九州大学学術情報リポジトリ
Kyushu University Institutional Repository

Stability Analysis using Higher Order Derivatives of Universal Learning Networks

Yu, Yunqing
Dept. of Electrical and Electronic Systems Engineering, Graduate School of Information Science and Electrical Engineering, Kyushu University : Graduate Student

Hirasawa, Kotaro
Dept. of Electrical and Electronic Systems Engineering, Graduate School of Information Science and Electrical Engineering, Kyushu University

Hu, Jinglu
Dept. of Electrical and Electronic Systems Engineering, Graduate School of Information Science and Electrical Engineering, Kyushu University

Murata, Junichi
Dept. of Electrical and Electronic Systems Engineering, Graduate School of Information Science and Electrical Engineering, Kyushu University

https://doi.org/10.15017/1515684

出版情報：九州大学大学院システム情報科学紀要. 5 (2), pp.193-207, 2000-09-26. 九州大学大学院システム情報科学研究院
バージョン：
権利関係：

九州大学大学院
システム情報科学紀要
第5巻 第2号 平成12年9月

Research Reports on Information Science and
Electrical Engineering of Kyushu University
Vol.5, No.2, September 2000

Stability Analysis using Higher Order Derivatives of Universal Learning Networks

Yunqing YU* , Kotaro HIRASAWA**, Jinglu HU** and Junichi MURATA**

(Received June 16, 2000)

Abstract: Stability is one of the most important subjects in the control performance. As for the stability analysis of nonlinear systems in the past, Lyapunov method and linear stability methods based on linear approximation of nonlinear systems have been used. But, in many cases when Lyapunov method is applied, it is difficult to find suitable Lyapunov functions. And when using linear approximation, it can not be decided whether nonlinear systems are stable or not on the outside of the region where linear stability theory can be applied. Therefore, a new stability analysis method is required to develop, which can be easily applied to nonlinear systems. In this paper, stability analysis based on the higher order derivatives of Universal Learning Networks (ULNs) and its application to nonlinear systems are discussed. The stability analysis using ULNs is studied by the deviation of the system dynamics. The deviation of the system dynamics can be calculated by the higher order derivatives of ULNs. In this paper, simulations of an inverted pendulum balancing system are carried out. From the results of the simulations, it is shown that the inverted pendulum can be controlled effectively by ULNs. In addition, it is clarified that the stability of the nonlinear systems is easily analyzed by using the stability analysis of ULNs.

Keywords: Universal Learning Networks, Neural networks, Higher order derivatives calculation, Nonlinear system, Stability analysis

1. Introduction

The study on the stability of the dynamic systems goes back to 1868 when J.C. Maxwell studied the stability of the systems by linearizing the differential equations and checking the solutions of their characteristic polynominal[1]. In 1966, A.M. Lyapunov established his direct method which is a very general one based on the energy concept in Lagrange's preserved system[2]. In addition, the construction procedures of the Lyapnov function were extensively studied by Krasovskii[3], D.G. Schultz[4][5], and Vannelli-Vidyasagar[6]. But, it is still hard to find an appropriate Lyapunov function of the nonlinear dynamic systems under consideration. On the other hand, the linearized stability analysis has been also utilized to study the stability of nonlinear dynamic systems, which uses the well known Routh and Hurwitz method[7][8] to the linearized nonlinear dynamic systems. But, these methods can only guarantee the locally asymptotical stability, in other words, the asymptotically stable region can not obtained by the linearized method. This region is also hard to obtain even by Lyapunov direct method. Therefore, it has been highly required to develop a new stability analysis method which can be easy to handle and can calculate the asymptotically stable region of the nonlinear dynamic systems even if approximately.

In this paper, a new stability analysis method is proposed in order to overcome the above mentioned problems using the higher order derivatives of Universal Learning Networks (ULNs)[9]. Furthermore, an inverted pendulum control system is constructed by ULNs and its stability is studied by the proposed method. Now, Universal Learning Networks are proposed, as the name indicates, to provide a universal framework for the class of neural networks.

Since the first proposal of a neuron model by McCulloch and Pitts in 1940's, especially after the revitalization[10] of artificial neural networks in 1980's, a variety of neural networks have been devised and are now applied in many fields. The vast majority of neural networks in use are those networks whose parameters or weights are tuned by gradient-based supervised learning. This category includes feedforward networks or multi-layer perceptions[11], various types of recurrent neural networks[12][13], radial basis function networks[14], fuzzy neural networks[15], and some networks with special architectures, such as time delay neural networks[16]. These networks seemingly have different architectures and are trained by distinguishable

* Dept. of Electrical and Electronic Systems Engineering, Graduate Student

** Dept. of Electrical and Electronic Systems Engineering

training algorithms. In essence, however, they can be unified in a single framework in regard to both their architectures and learning algorithms.

Unification of a variety of neural network architectures and their learning algorithms is an objective of ULNs; this provides a consistent viewpoint for the various kinds of neural networks. However, there is another benefit that is expected for ULNs. Generalization of the architectures and the gradient-based algorithms do not only give a unified description but attain new abilities of the networks. Allowing high degrees of freedom in their architectures gives them more flexibilities and representing abilities, and therefore the ULNs can be useful and effective tools for modeling and controlling large-scale nonlinear complex systems including physical, social and economical phenomena. In addition to calculation of the first order derivatives of the signals flowing in the networks that are necessary in gradient-based learning, the generalized ULNs learning algorithm is equipped with a systematic mechanism that can calculate their second or higher order derivatives. This allows elaborate criterion functions to be used in their learning, which enables more sophisticated specification of the network performance[17].

As an application field, stability analysis of neural networks has been studied in recent years. Hirsch[18] noted the importance of global stability or global attractiveness and obtained a few sufficient conditions using Gershgorin's circle theorem. Kelly[19] applied the contraction mapping technique to obtain some sufficient conditions for global stability. Matsuoka[20] generalized some of Hirsch's and Kelly's results using a new Lyapunov function. Recently, Kaszkurewicz and Bhaya[21] proved that the diagonal stability of the interconnection matrix implies the existence and uniqueness of equilibrium and global stability of the equilibrium. Forti et al.[22] showed that the negative semidefiniteness of the interconnection matrix guarantees the global stability of a Hopfield network with a certain robustness property. Yang and Dillon[23] also obtained some sufficient conditions for the local exponential stability and for the existence and uniqueness of an equilibrium point. However, they failed to give any conditions for global stability, because there exists such a system that although it has a unique equilibrium point and it is locally (exponentially) stable, it may not be globally stable.

Before going into the details, it should be worth giving to the readers, as preliminary knowledge, the salient features of the ULNs and their differences from the existing alternatives.

ULNs consist of a number of inter-connected nodes. They may have any continuously differentiable functions in them, and each pair of nodes can be connected by multiple branches with arbitrary (positive, zero, or even negative) time delays. Networks having time delays of multiple-length can be converted to those having unit-length time delay only by inserting an appropriate number of additional nodes between the nodes. However, zero or negative time delays can not be represented by the unit time delays. ULNs including negative time delays between the nodes can be used to model the systems with prediction mechanism.

Learning of a ULNs is defined as determination of its optimal parameter values that minimize a criterion function. Generalized learning algorithms are derived based on two ideas. One is that the static networks are merely special dynamic networks, and the other is that like the method proposed by Williams[12][13], the derivatives of the node outputs can be obtained by either propagating the changes of the node outputs caused by the changes of the parameters through the network forward in time and space or propagating the change of the criterion function caused by the changes of the node outputs backward in time and space.

The most important feature of the learning of ULNs is the use of the higher order derivatives of the criterion function with respect to parameters. Buntine[17] summarized the methods for calculating the second order derivatives in feedforward neural networks but in ULNs a generalized calculation method of the nth order derivatives even for the recurrent networks is proposed.

The ULNs allow any nonlinear functions to be embeded in their nodes. Therefore, both a controlled system and its controller can be represented by a single ULNs. If the controlled system is unknown, its identification can also be done by the ULNs learning algorithm[24]. Then an optimal controller can be obtained by off-line gradient learning.

The ULNs controllers have a similarity with traditional nonlinear optimal controllers in the sense that both are constructed by minimizing criterion functions or performance indexes. However, the control laws obtained by the traditional controller design methods are, in most cases, feedforward control laws, and the control signals are given as functions of time. On the other hand, the ULNs controllers form feedback control schemes and their parame-

ters are determined by optimization. Therefore, although the ULNs controllers do not attain better performance indexes because of their structural constraint, they exhibit better robustness against external disturbances due to their feedback configuration.

The higher order derivative calculation mechanism of ULNs renders additional advantages to the ULNs controllers. As a criterion function, one may employ a standard control performance index, for example, tracking error of the controlled system. In addition to this, one can put other terms to their criterion functions. Let us assume that our additional term is the sensitivity of the standard criterion function with respect to the changes in the plant parameters. Minimization of the sensitivity by a gradient method requires the second order derivatives of the standard criterion function since the sensitivity itself is the first order derivative. This derivative can be calculated by the proposed mechanism. Minimizing this extended criterion function will result in a robust controlled system that is not much affected by changes in the plant parameters.

Stability analysis of using ULNs are studied by the deviation of dynamics. System is said to be orbital stable if the system dynamic approaches the original system dynamic even though the state change of the system occur at initial time. The deviation of the dynamics can be described by using Talor series expansion, and calculated by using the higher order derivatives of ULNs. As we can calculate the deviation of the dynamics systems caused by the initial change, we can determine the stability of the systems. Higher order derivatives of ULNs have already been derived by forward and backward propagation computing methods[25].

In other words, using this method, we can check if the system with big disturbances is stable or not. Furthermore, it is investigated whether the dynamics of the system are strongly stable or weakly stable by using the high order derivatives. Here, strongly stable means that the stable dynamics remain stable even when the big disturbances are inserted to the system. On the contrary, weakly stable means that the dynamics change to unstable if the system is disturbed strongly.

In this paper, an application of the proposed stability analysis method to the inverted pendulum balancing system is presented. The inverted pendulum balancing system is often used as an example of inherently unstable dynamic systems which can be designed by both modern and classic control techniques, as well as the learning control techniques of neural networks using supervised learning method or reinforcement learning method. There are $sin\theta$ and $cos\theta$ functions in the inverted pendulum system. It is assumed in the conventional methods that the change of θ is very small, that is $sin\theta \approx \theta$, and $cos\theta \approx 1$ hold. So we can only study the inverted pendulum with small change of θ when using the conventional methods. On the contrary, ULNs can allow any nonlinear function to be used in the system. Through the parameters training of the controller by the supervised learning methods, a nonlinear system which contains $sin\theta$ and $cos\theta$ functions can be controlled by ULNs.

This time, first order derivative, second order derivative and third order derivative are calculated, so, third order stability analysis are done in simulations.

In this paper, the stability analysis using the higher order derivatives of ULNs and its application to a nonlinear control system will be highlighted among the features and applications of ULNs. In Section 2, the structure and the learning algorithms of ULNs are described. The stability analysis using ULNs and calculation of higher order derivatives are presented in Section 3. As a numerical example, simulations of an inverted pendulum balancing system are carried out in Section 4. Section 5 presents the conclusions.

2. Universal Learning Networks

In this section, we discuss the structure and learning algorithms of ULNs.

2.1 Structure of ULNs

The purpose of ULNs is to provide a general framework for modeling and control of the complex systems widely found in the real world. It is generally recognized that any dynamic system can be described by using a set of related equations. Depending on prior knowledge available about the system, the equations may be fully known, partly known, or totally unknown. To model such dynamic systems, we introduce a learning network consisting of two kinds of elements: nodes and branches.

As show in **Fig.1**, Universal Learning Networks[25] consist of a number of nodes and branches for inter-connecting the nodes. The nodes may have any continuously differentiable nonlinear functions in them, and each pair of nodes can be connected to each other by multiple branches with arbitrary time delays. Multiple branches and their

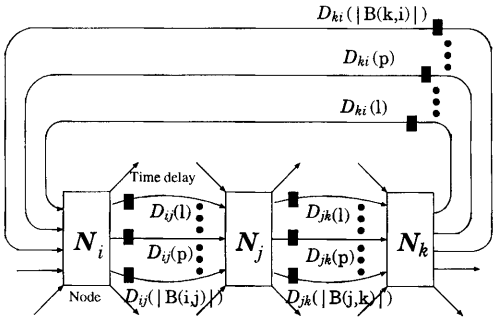

Fig.1 Structure of ULNs

arbitrary time delays can be effective to model the dynamic systems in a very compact network.

The generic equation that describes the behavior of ULNs is expressed as follows,

$$h_j(t) = f_j(\{h_i(t - D_{ij}(p)) | i \in JF(j), p \in B(i,j)\}, \{\lambda_m(t) \mid m \in M(j)\}, \{r_n(t) \mid n \in N(j)\}) \quad j \in J,\ t \in T. \tag{1}$$

where

$h_j(t)$	:	output value of node j at time t,
$r_n(t)$	:	value of nth external input variable at time t,
$\lambda_m(t)$	:	value of mth parameter at time t,
f_j	:	nonlinear function of node j,
$D_{ij}(p)$	:	time delay of pth branch from node i to node j,
J	:	set of nodes $\{j\}$,
$JF(j)$	:	set of nodes which are connected to node j,
N	:	set of external input variables $\{n\}$,
M	:	set of parameters $\{m\}$,
$B(i,j)$	:	set of branches from node i to node j,
T	:	discrete set of sampling instants.

The ULNs operate on a discrete-time basis. Each pair of nodes i and j may have multiple branches between them i.e. set $B(i,j)$ may contain more than one elements, and parameters $\lambda_m(t)$ can be time-varying. Functions $f_j(\cdot)$ that govern the operation of the nodes can be any continuously differentiable functions; typically, sigmoid functions can be employed. And in that case, Eq. (1) can be expressed specifically as

$$h_j(t) = f_j(\alpha_j(t)) = \frac{1}{1 + e^{-\phi_j \alpha_j(t)}},$$
$$\alpha_j(t) = \sum_{i \in JF(j)} \sum_{p \in B(i,j)} w_{ij}(p) h_i(t - D_{ij}(p)) + \theta_j, \tag{2}$$

where

$w_{ij}(p)$	:	weight parameter of pth branch from node i to node j,
θ_j	:	threshold parameter of node j,
ϕ_j	:	slope parameter of node j.

Therefore, ULNs can be recognized as a framework for modeling more general systems in the sense that after being processed by each function at each node, the output signal of the node is transferred to another node through multiple branches with arbitrary time delay, then it is again processed by another node with its function.

2.2 Learning of ULNs

The criterion function E is defined as a differentiable function of relevant node outputs and parameters at appropriate time instants:

$$E = E(h_r(s), \lambda_m(s)), \quad r \in J_0, m \in M_0, s \in T_0. \tag{3}$$

where

J_0	=	$\{r \mid h_r$ is related to evaluation of criterion function$\}$;
M_0	=	$\{m \mid \lambda_m$ is related to evaluation of criterion function$\}$;
T_0	=	$\{s \mid$time instant s is related to evaluation of criterion function$\}$.

A typical choice of E is the sum of errors between the network outputs and their desired values. And function E defined above is a generalized function in the sense that almost all the criterion functions used for neural network training are in the form of E.

Learning of λ_m in ULNs can be realized by minimizing the criterion function E based on the gradient method:

$$\lambda_m(t) \leftarrow \lambda_m(t) - \gamma \frac{\partial^\dagger E}{\partial \lambda_m(t)}, \tag{4}$$

where γ is the learning coefficient assigned a small positive value and $\frac{\partial^\dagger E}{\partial \lambda_m(t)}$ is the ordered derivative defined by Werbos[26] which means the change of the criterion function E caused by the change of a

parameter λ_m at time t with other variables being fixed,

$$\frac{\partial^\dagger E}{\partial \lambda_m(t)} = \sum_{r\in J_0}\sum_{s\in T_0}\left[\frac{\partial E}{\partial h_r(s)}\frac{\partial^\dagger h_r(s)}{\partial \lambda_m(t)}\right] + \frac{\partial E}{\partial \lambda_m(t)} \tag{5}$$

where, t is a certain specified time instant and $\lambda_m(t)$ is the value of parameter at time t.

3. Stability Analysis using Higher Order Derivatives of ULNs

3.1 Definition of Asymptotic Orbital Stability

As shown in **Fig.2**, the system is said to be orbital stable if the system dynamic approaches the original system dynamic even though the state change of the system occur at initial time.

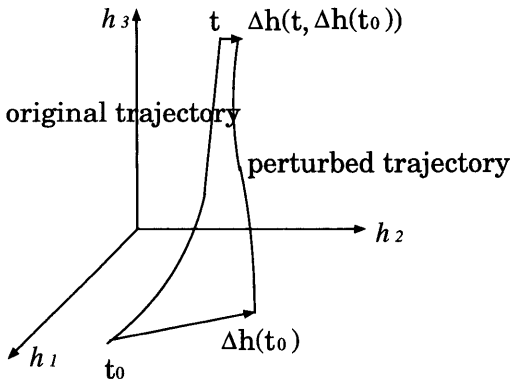

Fig.2 Deviation of the dynamics

So, the asymptotic orbital stability in the large using ULNs is defined as follows. For any initial orbital change $\Delta h(t_0)$

$$\text{if} \qquad \lim_{t\to\infty} \Delta h(t, \Delta h(t_0)) = 0, \tag{6}$$

then, system dynamic is said to be asymptotic orbital stable in the large.

where

$$\Delta h(t, \Delta h(t_0)) = \sqrt{\sum_{r\in J} \Delta h(r, t, \Delta h(t_0))^2},$$

$\Delta h(t, \Delta h(t_0))$: orbital change at time t caused by the initial orbital change $\Delta h(t_0)$ at time t_0,

$\Delta h(r, t, \Delta h(t_0))$: change of the output node r at time t caused by the initial orbital change $\Delta h(t_0)$ at time t_0,

J : set of suffixes of system nodes.

Using Talor series expansion, we can calculate the deviation of the dynamic at time s approximately as follows,

$$\begin{aligned}
&\Delta h(r, t, \Delta h(t_0)) \\
&\approx \sum_{r1\in J} \frac{\partial^\dagger h_r(t)}{\partial h_{r1}(t_0)} \times \Delta h_{r1}(t_0) \\
&+ \frac{1}{2}\sum_{r1\in J}\sum_{r2\in J} \frac{\partial^{\dagger 2} h_r(t)}{\partial h_{r1}(t_0)\partial h_{r2}(t_0)} \times \Delta h_{r1}(t_0)\Delta h_{r2}(t_0) \\
&+ \frac{1}{6}\sum_{r1\in J}\sum_{r2\in J}\sum_{r3\in J} \frac{\partial^{\dagger 3} h_r(t)}{\partial h_{r1}(t_0)\partial h_{r2}(t_0)\partial h_{r3}(t_0)} \times \Delta h_{r1}(t_0)\Delta h_{r2}(t_0)\Delta h_{r3}(t_0) \\
&+ \cdots + \frac{1}{i!}\sum_{r1\in J}\cdots\sum_{ri\in J} \frac{\partial^{\dagger i} h_r(t)}{\partial h_{r1}(t_0)\cdots\partial h_{ri}(t_0)} \times \Delta h_{r1}(t_0)\cdots\Delta h_{ri}(t_0),
\end{aligned} \tag{7}$$

where

$\Delta h(t_0) = [\Delta h_1(t_0), \cdots, \Delta h_{|J|}(t_0)]$.

So, as we can calculate the deviation of the dynamics system caused by the initial change by using the higher order derivatives of ULNs, we can determine the stability of the system. Higher order derivatives of ULNs have been already derived by forward and backward propagation computing methods[25].

3.2 Calculation of Nth Order Derivatives of ULNs

Since the deviation of the dynamics can be calculated by $\frac{\partial^{\dagger i} h_r(t)}{\partial h_{r1}(t_0)\cdots\partial h_{ri}(t_0)}$ from Eq.(7), then, in this section, how to calculate $\frac{\partial^{\dagger i} h_r(t)}{\partial h_{r1}(t_0)\cdots\partial h_{ri}(t_0)}$ is explained.

Although higher order derivatives of ULNs can be calculated by both forward propagation and backward propagation, nth order derivatives calculation method by forward propagation is summarized here.

$$P_1(r,t,h_{r1}(t_0)) = \frac{\partial^{\dagger} h_r(t)}{\partial h_{r1}(t_0)},$$

$$\begin{aligned} &P_1(k,t,h_{r1}(t_0)) \\ &= \sum_{j\in JF(k)} \sum_{p\in B(j,k)} \Big[\frac{\partial h_k(t)}{\partial h_j(t-D_{jk}(p))} \\ &\quad \times P_1(j,t-D_{jk}(p),h_{r1}(t_0))\Big] + \frac{\partial h_k(t)}{\partial h_{r1}(t_0)}, \\ &\qquad k\in J \quad t\in T, \end{aligned} \tag{8}$$

$$P_2(r,t,h_{r1}(t_0),h_{r2}(t_0)) = \frac{\partial^{\dagger 2} h_r(t)}{\partial h_{r1}(t_0)\partial h_{r2}(t_0)},$$

$$\begin{aligned} &P_2(k,t,h_{r1}(t_0),h_{r2}(t_0)) \\ &= \sum_{j\in JF(k)} \sum_{p\in B(j,k)} \Big[\frac{\partial^{\dagger}\left(\frac{\partial h_k(t)}{\partial h_j(t-D_{jk}(p))}\right)}{\partial h_{r2}(t_0)} \\ &\quad \times P_1(j,t-D_{jk}(p),h_{r1}(t_0))\Big] \\ &+ \sum_{j\in JF(k)} \sum_{p\in B(j,k)} \Big[\frac{\partial h_k(t)}{\partial h_j(t-D_{jk}(p))} \\ &\quad \times P_2(j,t-D_{jk}(p),h_{r1}(t_0),h_{r2}(t_0))\Big], \\ &\qquad k\in J \quad t\in T, \end{aligned} \tag{9}$$

$$\begin{aligned} &P_3(r,t,h_{r1}(t_0),h_{r2}(t_0),h_{r3}(t_0)), \\ &= \frac{\partial^{\dagger 3} h_r(t)}{\partial h_{r1}(t_0)\partial h_{r2}(t_0)\partial h_{r3}(t_0)} \end{aligned}$$

$$\begin{aligned} &P_3(k,t,h_{r1}(t_0),h_{r2}(t_0),h_{r3}(t_0)) \\ &= \sum_{j\in JF(k)} \sum_{p\in B(j,k)} \Big[\frac{\partial^{\dagger 2}\left(\frac{\partial h_k(t)}{\partial h_j(t-D_{jk}(p))}\right)}{\partial h_{r2}(t_0)\partial h_{r3}(t_0)} \\ &\quad \times P_1(j,t-D_{jk}(p),h_{r1}(t_0))\Big] \\ &+ \sum_{j\in JF(k)} \sum_{p\in B(j,k)} \Big[\frac{\partial^{\dagger}\left(\frac{\partial h_k(t)}{\partial h_j(t-D_{jk}(p))}\right)}{\partial h_{r2}(t_0)} \\ &\quad \times P_2(j,t-D_{jk}(p),h_{r1}(t_0),h_{r3}(t_0))\Big] \\ &+ \sum_{j\in JF(k)} \sum_{p\in B(j,k)} \Big[\frac{\partial^{\dagger}\left(\frac{\partial h_k(t)}{\partial h_j(t-D_{jk}(p))}\right)}{\partial h_{r3}(t_0)} \\ &\quad \times P_2(j,t-D_{jk}(p),h_{r1}(t_0),h_{r2}(t_0))\Big] \\ &+ \sum_{j\in JF(k)} \sum_{p\in B(j,k)} \Big[\frac{\partial h_k(t)}{\partial h_j(t-D_{jk}(p))} \\ &\quad \times P_3(j,t-D_{jk}(p),h_{r1}(t_0),h_{r2}(t_0),h_{r3}(t_0))\Big], \\ &\qquad k\in J \quad t\in T, \end{aligned} \tag{10}$$

.....

.....

$$\begin{aligned} &P_n(r,t,h_{r1}(t_0),h_{r2}(t_0),\cdots,h_{rn}(t_0)), \\ &= \frac{\partial^{\dagger n} h_r(t)}{\partial h_{r1}(t_0)\partial h_{r2}(t_0)\cdots\partial h_{rn}(t_0)} \end{aligned}$$

$$\begin{aligned} &P_n(k,t,h_{r1}(t_0),h_{r2}(t_0),\cdots,h_{rn}(t_0)) \\ &= \sum_{j\in JF(k)} \sum_{p\in B(j,k)} \Big[\frac{\partial^{\dagger n-1}\left(\frac{\partial h_k(t)}{\partial h_j(t-D_{jk}(p))}\right)}{\partial h_{r2}(t_0)\partial h_{r3}(t_0)\cdots\partial h_{rn}(t_0)} \\ &\quad \times P_1(j,t-D_{jk}(p),h_{r1}(t_0))\Big] \\ &+ \sum_{j\in JF(k)} \sum_{p\in B(j,k)} \Big[\frac{\partial^{\dagger n-2}\left(\frac{\partial h_k(t)}{\partial h_j(t-D_{jk}(p))}\right)}{\partial h_{r3}(t_0)\partial h_{r4}(t_0)\cdots\partial h_{rn}(t_0)} \\ &\quad \times P_2(j,t-D_{jk}(p),h_{r1}(t_0),h_{r2}(t_0)) \\ &+ \cdots\cdots\cdots\cdots \\ &+ \frac{\partial^{\dagger n-2}\left(\frac{\partial h_k(t)}{\partial h_j(t-D_{jk}(p))}\right)}{\partial h_{r2}(t_0)\partial h_{r3}(t_0)\cdots\partial h_{rn-1}(t_0)} \\ &\quad \times P_2(j,t-D_{jk}(p),h_{r1}(t_0),h_{rn}(t_0))\Big] \\ &+ \sum_{j\in JF(k)} \sum_{p\in B(j,k)} \Big[\frac{\partial^{\dagger n-3}\left(\frac{\partial h_k(t)}{\partial h_j(t-D_{jk}(p))}\right)}{\partial h_{r4}(t_0)\partial h_{r5}(t_0)\cdots\partial h_{rn}(t_0)} \\ &\quad \times P_3(j,t-D_{jk}(p),h_{r1}(t_0),h_{r2}(t_0),h_{r3}(t_0)) \\ &+ \cdots\cdots\cdots\cdots \\ &+ \frac{\partial^{\dagger n-3}\left(\frac{\partial h_k(t)}{\partial h_j(t-D_{jk}(p))}\right)}{\partial h_{r2}(t_0)\partial h_{r3}(t_0)\cdots\partial h_{rn-2}(t_0)} \\ &\quad \times P_3(j,t-D_{jk}(p),h_{r1}(t_0),h_{rn-1}(t_0),h_{rn}(t_0))\Big] \\ &+ \cdots\cdots\cdots\cdots \\ &+ \sum_{j\in JF(k)} \sum_{p\in B(j,k)} \Big[\frac{\partial^{\dagger}\left(\frac{\partial h_k(t)}{\partial h_j(t-D_{jk}(p))}\right)}{\partial h_{r2}(t_0)} \\ &\quad \times P_{n-1}(j,t-D_{jk}(p),h_{r1}(t_0),h_{r3}(t_0),\cdots, \\ &\quad h_{rn}(t_0)) \\ &+ \cdots\cdots\cdots\cdots \\ &+ \frac{\partial^{\dagger}\left(\frac{\partial h_k(t)}{\partial h_j(t-D_{jk}(p))}\right)}{\partial h_{rn}(t_0)} \\ &\quad \times P_{n-1}(j,t-D_{jk}(p),h_{r1}(t_0),h_{r2}(t_0),\cdots, \\ &\quad h_{rn-1}(t_0))\Big] \end{aligned}$$

$$+\sum_{j\in JF(k)}\sum_{p\in B(j,k)}\Big[\frac{\partial h_k(t)}{\partial h_j(t-D_{jk}(p))} \times P_n(j,t-D_{jk}(p),h_{r1}(t_0),h_{r2}(t_0),\cdots, h_{rn}(t_0))\Big],$$
$$k\in J \quad t\in T, \tag{11}$$

where

$JF(k)$: set of suffixes of nodes whose outputs are connected to node k

$B(j,k)$: set of branches from node j to node k,

Initial values are as follows

$$P_1(k,t_0,h_{r_1}) = \begin{cases} 1.0 & when \;\; k=r_1 \\ 0 & when \;\; k\neq r_1. \end{cases}$$

$$P_i(k,t_0,h_{r1},\cdots,h_{ri}) = 0 \quad i=2,3,\cdots,n.$$

3.3 Stability Analysis using ULNs

From Eq.(7), we can see the deviation of the dynamics can be calculated by the higher order derivatives of ULNs, so, the system stability can be determined as follows.

1. Unstable Orbital

If there exists $r, r_1 \in J$ such that

$$\lim_{t\to\infty}\Big|\frac{\partial^\dagger h_r(t)}{\partial h_{r1}(t_0)}\Big| = \infty,$$

then $\lim_{t\to\infty}\Delta h(t,\Delta h(t_0)) = \infty,$

therefore the system is unstable.

That is, the system dynamics are away from the original system dynamics no matter what initial orbital changes are small or large.

2. Asymptotic Stable Orbital in the Large

If $\lim_{t\to\infty}\Big|\frac{\partial^\dagger h_r(t)}{\partial h_{r1}(t_0)}\Big| = 0$

$$\lim_{t\to\infty}\Big|\frac{\partial^{\dagger 2} h_r(t)}{\partial h_{r1}(t_0)\partial h_{r2}(t_0)}\Big| = 0$$

$$\cdots\cdots$$

$$\lim_{t\to\infty}\Big|\frac{\partial^{\dagger i} h_r(t)}{\partial h_{r1}(t_0),\cdots\partial h_{ri}(t_0)}\Big| = 0$$

for any $r, r_1, \cdots r_i \in J$,

then $\lim_{t\to\infty}\Delta h(t,\Delta h(t_0)) = 0,$

therefore the system is asymptotic stable in the large.

That is, the system dynamic approaches the original system dynamic approximately no matter what initial orbital changes are small or large.

3. Strongly and Weakly Asymptotic Stable Orbital

From Eq.(7) and characteristic of Talor series expansion, we can obtain the following: (1) the system dynamics are effected by lower order derivatives when initial orbital changes are small; (2) the system dynamics are effected by higher order derivatives when initial orbital changes are large.

So, the stable orbital excluding asymptotic stable orbital in the large is classified as follows.

(3a) Strongly stable orbital

The system dynamic approaches the original system dynamic even though the initial change is large.

That is, there exists large l satisfying that

$$\operatorname*{Max}_{r,r_1,\cdots,r_i\in J,t\in T}\Big|\frac{\partial^{\dagger i} h_r(t)}{\partial h_{r1}(t_0)\cdots\partial h_{ri}(t_0)}\Big|$$

is small for all $i < l$.

In this case the system is strongly stable;

(3b) Weakly stable orbital

The system dynamic deviates from the original system dynamic even when the initial change is small.

That is, there exists small l satisfying that

$$\operatorname*{Max}_{r,r_1,\cdots,r_i\in J,t\in T}\Big|\frac{\partial^{\dagger i} h_r(t)}{\partial h_{r1}(t_0)\cdots\partial h_{ri}(t_0)}\Big|$$

is large for a certain $i > l$

In this case the system is weakly stable;

3.4 Region Suitable for Linear Approximation Method

As was stated in Section 1, when using stability analysis of linear approximation method, it can not be decided whether nonlinear systems are stable or not on the outside of the region where linear stability theory can be applied. Therefore, a method is required in order to determine the region suitable for linear approximation.

The method is described as follows:

Let us use the following notations ,

$$\underset{r,r1,\cdots,ri\in J,t\in T}{\text{Max}} \left| \frac{\partial^{\dagger i} h_r(t)}{\partial h_{r1}(t_0)\cdots\partial h_{ri}(t_0)} \right| = H(i),$$

$$\underset{ri\in J}{\text{Max}} |\Delta h_{ri}(t_0)| = R,$$

then, the ith order value of Talor series expansion is satisfied as follows.

$$\begin{aligned} &\left| \frac{1}{i!} \sum_{r1\in J} \cdots \sum_{ri\in J} \frac{\partial^{\dagger i} h_r(t)}{\partial h_{r1}(t_0)\cdots\partial h_{ri}(t_0)} \right. \\ &\quad \left. \times \Delta h_{r1}(t_0)\cdots\Delta h_{ri}(t_0) \right| \\ &\le \frac{1}{i!} \sum_{r1\in J} \cdots \sum_{ri\in J} \left| \frac{\partial^{\dagger i} h_r(t)}{\partial h_{r1}(t_0)\cdots\partial h_{ri}(t_0)} \right| \\ &\quad \times |\Delta h_{r1}(t_0)|\cdots|\Delta h_{ri}(t_0)| \\ &\le \frac{1}{i!} J^i H(i) R^i. \end{aligned}$$

So, the maximum first order value of Talor series expansion, that is, maximum linear approximation value, becomes $JH(1)R$.

Compared $\frac{1}{i!}J^iH(i)R^i$ with $JH(1)R$, for small $\varepsilon > 0$,

if $\dfrac{1}{i!}\dfrac{J^iH(i)R^i}{JH(1)R} \le \varepsilon \quad i = 2, 3, \cdots$

then $J^iH(i)R^i \le \varepsilon i! JH(1)R$,

As a result, R satisfying the following Eq.(12) could be regarded as the diameter of region suitable for linear approximation.

$$R \le \underset{i=2,3,\cdots}{\text{Min}} \frac{1}{J} \sqrt[i-1]{\frac{i!\varepsilon H(1)}{H(i)}} \tag{12}$$

According to the definition of strongly and weakly stable orbital which were descried in Section 3.3, the following results are obtained and shown in **Fig.3**,

(1) In case of strongly stable orbital, $H(i)_{i<l}$ is small, then R is big.

(2) In case of weakly stable orbital, $H(i)_{i>l}$ is big, then R is small.

4. Numerical Example

In this section, an application of first, second and third order stability analysis to an inverted pendulum system is presented, where it is designed by using ULNs.

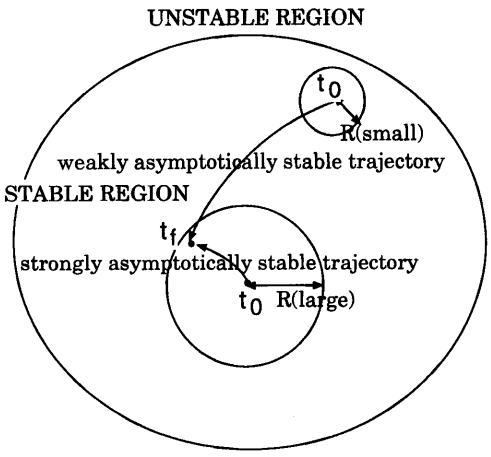

Fig.3 The diameter of the region suitable for linear approximation

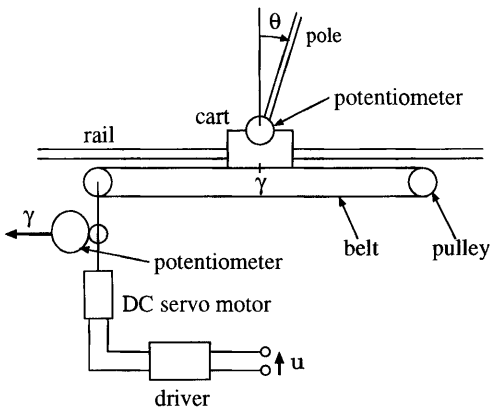

Fig.4 Structure of inverted pendulum

4.1 Controlled System and Controller

The structure of the inverted pendulum is shown in **Fig.4**. The bottom of the pole is hinged to a cart that travels along a finite length track to its right or left. Both the cart and the pole can move only in the vertical plane; that is, each has only one degree of freedom. The horizontal position of the cart's center(in meter) and the angle of the pole from an upright position(in radian) are represented by r and θ, respectively. The only control action u is the output from a DC servo motor which makes the track of the cart to move toward its left or right.

The nonlinear inverted pendulum system is described as follow,

$$\begin{aligned} \frac{d^2r}{dt^2} &= \frac{\alpha}{M}u - \frac{F}{M}\frac{dr}{dt} \\ \frac{d^2\theta}{dt^2} &= \frac{1}{J+ml^2}[mglsin\theta - ml\frac{d^2r}{dt^2}cos\theta - c\frac{d\theta}{dt}], \end{aligned} \tag{13}$$

where,

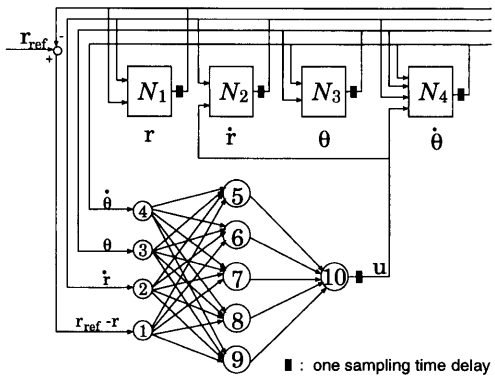

Fig.5 Layered type control model of the inverted pendulum using ULNs

M is mass of the cart and m is mass of the pole;
α: coefficient for transforming voltage to torque;
F, c: friction factors for cart and pole, respectively;
J: moment of inertia of the pole;
g: gravity acceleration.

Because ULNs operate on a discrete time basis, introducing

$$h_1(t) = r(t),\ h_2(t) = \dot{r}(t),$$
$$h_3(t) = \theta(t),\ h_4(t) = \dot{\theta}(t),$$

Eq.(15) can be transformed into discrete type equations as follows:

$$\begin{aligned} h_1(t) &= a_{11}h_1(t-1) + a_{21}h_2(t-1) \\ h_2(t) &= a_{22}h_2(t-1) + b_1u(t) \\ h_3(t) &= a_{33}h_3(t-1) + a_{43}h_4(t-1) \\ h_4(t) &= a_{24}cos(h_3(t-1))h_2(t-1) + a_{34}sin(h_3(t-1)) \\ &\quad + a_{44}h_4(t-1) + b_2cos(h_3(t-1))u(t) \end{aligned} \tag{14}$$

where

$$a_{11} = a_{33} = 1.0,\ a_{21} = a_{43} = \Delta t,$$
$$a_{22} = 1 - \frac{F}{M}\Delta t,\ a_{24} = \frac{mlF}{(J+ml^2)M}\Delta t,$$
$$a_{34} = \frac{mgl}{J+ml^2}\Delta t,\ a_{44} = 1 - \frac{c}{J+ml^2}\Delta t,$$
$$b_1 = \frac{\alpha}{M}\Delta t,\ b_2 = -\frac{ml\alpha}{(J+ml^2)M}\Delta t.$$

In addition, when the inverted pendulum system is controlled by feed-forward layered neural networks, in other words, control input u is constructed by neural networks, the controller are incorporated in the ULNs to form a ULN-based feedback nonlinear system as shown in **Fig.5**.

The middle layer of the neural networks controller is made of sigmoid functions which is expressed specifically as the following,

$$\begin{aligned} h_j(t) &= f_j(\alpha_j(t)) = \frac{1-e^{-\alpha_j(t)}}{1+e^{-\alpha_j(t)}}, \\ \alpha_j(t) &= \sum_{i \in JF(j)} w_{ij} \cdot h_i(t-1), \\ i &= 1,2,3,4, \\ j &= 5,6,7,8,9. \end{aligned} \tag{15}$$

The output layer of the neural networks controller is composed of a linear function as,

$$\begin{aligned} h_j(t) &= \sum_{i \in JF(j)} w_{ij} \cdot h_i(t-1), \\ i &= 5,6,7,8,9, \\ j &= 10. \end{aligned} \tag{16}$$

where, w_{ij} is weight parameter.

4.2 Simulation

M=3.7[kg], α=7.77[n/v], F=23.31[kg/s], l=0.11[m], m=0.007[kg], J=0.00103[kgm^2], c=0.0007[kgm^2/s], g=9.8[m/s^2] are used in simulations, and the reference of the cart (r_{ref}) is 0.2[m] and the reference of the angle (θ_{ref}) is 0.0[rad], respectively.

The criterion function E to achieve the desired dynamics of the system is defined as follows,

$$\begin{aligned} E = \frac{1}{2}\Big[&\sum_{t\in T}\{Q_{11}(r_{ref} - h_1(t))^2\} + Q_{12}(h_2(t_f))^2 \\ &+ \sum_{t\in T}\{Q_{13}(h_3(t))^2\} + Q_{14}(h_4(t_f))^2 \\ &+ \sum_{t\in T}\{R_1(h_{10}(t))^2\}\Big], \end{aligned} \tag{17}$$

where

T	:	10000
t_f	:	final instant (=10[sec])
the number of training	:	10000.

In order to minimize the criterion function, the parameter should be trained by a gradient method according to Eq.(4) in section 2.1.

$$\lambda_m(t) \leftarrow \lambda_m(t) - \gamma\frac{\partial^\dagger E}{\partial \lambda_m(t)}, \tag{18}$$

As shown in **Fig.6**, six cases are studied in a strongly and weakly stable region. The initial positions and the target values of the cart and angle of the pole are assumed as **Table 1**.

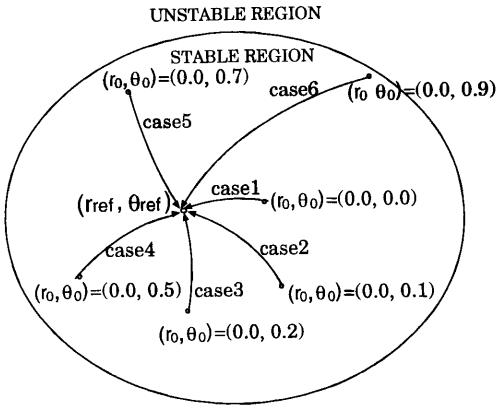

Fig.6 Six cases in a stable region

Table 1 Simulation conditions regarding initial values and target values

	initial		reference	
	r_0[m]	θ_0[rad]	r_{ref}[m]	θ_{ref}[rad]
case1	0.0	0.0	0.2	0.0
case2	0.0	0.1	0.2	0.0
case3	0.0	0.2	0.2	0.0
case4	0.0	0.5	0.2	0.0
case5	0.0	0.7	0.2	0.0
case6	0.0	0.9	0.2	0.0

Learning are carried out in the six cases. In order to obtain suitable parameters in controller through learning, the coefficient $Q_{11} \sim Q_{14}$ and R_1 of the criterion function are assumed as **Table 2**. And the learning coefficient of γ is assumed to be 0.00001 in *case*1 $\sim$ *case*5 and 0.000001 in *case*6, respectively.

Table 2 Simulation conditions regarding coefficients of the criterion function

	Q_{11}	Q_{12}	Q_{13}	Q_{14}	R_1
case1	10.0	10.0	10.0	10.0	1.0
case2	10.0	10.0	10.0	10.0	1.0
case3	10.0	10.0	10.0	10.0	1.0
case4	1.0	1.0	1.0	1.0	0.1
case5	1.0	1.0	1.0	1.0	0.01
case6	1.0	1.0	1.0	1.0	0.01

Using Eq.(8), Eq.(9) and Eq.(10), we can calculate the first, second and third order derivative of six cases. Here, three variables $h_{r1}(t_0)$, $h_{r2}(t_0)$ and $h_{r3}(t_0)$ are regarded as the same variable $\theta(t_0)$, in other words, $P1$, $P2$ and $P3$ are first, second and third order derivatives for the change of θ at initial time. The value of P_1, P_2 and P_3 of r and θ are calculated as follows.

$$P_1(k,t,h_3(t_0)) = \sum_{j\in JF(k)} \left[\frac{\partial h_k(t)}{\partial h_j(t-1)} \times P_1(j,t-1,h_3(t_0))\right] + \frac{\partial h_k(t)}{\partial h_3(t_0)} \qquad k \in 1-10, \quad t \in T, \tag{19}$$

$$P_2(k,t,h_3(t_0),h_3(t_0)) = \sum_{j\in JF(k)} \left[\frac{\partial^\dagger\left(\frac{\partial h_k(t)}{\partial h_j(t-1)}\right)}{\partial h_3(t_0)} \times P_1(j,t-1,h_3(t_0))\right] + \sum_{j\in JF(k)} \left[\frac{\partial h_k(t)}{\partial h_j(t-1)} \times P_2(j,t-1,h_3(t_0),h_3(t_0))\right] \qquad k \in 1-10, \quad t \in T, \tag{20}$$

$$P_3(k,t,h_3(t_0),h_3(t_0),h_3(t_0)) = \sum_{j\in JF(k)} \left[\frac{\partial^{\dagger 2}\left(\frac{\partial h_k(t)}{\partial h_j(t-1)}\right)}{\partial^2 h_3(t_0)} \times P_1(j,t-1,h_3(t_0))\right] + 2\sum_{j\in JF(k)} \left[\frac{\partial^\dagger\left(\frac{\partial h_k(t)}{\partial h_j(t-1)}\right)}{\partial h_3(t_0)} \times P_2(j,t-1,h_3(t_0),h_3(t_0))\right] + \sum_{j\in JF(k)} \left[\frac{\partial h_k(t)}{\partial h_j(t-1)} \times P_3(j,t-1,h_3(t_0),h_3(t_0),h_3(t_0))\right] \qquad k \in 1-10, \quad t \in T, \tag{21}$$

The initial values are set as follows,

$$P_1(k,t_0,h_3) = \begin{cases} 1.0 & when \;\; k=3 \\ 0.0 & when \;\; k \neq 3, \end{cases}$$

$$P_2(k,t_0,h_3,h_3) = 0,$$
$$P_3(k,t_0,h_3,h_3,h_3) = 0.$$

According to Eq.(12), the calculation of the region suitable for linear approximation method is described as follows,

$$R \leq \underset{i=2,3}{\mathrm{Min}} \frac{1}{J} \sqrt[i-1]{\frac{i!\varepsilon H(1)}{H(i)}}$$

Because only the change of $\theta(t_0)$ of the dynamic is considered, $|J|$ is equal to one, therefore,

$$R \leq \mathrm{Min}\Big(\frac{2\varepsilon H(1)}{H(2)},\ \sqrt[2]{\frac{6\varepsilon H(1)}{H(3)}}\Big). \tag{22}$$

4.3 Results

Learning are carried out in the six cases mentioned before. The system dynamics after training and the value of P_1, P_2 and P_3 are shown in **Fig.7** ~ **Fig.24**. Let's divide up the six cases to two parts. Strongly stable phenomena are to be seen in *part*1 which includes *case*1, *case*2 and *case*3, *part*2 which includes *case*4, *case*5 and *case*6 tends to show weak stability.

Using the parameters in the controller obtained through learning, the output values of r and θ reach the target values in all six cases, shown in **Fig.7** ~ **Fig.12**. But the process where output value reaches the target value are not same in six cases because the initial values are not the same.

The value of P_1, P_2 and P_3 are determined by the output behaviors. (see **Table 3**)

In *part*1, the values of P_1 in *case*1, *case*2 and *case*3 are almost the same, but the values of P_2 and P_3 attend to become big values in the order of *case*1, *case*2 and *case*3. Furthermore, in *part*1, $\mathrm{Max}|P_1| > \mathrm{Max}|P_2| > \mathrm{Max}|P_3|$ is satisfied as shown in **Fig.13** ~ **Fig.18**.

On the contrary, in *part*2, that is, in *case*4, *case*5 and *case*6, as can be seen **Fig.19** ~ **Fig.24**, $\mathrm{Max}|P_1| < \mathrm{Max}|P_2| < \mathrm{Max}|P_3|$ is satisfied and the difference of P_1 values among three cases is relatively small, while that of P_3 values becomes enormous.

So, we can obtain the following results.

(1) In *part*1, the system is strongly stable, therefore, it is sufficient to use only the first order derivative.

(2) In *part*2, the system is weakly stable, therefore, it is suggested to use the third order derivative.

In order to calculate the diameter of region suitable for linear approximation, $\mathrm{Max}|P_1|(=H(1))$, $\mathrm{Max}|P_2|(=H(2))$ and $\mathrm{Max}|P_3|(=H(3))$ are required.

Table 3 The maximum value of P_1, P_2 and P_3 in six cases

	θ		
	Max $\|P_1\|$	Max $\|P_2\|$	Max $\|P_3\|$
case1	1.000	0.022	0.003
case2	1.000	0.247	0.122
case3	1.000	0.534	0.530
case4	1.000	2.058	5.839
case5	1.424	5.832	39.362
case6	2.945	38.179	678.204

As only $\theta(0)$ is changed at initial time in this simulation, it is assumed ε is equal to 0.01, so, the diameter of θ in the six cases are shown in **Table 4**.

Table 4 The diameter of θ in six cases

	θ		
	$\frac{2\varepsilon H(1)}{H(2)}$	$\sqrt[2]{\frac{6\varepsilon H(1)}{H(3)}}$	R_{Max}[rad]
case1	0.909	4.472	0.909
case2	0.081	0.701	0.081
case3	0.037	0.336	0.037
case4	0.010	0.101	0.010
case5	0.005	0.047	0.005
case6	0.002	0.016	0.002

From **Table 4**, we can obtain the following result: the more stronger stable orbital is, the more bigger diameter of region suitable for linear approximation is obtained. Furthermore, even in the inherently unstable inverted pendulum system, the use of the second order derivative is enough for the study of the stability of the system.

5. Conclusion

Stability analysis using the higher order derivatives of ULNs and its application to nonlinear control system have been discussed. With regards to the architecture and learning algorithms, a class of neural networks is unified in the framework of ULNs, and salient features added. In an architectural aspect, ULNs have higher degree of freedom: arbitrary continuously differentiable functions in the nodes, multiple branches and arbitrary time delays. The ULNs training algorithm is equipped with a mechanism for calculating the higher order derivatives as a noble feature that has not been possessed by common neural networks.

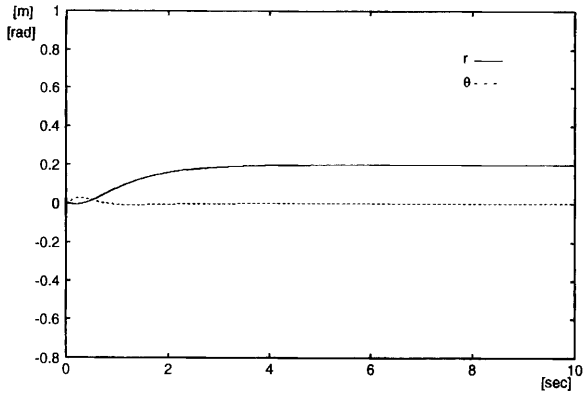

Fig.7 Output values of r and θ in *case*1

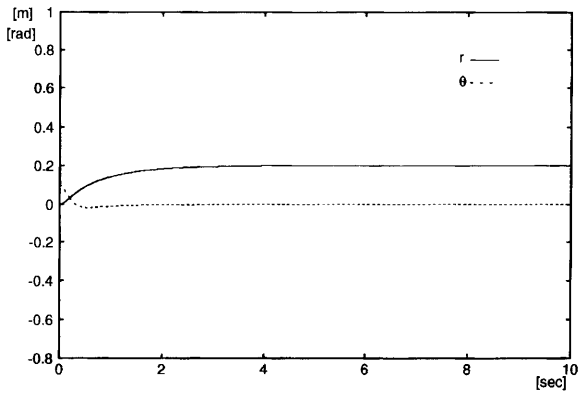

Fig.8 Output values of r and θ in *case*2

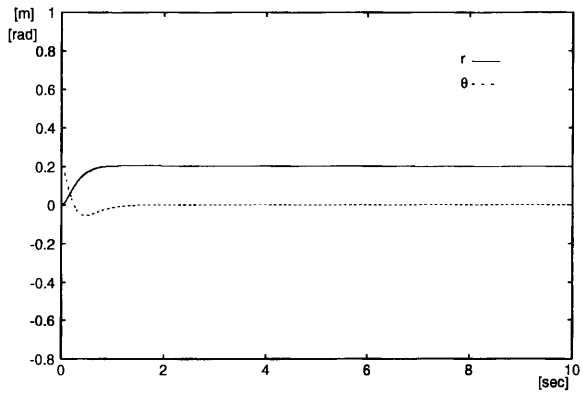

Fig.9 Output values of r and θ in *case*3

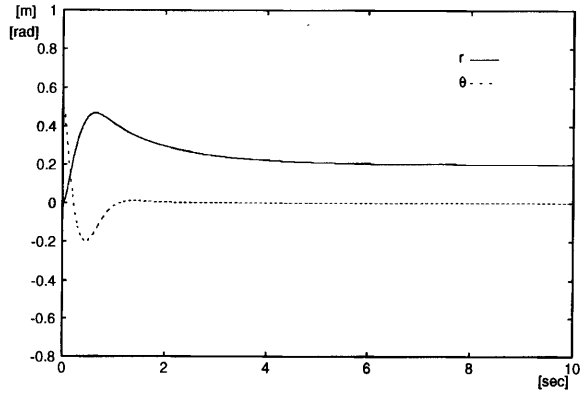

Fig.10 Output values of r and θ in *case*4

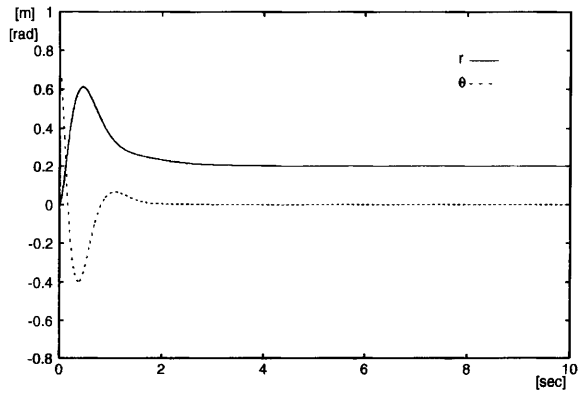

Fig.11 Output values of r and θ in *case*5

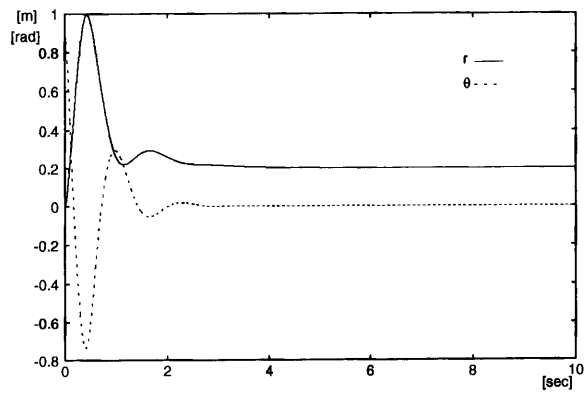

Fig.12 Output values of r and θ in *case*6

Therefore, ULNs are useful for modeling and controlling large scale complicated systems such as industrial plants, economic, social and life phenomena, and the ULNs can be applied widely to many kinds of problems.

In this paper, a new method for stability analysis of nonlinear systems is also proposed which is based on the higher order derivatives of ULNs. Using the proposed method for stability analysis of nonlinear systems can avoid to find Lyapunov function, and determine the region suitable for linear approximation.

From the results of the simulations, it is shown that the inverted pendulum can be controlled effectively by ULNs. In addition using the proposed stability analysis, it is clarified that the stability of a nonlinear systems is easily analyzed even when the big disturbances are inputed to the systems.

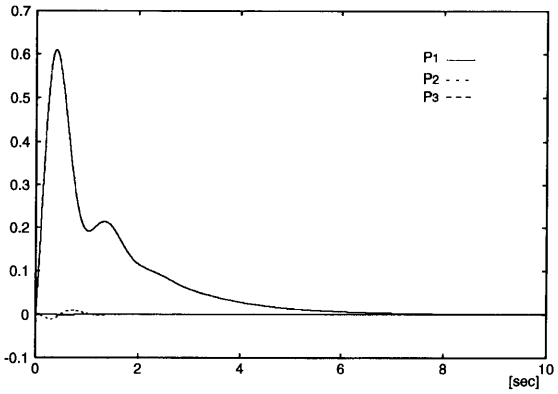

Fig.13 The values of P_1, P_2 and P_3 for r in $case1$

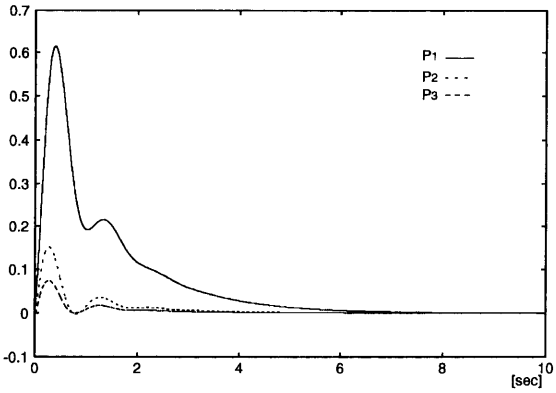

Fig.14 The values of P_1, P_2 and P_3 for r in $case2$

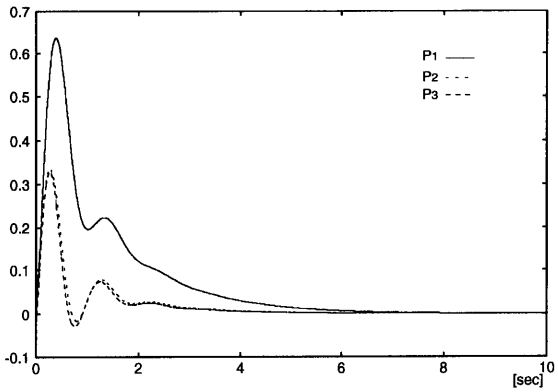

Fig.15 The values of P_1, P_2 and P_3 for r in $case3$

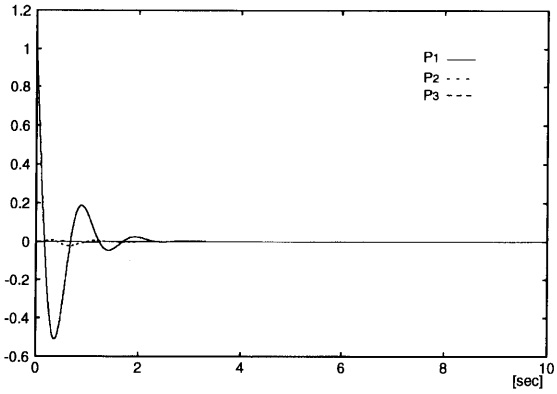

Fig.16 The values of P_1, P_2 and P_3 for θ in $case1$

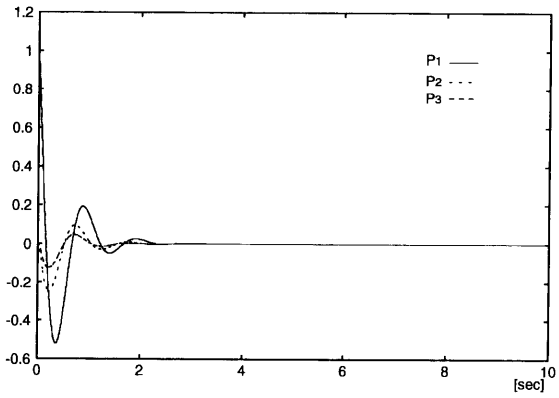

Fig.17 The values of P_1, P_2 and P_3 for θ in $case2$

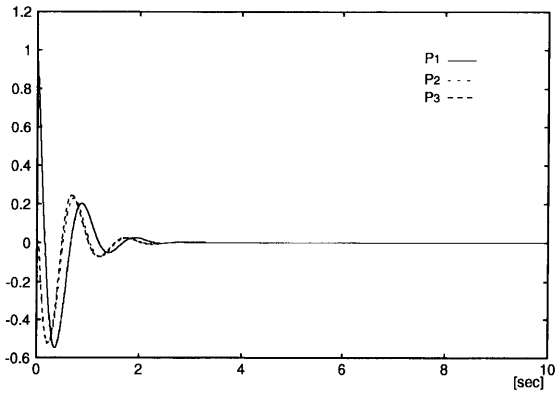

Fig.18 The values of P_1, P_2 and P_3 for θ in $case3$

References

1) J.C. Maxwell : **On Governers**, Proc. R. Soc. London, Vol. 16, 1868.

2) A.M. Lyapunov : **Stability of Motion**, Academic Press, 1966.

3) N.N. Krasovski : **On the Stability in the Large of a System of Nonlinear Differential Equations**, Prikl. Math. i Mekh., Vol. 18, 1954.

4) D.G. Schultz and J.E. Gibson : **The Variable Gradient Method for Generating Lyapunov Functions**, Trans. AIEE, Application and Industory, Vol. 81, Pt. 2, 1962.

5) S.G. Margolis and W.G. Vogt : **Control Engineering Applications of Zubov's Construction Procedure Lyapunov Functions**, IEEE Tran., AC-8, April, 1963.

6) A. Uannelli and M. Vidyasagar : **Maximum Lyapunov Functions and Domains of Attraction for Autonomous Nonlinear Systems**, Automatica, Vol. 21, No. 1, 1985.

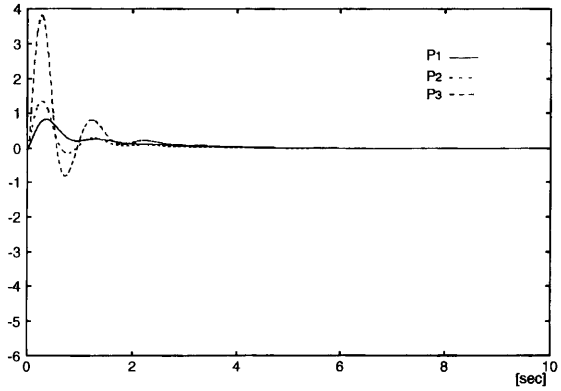

Fig.19 The values of P_1, P_2 and P_3 for r in *case*4

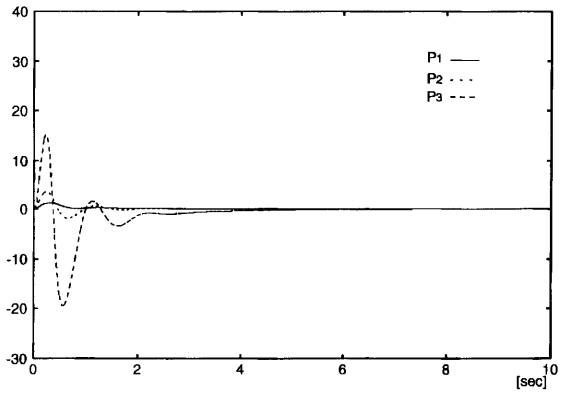

Fig.20 The values of P_1, P_2 and P_3 for r in *case*5

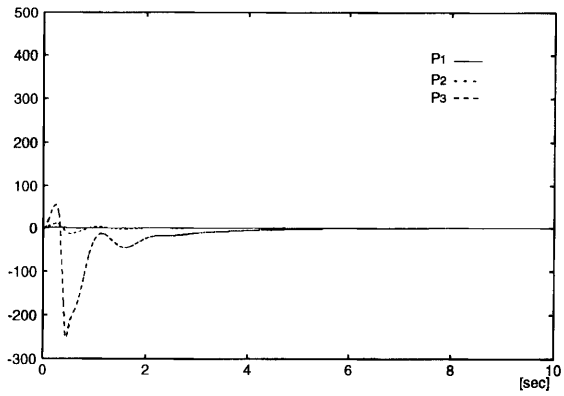

Fig.21 The values of P_1, P_2 and P_3 for r in *case*6

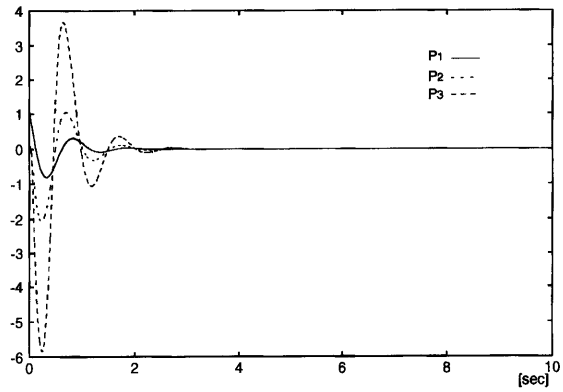

Fig.22 The values of P_1, P_2 and P_3 for θ in *case*4

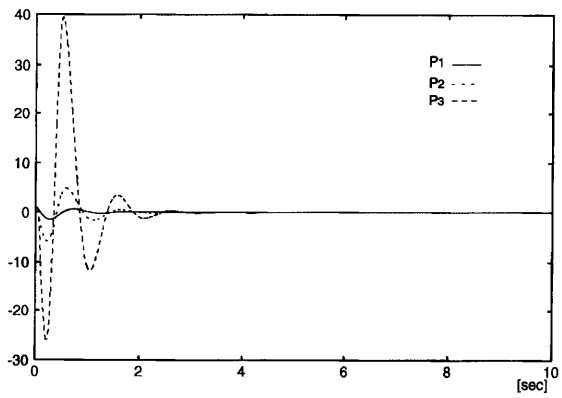

Fig.23 The values of P_1, P_2 and P_3 for θ in *case*5

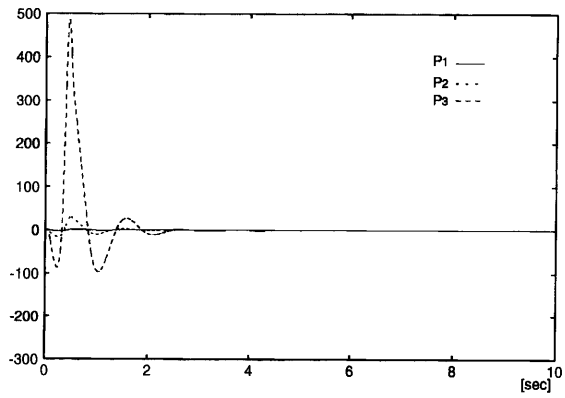

Fig.24 The values of P_1, P_2 and P_3 for θ in *case*6

7) S. Barnett :**Polynominals and Linear Systems**, Marcel Dekker, Inc., 1983.

8) F.R. Gantmacher :**The Theory of Matrices**, Chelsea Publishing Company, Volume one, Volume two, 1960.

9) K. Hirasawa, M. Ohbayashi, M. Koga :**Stability Theory of Universal Learning Networks**, Proc. of 1996 IEEE International Conference on Systems, Man, and Cybernetics, 1996

10) D.E Rumelhart, G.E Hinton and R.J Williams :**Learning Representation by Back-Propagating Errors**, Nature, Vol. 323, pp.533-536, 1986

11) F. Rosenblat :**The Perceptron, a Probabilistic Model for Information Storage and Organization in the Brain**, Psychological Review, Vol. 65, pp.386-408, 1958

12) R.J Williams and D. Zipser :**A Learning Algorithm for continually Running Fully Recurrent Neural Networks**, Neural Computation, Vol. 1, No.2, pp.270-280, 1989

13) R.J Williams and D. Zipser :**Gradient-Based Learn-**

ing Algorithms for Recurrent Connectionist Networks, College of Computer Science Technical Report, Northeastern University, NU-CCS-90-9, pp.1-45, 1990

14) T. Poggio and F. Girosi :**Networks for Approximation and Learning**, Proc.IEEE, Vol. No. 78, 9, pp.1481-1497, 1990

15) M.M Gupta and J. Qi :**On fuzzy neuron models**, Proc of Int. Joint Conf. on Neural Networks, Vol.2, pp.431-436, 1991

16) D.T Lin, J.E Dayhoff and P.A Ligomenides :**Trajectory Production with the Adaptive Time-Delay Neural Network**, Neural Networks, Vol.8, No.3, pp.447-461, 1995

17) W. L. Buntine and A. S. Weigend :**Computing Second Derivatives in Feed-Forward Network: A Riview**, IEEE Trans. on Neural Networks, Vol. 5, No. 3, 1994

18) M. W. Hirsch :**Convergent activation dynamics in continuous time networks**, Neural Networks, Vol. 2, pp. 331-349, 1989

19) D. G. Kelly, :**Stability in contractive nonlinear neural networks**, IEEE Trans. Biomed. ENg., Vol. 3, No. 3, pp. 231-242, 1990

20) K. Matsuoka, :**Stability conditions for nonlinear continuous neural networks with asymmetric connection weights**, Neural Networks, Vol. 5, pp. 495-500, 1992

21) A. Fuez, V. Protopopsecu, and J. Barhen, :**On the stability, storage capacity, and design of nonlinear continuous neural networks**, IEEE Trans. Syst., Man and Cybern., Vol. 18, No. 1, pp. 80-87, 1988

22) M. Forti, S. Manetti, and M. Marini, :**Necessary and sufficient conditions for absolute stability of neural networks**, IEEE Trans. Circuits Syst.: Fundamental Theory Appl., Vol. 41, No. 7, pp. 491-494, 1994

23) H. Yang and T. S. Dillon, :**Exponential stability and oscillation of Hopfield graded response neural network**, IEEE Trans. Neural Networks, Vol. 5, No. 5, pp. 719-729, 1994

24) M. Han, K. Hirasawa, M. ohbayashi and M.Fujita :**Modeling Dynamic Systems Using Universal Learning Network**, Proc of IEEE Conference on System, Man and Cybernetics, pp.1172-1177, 1996

25) K. Hirasawa, X. Wang, J. Murata, J. Hu and C. Jin :**Universal Learning Networks and Its Application to Chaos Control**, Neural Networks, Vol-13, No. 2, pp.239-253, 2000

26) P.Werbos :**Beyond regression: New Tools for Prediction and Analysis in the Behavior Science**, Ph.D. dissertation, Harvard University, 1974