L2) apart from the equilibrium point they converge to the equilibrium point because in this region it is always satisfied that $|x(k+1)-x^*| < |x(k)-x^*|$. Let's assume that we have a function N_f which approximates $f(x(k))$ with the error less than δ in the region $|x-x^*| < D$. Then, we can make an equilibrium point at the center of the known region by subtracting N_f from f and adding x^*. The function $f-N_f+x^*$ has the value near x^* in the region $|x-x^*| < D$ and does not vary fast outside this region by the generalization property of the neural networks(figure (b)). This makes us get some stable region and if we identify the region where $|x(k+1)-x^*| < D$, we can satisfy $|x(k)-x^*| < \delta$ in 2-step. After finishing the 1st identification we use another neural network and repeat this procedure to reach the goal.

We explain the procedure with the aid of the graphical analysis in the case of the 1st order system, but this algorithm can be applied for the n-th order system which can be expressed as (17).

SIMULATION STUDY:

We peform the simulation for the 1st order case with the assumption that we know the region [-2, -1.5]. The result is in Fig. 3.

IV. CONCLUSION

The most serious problem of this paper is that the bridge between supervisory control and successive identification and control algorithm is too narrow. The combination of both concepts are only available for the 1st order system. But, the neural network identifier with the supervisory control provides the fine control where the PD-control is roughly working and successive identification provides uniform and stable sampling of training data in the restricted class of systems.

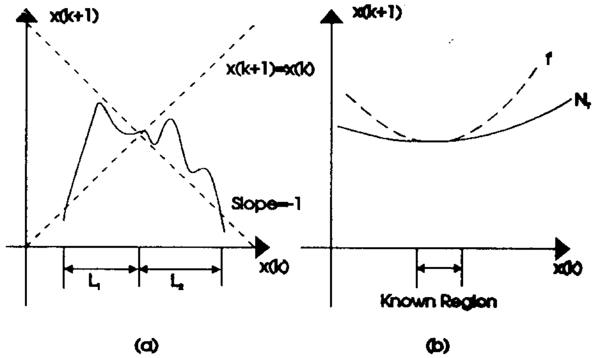
References

[1] D. Psaltis, A. Sideris, and A. A. Yamamura, "A multilayered Neural Network controller," *IEEE Control System. Magazine*, pp. April 1988

[2] R. L. Devaney, "An Introduction to Chaotic Dynamical Systems" *Addison Welsley*, Second Edition, 1989

[3] Kumpati S. Narendra and Asriel U. Levin, "Regulation of Nonlinear Dynamical Systems Using Multiple Neural Networks", *procedings of ACC*, 1992.

[4] Li-Xin Wang, "Stable Adaptive Fuzzy Control of Nonlinear Systems", *IEEE Transactions on Fuzzy Systems*, vol. 1, no. 2, 1993



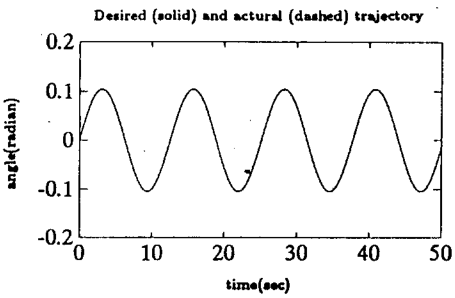


Fig. 1: Tracking Control Using Neural Network Identifier

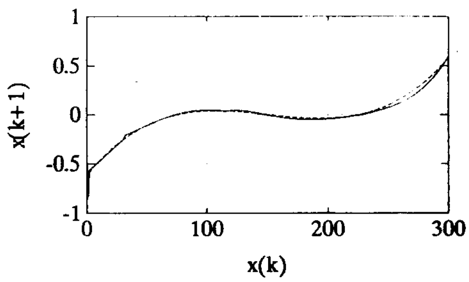


Fig. 3 : Approximation of x^3-x by multiple neural networks