Neural Network Identification and Control of Unstable Systems
Using Supervisory Control While Learning

Sung-Woo Kim, Sun-Gi Hong, Tae-Duck Ohm and Ju-Jang Lee

Department of Electrical Engineering
Korea Advanced Institute of Science and Technology
373-1 Kusong-dong, Yusong-gu, Taejon 305-701, Korea
E-mail: jjlee@ee.kaist.ac.kr

Abstract — We focus on the training scheme for the neural networks to learn in the regions of unstable equilibrium states and the identification and the control using these networks. These can be achieved by introducing a supervisory controller during the learning period of the neural networks. The supervisory controller is designed based on Lyapunov theory and it guarantees the boundedness of the system states within the region of interest. Therefore the neural networks can be trained to approximate sufficiently accurately with uniformly distributed training samples by properly choosing the desired states covering the region of interest. After the networks successfully trained to identify the system, the controller is designed to cancel out the nonlinearity of the system.

I. INTRODUCTION

The learning and approximation capability of neural networks are leading to their application in nonlinear system identification and control. However, for neural networks to successfully identify and to control the nonlinear system, several assumptions have been made concerning the characteristics of the system. For example, it has been assumed in identification problem that the system has bounded outputs for the class of inputs specified and in control problem that the inverse of operator for the system exists and can be approximated by the neural networks [4]. Therefore, to control unstable systems was a very difficult problem. Moreover, it is well known that neural networks cannot approximate nonlinear maps sufficiently accurately when their inputs are not uniformly distributed in the region of interest. This implies that the identification (and hence control) of nonlinear systems in the vicinity of unstable equilibrium states will pose serious difficulties, since adequate samples for training the neural networks in these regions in not generally available [5].

To overcome the difficulties listed above, we introduce the supervisory controller to train the neural networks in the specified regions of interest. The supervisory controller is designed based on Lyapunov theory to guarantee the boundedness of the system. Since the trajectory of the system can be made bounded in the arbitrary specified region, even though the system is unstable, we can obtain the training samples distributed in the region for the neural networks. After the system is sufficiently approximated within the region, then the neural network controller is designed to cancel out the nonlinear map of the system. In this case, the system is not nonlinear any more if the cancellation is sufficiently accurate, so that the system can be controlled by such simple controllers as PD or PID.

II. PROBLEM STATEMENT

This paper focuses on a class of dynamic systems describing by the following equation of motion:

\[ x^{(n)} = f(x, \dot{x}, \ldots, x^{(n-1)}) + bu \]  

where \( f \) is an unknown nonlinear function, \( b \) is a known positive constant and \( u \in \mathbb{R} \) is the input. These systems are normal form and have relative degree equal to \( n \). More general classes of nonlinear equations of motion can be transformed into this class [10].

Define the state vector \( x = [x_1, x_2, \ldots, x_n] = [x, \dot{x}, \ldots, x^{(n)}] \) and it is available by measurement. Our control objective is to design a neural network identifier and controller to track the given reference signal.
and to training the neural networks in the specified region by using the pre-designed supervisory controller. The nonlinear function \( f(x) \) is known and assumed to be bounded such that

\[
|f(x)| \leq f^U(x).
\]

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

\[
u^* = \frac{1}{b} \left[ -f(x) + x_d^{(n)} - k^T x_e \right]
\]

where the gain \( k = [k_1, k_2, \ldots, k_n]^T \) is chosen for the polynomial \( h(s) = s^n + k_1 s^{n-1} + \cdots + k_1 \) to be a Hurwitz. 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 training the networks within the region of interest.

### III. Design of the Supervisory Controller

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 next Section, and the supervisory control, \( u_s \), that is,

\[
u = u_c + u_s
\]

Substituting (4) into (1), we have

\[
x^{(n)} = f(x) + b(u_c + u_s).
\]

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

\[
x_e^{(n)} = -k^T x_e + b(u^* - u_c - u_s),
\]

or equivalently

\[
x_e = A x_e + b(u^* - u_c - u_s),
\]

where

\[
A = \begin{bmatrix}
0 & 1 & 0 & \cdots & 0 \\
0 & 0 & 1 & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & \cdots & 1 \\
-k_1 & -k_2 & -k_3 & \cdots & -k_n
\end{bmatrix}, \quad b = \begin{bmatrix}
0 \\
0 \\
\vdots \\
0 \\
b
\end{bmatrix}
\]

Define \( V = \frac{1}{2} x_e^T P x_e \), where \( P \) is a symmetric positive definite matrix satisfying the following Lyapunov equation

\[
A^T P + PA = -Q,
\]

where \( Q > 0 \). Then the derivative of \( V \) along the system trajectory becomes

\[
\dot{V} = -\frac{1}{2} x_e^T Q x_e + x_e^T P b (u^* - u_c - u_s)
\leq -\frac{1}{2} x_e^T Q x_e + x_e^T P b (|u^*| + |u_c|) - x_e^T P b u_s.
\]

We build the supervisory control \( u_s \) as follows

\[
u_s = I \text{sgn}(x_e^T P b) \left[ |u_c| + \frac{1}{b} (f^U + |x_d^{(n)}| + |k^T x_e|) \right],
\]

2501
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

$$
\dot{V} \leq -\frac{1}{2}x_eQx_e + |x_e|^2Pb \left[ \frac{1}{b} \left( |f| + |x_d^{(n)}| + |k'x_e| \right) - \frac{1}{b} \left( fU + |x_d^{(n)}| + |k'x_e| \right) \right] \\
\leq -\frac{1}{2}x_eQx_e < 0. \tag{12}
$$

Therefore using the supervisory control $u_s$ of (11), we always have $V < V_M$. Further, since $P > 0$, the boundedness of $V$ implies the boundedness of $x_e$, which implies the boundedness of $x$.

IV. SYSTEM IDENTIFICATION AND CONTROL USING NEURAL NETWORKS

A. Identification

We want to identify the nonlinear function $f(x)$ of the system using neural networks. Among a lot of neural network structures, we choose the feedforward networks having multi-layered perceptrons. The networks containing $N$ layers denoted by the symbol $N_{i_1,i_2,...,i_{N+1}}$. Such networks have $i_1$ inputs, $i_{N+1}$ outputs and $(N - 1)$ sets of nodes in the hidden layers, each containing $i_2, i_3, \ldots, i_N$ nodes, respectively. The output approximated by the networks will be denoted by $N_f(x)$. The training algorithm for the networks is the generalized delta rule. For the activation function of each node, the sigmoid function $S(z) = \tanh(z)$ is used. We can obtain the information on the nonlinear function of the system at every sampling time by subtracting $U$ from $z$, that is $f(x(nT)) = z^{(n)}(n(T + 1)) - bu(nT)$, \(13\)

where $T$ is sampling time. We use this information as a training samples. In such case, the inputs for the networks are the elements of the measured state vector $x$ at every sampling time. Our aim is to train the networks for the output of the networks $N_f(x)$ to approximate the function $f(x)$ as closely as possible.

If the system generates bounded outputs in the specified input regions, the neural networks can sufficiently approximate the nonlinear function. However, if the system is unstable in the specified region, then the training samples for neural networks are not adequate to approximate the mapping. In this case, the neural network identification is inaccurate, so in turn the control based on the identified networks is not accurate. Therefore, it is needed a scheme that can generate good training samples for the neural networks to be learned properly. In order to get this purpose, we utilize the supervisory controller designed in the previous Section.

The identification procedure is: At first, the system is initiated with no control inputs. If the system is stable, then the system trajectory goes to one of its stable equilibrium points. In this case, there are no needs to use the supervisory controller. The neural networks learn and identify the nonlinear function during the iteration process. On the contrary, if the system is unstable, the system trajectory will be blown up. In the case when the system trajectory goes over the pre-determined region by $V_M$, that is $x(t) \in \Phi(x)$, where $\Phi(x) = \{x \mid x_e, P x_e < V_M\}$, the supervisory controller is active to push the trajectory into the region. Therefore we can continuously obtain the training input/output samples in the region.

B. Control

The control strategy is to cancel out the nonlinear function in the system using the trained neural networks. The control input is generated by the following:

$$
u_c = -\frac{1}{b}N_f(x) + u_{PD} = -\frac{1}{b}N_f(x) - k'x_e \tag{14}$$

where $u_{PD} = -k'x_e$ is a PD type controller. If the networks are sufficiently accurate to verify the cancellation, then the error is described by

$$
x_e^{(n)} + k_1x_e^{(n-1)} + \cdots + k_1x_e \equiv 0, \tag{15}$$

2502
so that the system is controlled to be asymptotically stable.

V. SIMULATION STUDY

The inverted pendulum shown in Fig. 1 is considered. The nonlinear dynamics is governed by

\[ ml^2 \ddot{z} + B \dot{z} + mgl \cos(z) = u \]  

where \( z \) denote the joint angle.

For the simplicity of simulation, assume all the parameters of (16) are

\[ m = \ell = B = g = 1. \]  

Then the equation (16) results in

\[ \dot{z} = -\dot{z} - \cos(z) + u. \]

In this case, \( x = [z, \dot{z}] \), and \( f(x) = -\dot{z} - \cos(z) \). Such nonlinear function is bounded by \( f(x) \leq f^U = 1 + |z| \).

The considering neural networks have the structure of \( N_{2, 20, 10, 1} \). The sigmoid functions mentioned before are used as activation functions of the node, but for the output node they are scaled properly according to the variation range of the nonlinear function \( f(x) \), that is,

\[ S_o(x) = K \tanh(z), \]

where \( S_o(x) \) and \( K \) are the activation function of the output node and scaling factor, respectively.

The following figures show the results of the identification and control using neural networks under the supervisory control while learning. We select \( V_M \) such that \( |z(t)| \leq \frac{\pi}{2} \). Fig. 2 depicts the regulation using the networks to the origin that is an unstable equilibrium point of the system. Fig. 3 shows the tracking control of the desired trajectory given by \( z_d(t) = \frac{\pi}{2} \sin(\frac{t}{2}) \). The result of the same desired trajectory tracked using only PD controller is shown in Fig. 4. We can see that the system performance is not satisfactory in this case while the neural network controller yields good performance.

VI. CONCLUSION

The identification of unstable systems using neural networks, which have been unsolved problem can be achieved by introducing the supervisory controller. The role of the supervisory controller is to obtain the training samples in the vicinity of unstable equilibrium states by making the system bounded within the region of specified. Based on these training samples the neural networks identify the nonlinear system, and then the controller is designed according the identified networks.

If we choose \( I = 1 \) in (11), then from (10) we can guarantee not only the boundedness of the system, but also the convergence of the error to zero. However we do not choose this strategy since the \( u_\ast \) is
usually very large. It is undesirable because the larger the control, the higher the implementation cost is. Further, in this case, we only know the upper bound of the nonlinear function $f(x)$. By introducing the neural network identifier and controller we can know more about the system characteristics. Therefore it is our strategy that the more we know about the system, the better we can control it and the lower the implementation cost is.

REFERENCES


![Desired (solid) and actual (dashed) trajectory](image)

Fig. 2: Regulation Using Neural Network Identification and Control

2504
Fig. 3: Tracking Control Using Neural Network Identification and Control

Fig. 4: Tracking Control Using only PD control