A HYBRID NEURO-INVERSE CONTROL APPROACH WITH KNOWLEDGE-BASED NONLINEAR SEPARATION FOR INDUSTRIAL NONLINEAR SYSTEM WITH UNCERTAINTIES

T. Zhang, M. Nakamura†

† Department of Advanced Systems Control Engineering, Graduate School of Science and Engineering, Saga University, Honjomachi, Saga 840-8502, Japan. (E-mail:zhangtao@cntl.ee.saga-u.ac.jp).

Keywords: Hybrid neuro-inverse control, knowledge-based nonlinear separation, industrial nonlinear system, high-precision contour control, industrial articulated robot arm

Abstract

This paper presents a general methodology of controller design by the hybrid neuro-inverse control with the knowledge-based nonlinear separation for industrial nonlinear systems. In industrial nonlinear systems, various kinds of uncertainties may cause serious deterioration of system performances. Unfortunately, these uncertainties are usually difficult to identify and compensate from the entire system point of view. With using the knowledge-based nonlinear separation, nonlinear dynamics of a nonlinear system is possibly separated into the input-output nonlinear static part and the nonlinear dynamic part to form a nonlinear separation structure. Hence, partial nonlinear factors of the nonlinear system are described by the input-output nonlinear static part. Uncertainties in the nonlinear system are bounded in the nonlinear dynamic part. In the proposed hybrid neuro-inverse control method, the input-output nonlinear static part is controlled by an inverse controller. A neurocontroller with a rigidly defined and trained neural network using available prior knowledge of the nonlinear system is constructed for the control of the nonlinear dynamic part. With respect to some cases, a PID controller is supplementarily employed to reduce the influence from big uncertainties in the nonlinear dynamic part. Owing to using the knowledge-based nonlinear separation and a PID controller, the neurocontroller is only needed to control a part of the original nonlinear dynamics of industrial nonlinear systems contaminated by uncertainties. The structure of the neural network employed in the neurocontroller becomes simpler and the consumption of time in training is reduced. Meanwhile, system performances of the nonlinear system can be improved by the proposed method. Based on this method, high-precision contour control of industrial articulated robot arm was solved. It demonstrated the generality, practicality and significant potential of this method for realizing the high-performance control of industrial nonlinear systems.

1 Introduction

In industrial nonlinear systems, compensation of uncertainties is one of the important means to improve system performances. Conventionally, an industrial nonlinear system is modeled by a series of differential equations or discrete equations, in which uncertainties are simplified as constants or typical statistical distributions. Based on the model of the system, uncertainties are always compensated by the states or output feedback. However, if high system performances are required in industrial nonlinear systems, such as high precision, high speed, etc., the impact of uncertainties, as the main reason of deterioration of system performances, must be taken into account seriously and compensated precisely.

To reduce the influence of uncertainties upon system performances of industrial nonlinear systems, a vast number of new approaches on the control of nonlinear systems by neural networks has increased significantly in recent years [3, 4, 7, 8]. The use of the learning ability of neural networks helps the design of controllers to be rather flexible, especially when system dynamics are complicated and highly nonlinear with many uncertainties. In order to enable the designed nonlinear controller with neural networks acceptable to industries, almost all above approaches attempted to narrow down the gap between theory and applications. When employing a neural network controller (neurocontroller) for an actual industrial nonlinear system to improve system performances, one of the crucial problems is how to design a powerful neurocontroller with a simple structure and lesser time in training. Concerning this problem, several approaches have been exploited, such as the design of various architectures of neurocontrollers, development of hybrid control methods, improvement of learning algorithms, on-line training, use of prior knowledge, etc.

However, many kinds of neurocontrollers with simple structures always need a tremendous amount of training data in order to include all possible operating conditions. Hence, the training time undertaken is lengthy. They are difficult to apply to a wide range of real-time control problems [8]. Moreover, this problem may become more serious when these neurocontrollers try to control complex and highly nonlinear systems with uncertainties.

The method proposed in this paper is a hybrid neuro-inverse control approach with the knowledge-based nonlinear separation. It is simple and well suited for industrial applications meanwhile it can improve system performances remarkably. Additionally, it combined various kinds of approaches mentioned above for improving system performances. With this

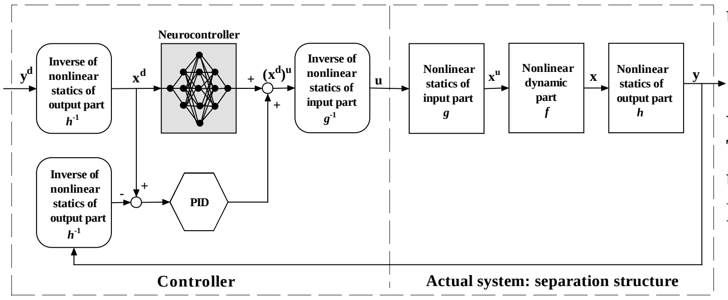

Figure 1: Nonlinear separation structure and controller of hybrid neuro-inverse control approach with knowledge-based nonlinear separation

method, a complex industrial nonlinear system with uncertainties is separated into several parts using prior knowledge. A hybrid control method is adopted for controlling different parts, where a neurocontroller is mainly responsible for the compensation of the nonlinear dynamic part contaminated by uncertainties and an inverse controller is to compensate the nonlinear static part. In addition, this method emphasizes the use of prior knowledge for the controller construction. Therefore, a powerful neurocontroller with a simple structure and lesser time in training can be implemented. Comparing with the neurocontroller for the entire nonlinear system control, the structure of neurocontroller designed by the proposed method for the nonlinear dynamic part becomes simpler. And this method has great potential for realizing real-time control. Besides, different types of neural network architectures, learning algorithms, etc., can be integrated into this method for various kinds of industrial nonlinear systems.

2 Hybrid Neuro-Inverse Control Approach with Knowledge-based Nonlinear Separation

2.1 Nonlinear separation structure

The knowledge-based nonlinear separation method is to model a nonlinear system based on the prior knowledge obtained from a mathematical model and actual field data of the system [6]. With this method, a nonlinear system can be described by a nonlinear separation structure as illustrated in the right side of Fig.1. The nonlinear separation structure consists of three parts: input nonlinear static part, nonlinear dynamic part and output nonlinear static part. The input nonlinear static part conducts the nonlinear transformation of the control input. The output nonlinear static part represents the nonlinear relation between the output variables and the states of the system under the steady state. They can be defined by a known mathematical model or an approximation model identified by the actual data. The other part of original nonlinear dynamics contaminated by uncertainties is regarded as the nonlinear dynamic part of the system.

The input nonlinear static part expressed by the function g with the vector control input $\mathbf{u}$ is

$$\mathbf{x}^u(t) = g[\mathbf{u}(t)], \tag{1}$$

where $\mathbf{x}^u$ denotes the output of the input nonlinear static part. The nonlinear dynamic part, expressed by the function f with the state variables $\mathbf{x}$, the output of the input nonlinear static part $\mathbf{x}^u$, the exogenous input $\mathbf{w}$ and the uncertain factor v, is

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{x}^u(t), \mathbf{w}(t), v(t)). \tag{2}$$

The output nonlinear static part expressed by the function h with the vector output $\mathbf{y}$ is

$$\mathbf{y}(t) = h[\mathbf{x}(t)]. \tag{3}$$

This model is the improvement of the nonlinear separation model proposed in [6] by replacing the linear dynamic part with the nonlinear dynamic part.

2.2 Inverse controller of nonlinear static part

In the knowledge-based nonlinear structure, the input nonlinear static function g and the output nonlinear static function h are supposed to describe rigorously by a mathematical model without any uncertainties. Therefore, the control of the input-output nonlinear static part is carried out by the inverse control method.

In general, the controller of the input nonlinear static part is

$$\mathbf{u}(t) = g^{-1}[(\mathbf{x}^d)^u(t)], \tag{4}$$

where $(\mathbf{x}^d)^u$ is the control input of the input nonlinear static part. The controller of the output nonlinear static part is

$$\mathbf{x}^d(t) = h^{-1}[\mathbf{y}^d(t)], \tag{5}$$

where $\mathbf{y}^d$ and $\mathbf{x}^d$ denote the input and the output of the controller of the output nonlinear static part, respectively. The above inverse controller can be derived from nonlinear static models as long as nonlinear static models are reversible. If nonlinear static models are irreversible, there are two approaches for defining inverse controllers. One is to transform the nonlinear static model into a reversible form through the appropriate approximation. Another approach is to employ the inverse process of the knowledge-based nonlinear separation to define an inverse controller of the nonlinear static part [6].

Besides, since the nonlinear static part is defined under the steady state of the system, the nonlinear static part is hard to be contaminated by uncertainties. It means that uncertainties are merely laid in the nonlinear dynamic part. Although it can not guarantee that an input-output nonlinear static part could represent the overall nonlinear factors of the original nonlinear system, the input-output nonlinear static part is at least taking away some of the nonlinear factors of the original nonlinear system. Therefore, the nonlinear factors of the original nonlinear system can be reduced. The compensation process of uncertainties in the nonlinear dynamic part can undoubtedly become easy to carry out. Furthermore, system performances can be possibly improved.

2.3 Neurocontroller of nonlinear dynamic part

In an industrial nonlinear system, PID control is an effective approach to reduce the big influence of uncertainties in the system. Therefore, in the proposed method, a PID controller is also employed like conventional control methods to set up the controller as illustrated in the left side of Fig.1 in order to reduce the big influence of uncertainties in the nonlinear dynamic part. As we above mentioned, the function of PID control is insufficient to achieve the high performance control perfectly. Therefore, besides the installation of PID controller for the nonlinear dynamic part, a neurocontroller is designed for the nonlinear dynamic part in the proposed method to improve system performances.

A neurocontroller mainly consists of a neural network for controlling the nonlinear dynamic part f contaminated by uncertainties. Since the learning capability of neural networks, it has proved that any nonlinear functions can be approximated to any desired accuracy by a two-layer neural network with a sufficiently large number of neurons, over a compact domain of a finite dimensionally formed space [1]. Therefore, the features of adaptiveness and robustness of the system with respect to uncertainties can be improved and thereby the high performance control of the system can be achieved. Besides, since most of the nonlinear factors have been brought away by the nonlinear static part and the big influence of uncertainties in the nonlinear dynamic part has been reduced by the PID control, it becomes much easier for neural networks to realize the desired high performance. The parameters of neural networks also converge more easily during the process of training.

The nonlinear dynamic part is as given in (2). If the output of the controller of the nonlinear dynamic part is supposed as $(\mathbf{x}^d)^u$, the controller of the nonlinear dynamic part is expressed by

$$(\mathbf{x}^d)^u(t) = \bar{f}(\mathbf{x}^d(t), \dot{\mathbf{x}}^d(t), \mathbf{w}(t), \upsilon(t)). \quad (6)$$

If concerning the effect of the PID control further, the equation (6) is then changed into

$$(\mathbf{x}^d)^u(t) = \hat{f}(\mathbf{x}^d(t), \dot{\mathbf{x}}^d(t), \omega(t)), \quad (7)$$

where ω denotes the remained uncertainties in the nonlinear dynamic part with a PID controller.

Many kinds of neural networks can be adopted to design the neurocontroller of the nonlinear dynamic part. For instance, a three-layer feedforward neural network, as one of the typical neural networks, is adopted here. The inputs of the neural network are the objective values of elements of the state $\mathbf{x}^d$ and its differential $\dot{\mathbf{x}}^d$, as well as the remained uncertainties ω. The output of the neural network is the objective control signal $(\mathbf{x}^d)^u$, which will be put into the controller of the input nonlinear static part. Linear nodes with linear functions can be used for the input layer and the output layer of the neural network. The units in the hidden layer can be defined by some typical nodes, such as gaussian nodes, sigmoid nodes, etc. Therefore, the neurocontroller for the nonlinear dynamic part of the system is described by

$$(\mathbf{x}^d)^u(t) = \sum_{i=1}^{N} w_i p_i(\mathbf{x}^d(t), \dot{\mathbf{x}}^d(t), \omega(t)), \quad (8)$$

where p_i denotes the activation function of the neural network unit and N denotes the number of hidden nodes.

In order to define appropriate neural network components and enable the parameters of neural network nodes to converge easily, prior knowledge on the nonlinear dynamic part of the system is very helpful to determine the structure and the initial parameters of the neural network as well, even only with a given linear dynamic model. When defining initial parameters of a neural network, such as weights of units, coefficients of activation functions of units, etc, they can be given arbitrarily or determined with a known inverse nonlinear dynamic model or an identification model built by the actual data of the system. Some typical learning algorithms can be chosen for the above neural networks, such as backpropagation method, etc.

3 High-Precision Contour Control of Industrial Articulated Robot Arm

Contour control is one of the important working patterns of industrial articulated robot arm (IARA). High-precision and high-speed movement is the expected performances of contour control in industry. IARA contains many uncertainties from its mechanism and working circumstance during the working process, such as friction, gravity, interference between robot links, parameter errors in the adopted model, disturbance, and so on. In order to realize the high-precision contour control of IARA with the high-speed movement, it is necessary to reduce the influence of these uncertainties. Therefore, a hybrid neuro-inverse nonlinear control with the knowledge-based nonlinear separation is employed for the contour control of IARA.

3.1 Nonlinear separation structure of industrial articulated robot arm

In this research, a two-joint IARA is concentrated. The nonlinear separation structure of this robot arm is constructed based on the physical model of mechanism. Fig.2 illustrates the system architecture of one link of IARA with a neurocontroller. From this figure, the nonlinear separation structure of IARA can be defined as below.

The function g of the input nonlinear static part as (1) is defined by the inverse kinematics which refers to the space transformation from the cartesian coordinate to the joint coordinate, as

$$\begin{cases} x_1^u = \arctan\frac{u_1}{u_2} + \arccos\left[\frac{(l_1)^2-(l_2)^2+u_1^2+u_2^2}{2l_1\sqrt{u_1^2+u_2^2}}\right] \\ x_2^u = \pi - \arccos\left[\frac{(l_1)^2+(l_2)^2-u_1^2-u_2^2}{2l_1 l_2}\right], \end{cases} \quad (9)$$

where $u_j (j = 1, 2)$ is the trajectory in the cartesian coordinate, $x_j^u (j = 1, 2)$ is the trajectory in the joint coordinate, $l_j (j = 1, 2)$ is the length of the robot link.

Similarly, the function h of the output nonlinear static part as (3) is defined by the kinematics which refers to the space transformation from the joint coordinate to the cartesian coordinate, as

$$\begin{cases} y_1 = l_1 \sin x_1 + l_2 \sin(x_1 + x_2) \\ y_2 = l_1 \cos x_1 + l_2 \cos(x_1 + x_2), \end{cases} \quad (10)$$

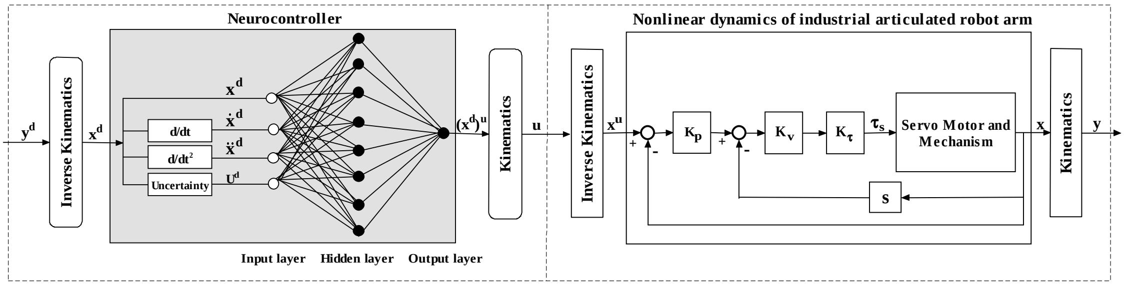

Figure 2: System architecture of industrial articulated robot arm

where $x_j(j = 1,2)$ is the trajectory in the joint coordinate, $y_j(j = 1,2)$ is the trajectory in the cartesian coordinate.

The other part of the original nonlinear dynamics is defined as the nonlinear dynamic part f in (2) of the nonlinear separation structure of IARA, which can be described by Euler-Lagrange equations as

$$\begin{cases} [(k_1)^2 J_1^M + m_1(l_1^G)^2 + m_2\{(l_1)^2 + (l_2^G)^2\} \\ +m_1(l_1)^2/3 + m_2(l_2)^2/3]\ddot{x}_1/(K_1^p K_1^v K_1^\tau) \\ +\dot{x}_1/K_1^p + x_1 + U_1 = x_1^u \\ \{(k_2)^2 J_2^M + m_2(l_2^G)^2 + m_2(l_2)^2/3\}\ddot{x}_2 \\ /(K_2^p K_2^v K_2^\tau) + \dot{x}_2/K_2^p + x_2 + U_2 = x_2^u, \end{cases} \quad (11)$$

where $x_j^u(j = 1,2)$ is the control signal in the joint coordinate, $x_j,\ \dot{x}_j,\ \ddot{x}_j(j = 1,2)$ are the actual joint trajectory, velocity and acceleration, $K_j^p(j = 1,2)$ is the position loop gain, $K_j^v(j = 1,2)$ is the velocity loop gain, $K_j^\tau(j = 1,2)$ is the torque coefficient, $J_j^M(j = 1,2)$ is the inertial coefficient, $m_j(j = 1,2)$ is the mass of robot link, $k_j(j = 1,2)$ is the gear ratio, $l_j(j = 1,2)$ is the length of robot link, $l_j^G(j = 1,2)$ is the length between the axis and the center of gravity. $U_j(j = 1,2)$ is the uncertainty. Due to the installation of position loop gain, velocity loop gain as well as torque coefficient, a PI controller, as a semi-closed loop, has been given for the control of the nonlinear dynamic part. Therefore, the influence of the original uncertainties of IARA has been reduced and the uncertainty U in (11) denotes the remained uncertainty of IARA.

3.2 Neuro-inverse controller of industrial articulated robot arm

As illustrated in the left side of Fig.2, a hybrid neuro-inverse controller for each link of IARA contains three parts. Based on the inverse control, the controller g^{-1} in (4) of the input nonlinear static part of IARA is defined by the kinematics as

$$\begin{cases} u_1 = l_1 \sin{(x_1^d)^u} + l_2 \sin{((x_1^d)^u + (x_2^d)^u)} \\ u_2 = l_1 \cos{(x_1^d)^u} + l_2 \cos{((x_1^d)^u + (x_2^d)^u)}, \end{cases} \quad (12)$$

where $(x_j^d)^u(j = 1,2)$ is the objective control input in the joint coordinate, $u_j(j = 1,2)$ is the control input in the cartesian coordinate.

Similarly, the controller h^{-1} in (5) of the output nonlinear static part is defined by the inverse kinematics as

$$\begin{cases} x_1^d = \arctan\frac{y_1^d}{y_2^d} + \arccos[\frac{(l_1)^2-(l_2)^2+(y_1^d)^2+(y_2^d)^2}{2l_1\sqrt{(y_1^d)^2+(y_2^d)^2}}] \\ x_2^d = \pi - \arccos[\frac{(l_1)^2+(l_2)^2-(y_1^d)^2-(y_2^d)^2}{2l_1 l_2}], \end{cases} \quad (13)$$

where $y_j^d(j = 1,2)$ is the objective trajectory in the cartesian coordinate, $x_j^d(j = 1,2)$ is the objective trajectory in the joint coordinate.

For controlling the nonlinear dynamic part of the nonlinear separation structure of IARA, a gaussian neural network (GNN) is adopted [5]. For each link of IARA, there is a gaussian neurocontroller including one GNN. With the method of the selection of GNN structure [9], each GNN has four input nodes which represent the realizable objective joint trajectory x^d, velocity $\dot{x}^d$, acceleration $\ddot{x}^d$ and the compensation value of the uncertainty U^d. GNN also has eight hidden nodes and one output node. If defining the initial parameters of GNN by the known Euler-Lagrange equations (11) [9], the gaussian neurocontroller for the first link of IARA can be expressed by

$$(x^d)^u = 1.321\left(w_1\frac{\ddot{x}_1^d}{\ddot{X}_{1max}} + w_2\frac{\dot{x}_1^d}{\dot{X}_{1max}} + w_3\frac{x_1^d}{X_{1max}} + w_4\frac{U_1^d}{U_{1max}}\right), \quad (14)$$

where $w_1 = 0.757C_1\ddot{X}_{1max}$, $w_2 = 0.757\dot{X}_{1max}/K_1^p$, $w_3 = 0.757X_{1max}$, $w_4 = 0.757U_{1max}$ and $C_1 = ((k_1)^2J_1^M + m_1(l_1^G)^2 + m_2\{(l_1)^2 + (l_2^G)^2\} + m_1(l_1)^2/3 + m_2(l_2)^2/3)/(K_1^pK_1^vK_1^\tau)$. Similarly, the gaussian neurocontroller for the second link of IARA can be expressed by

$$(x^d)^u = 1.321\left(w_1\frac{\ddot{x}_2^d}{\ddot{X}_{2max}} + w_2\frac{\dot{x}_2^d}{\dot{X}_{2max}} + w_3\frac{x_2^d}{X_{2max}} + w_4\frac{U_2^d}{U_{2max}}\right), \quad (15)$$

where $w_1 = 0.757C_2\ddot{X}_{2max}$, $w_2 = 0.757\dot{X}_{2max}/K_2^p$, $w_3 = 0.757X_{2max}$, $w_4 = 0.757U_{2max}$ and $C_2 = ((k_2)^2J_2^M + m_2(l_2^G)^2 + m_2(l_2)^2/3)/(K_2^pK_2^vK_2^\tau)$.

A well-defined trial in the actual system is made to get the compensation values of uncertainties. In the trial, the realizable objective joint trajectory for training and its velocity, acceleration as well as small arbitrary values representing the compensation values of uncertainties are put into the GNN whose parameters are defined with initial parameters beforehand. And then GNN generates control signals and these signals are used to control the actual system. The errors between the actual output joint trajectory x_j and the realizable objective joint trajectory x_j^d are supposed as the compensation values of uncertainties, which can be calculated by

$$U_j = x_j^d - x_j \quad (j = 1, 2). \tag{16}$$

With the realizable objective joint trajectory, its velocity, acceleration and the above generated compensation values of uncertainties, regarded as the teaching pattern, the well-defined GNN is trained to generate optimal control signals. The backpropagation algorithm is used as the learning algorithm.

3.3 Experimental work

In order to verify the effectiveness of the proposed method for the high-performance contour control of IARA, an experiment work has been carried out. The object of the experiment is an actual Performer MK3S produced by Yahata Co., Japan.

According to the controller construction method for IARA in 3.2, the process of the controller construction for the Performer MK3S was firstly carried out.

As (12), the inverse controller of the input nonlinear static part is the kinematics of IARA described by

$$\begin{cases} u_1 = 0.25\sin{(x_1^d)^u} + 0.215\sin{((x_1^d)^u + (x_2^d)^u)} \\ u_2 = 0.25\cos{(x_1^d)^u} + 0.215\cos{((x_1^d)^u + (x_2^d)^u)}. \end{cases} \tag{17}$$

Similarly, according to (13), the inverse controller of the output nonlinear static part is the inverse kinematics of IARA described by

$$\begin{cases} x_1^d = \arctan\frac{y_1^d}{y_2^d} + \arccos\left[\frac{0.0163+(y_1^d)^2+(y_2^d)^2}{0.5\sqrt{(y_1^d)^2+(y_2^d)^2}}\right] \\ x_2^d = \pi - \arccos\left[\frac{0.109-(y_1^d)^2-(y_2^d)^2}{0.1075}\right]. \end{cases} \tag{18}$$

In order to build an appropriate gaussian neurocontroller for IARA, first of all, the definition of the gaussian neurocontroller structure was made. The gaussian neurocontroller with initial parameters determined by the known Euler-Lagrange equations for the first link of IARA is given by (14), where $C_1 = 7.67 \times 10^{-4}$, $w_1 = 5.81 \times 10^{-4}\ddot{X}_{1max}$, $w_2 = 0.03\dot{X}_{1max}$, $w_3 = 0.757X_{1max}$ and $w_4 = 0.757U_{1max}$. Similarly, the gaussian neurocontroller for the second link of IARA is given by (15), where $C_2 = 2.84 \times 10^{-4}$, $w_1 = 2.15 \times 10^{-4}\ddot{X}_{2max}$, $w_2 = 0.03\dot{X}_{2max}$, $w_3 = 0.757X_{2max}$ and $w_4 = 0.757U_{2max}$.

For obtaining proper parameters of the GNN in order to construct the final neurocontroller of IARA, a training trajectory in the cartesian coordinate was defined by (see Fig.3(a))

$$\begin{cases} y_1^d = R\cos(0.2\pi t) + R\cos(0.2\pi t)/5 \\ y_2^d = R\sin(0.2\pi t) + R\sin(0.2\pi t)/5 \\ 0 \le t \le 10[s]. \end{cases} \tag{19}$$

The radius R of the training trajectory is 2[cm]. In addition, the teaching patterns were generated by the following method. With (18), the training trajectory was transformed from the cartesian coordinate into the joint coordinate. After sampled by the time interval Δt, the training trajectory $x_i^d(i = 1, 2)$ and its relative velocity $\dot{x}_i^d$, acceleration $\ddot{x}_i^d$ in the joint coordinate calculated by the Euler equations as $\dot{x}_i^d(t) = \{x_i^d(t) - x_i^d(t-1)\}/\Delta t$ and $\ddot{x}_i^d(t) = \{\dot{x}_i^d(t) - \dot{x}_i^d(t-1)\}/\Delta t$,

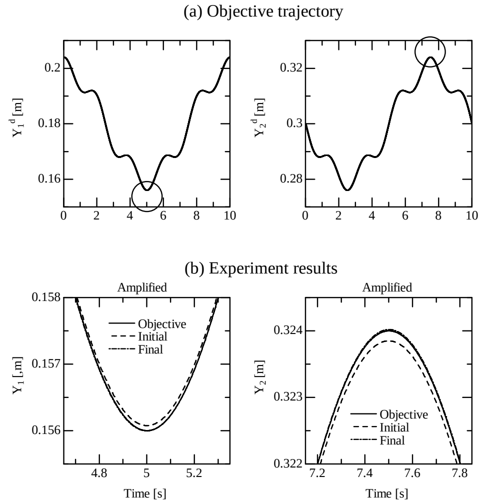

Figure 3: Experiment results of training trajectory

and small arbitrary values representing the compensation values of the uncertainty U_i^d, were put into the current GNN without training. Next, the output of the GNN $(x_i^d)^u$ was transformed by (17) from the joint coordinate into the cartesian coordinate. Then, the output of the entire neuro-inverse controller of IARA, i.e., the control signal u_i, was employed to undertake the contour control of the Performer MK3S. Although the actual output trajectory of the Performer MK3S was not satisfied due to the influence from the uncertainties comparing with the objective trajectory, the compensation values of uncertainties can be obtained by (16) through the experiment. Eventually, the teaching patterns, including the training trajectory $x_i^d(i = 1, 2)$ and its relative velocity $\dot{x}_i^d$, acceleration $\ddot{x}_i^d$ in the joint coordinate as well as the compensation values of uncertainties U_i^d, were generated.

With these teaching patterns, the training process was carried out. The training rate η was selected as a small value of 0.001 because the GNN is already closed to the inverse nonlinear dynamic part of IARA due to its rigid definition of the structure and its initial parameters based on the known Euler-Lagrange equations of IARA given in (11). With the teaching patterns, the parameters of GNN can be updated until convergence. After the training process, the parameters of GNN were fixed for the gaussian neurocontroller.

Fig.3(b) illustrates the experimental results of the training trajectory under enlarged scale in the cartesian coordinate. The solid line represents the objective trajectory. The dashed line, named by “Initial”, represents the actual trajectory of IARA controlled by the neuro-inverse controller whose gaussian neurocontroller was not trained. The dash dot line, named by “Final”, represents the actual trajectory of IARA controlled by the well-trained neuro-inverse controller.

From these results, it proves that the actual trajectory controlled by the well-trained neuro-inverse controller is more accurate than the actual trajectory controlled by the neuro-inverse con-

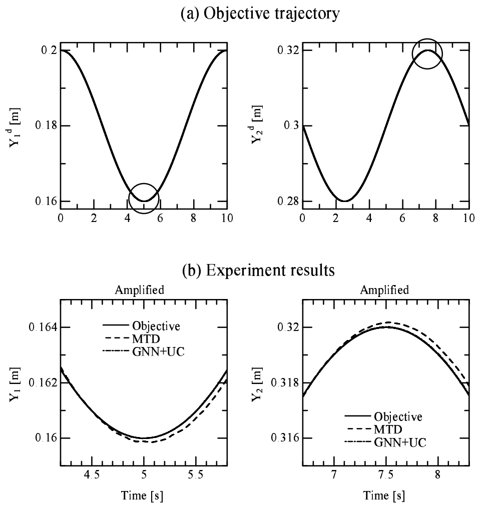

Figure 4: Experiment results of test trajectory comparing with modified taught data method

troller without training. The system performance has been improved using the proposed control method.

In order to evaluate the effectiveness of the proposed control method and also illustrate the adaptive feature of the gaussian neurocontroller, a circle trajectory which is different from the training trajectory (see Fig.4(a)) was defined as an evaluation trajectory, which is described by

$$\begin{cases} y_1^d = R\cos(0.2\pi t) \\ y_2^d = R\sin(0.2\pi t) \\ 0 \le t \le 10. \end{cases} \tag{20}$$

The experimental setup was exactly the same as the experiment for the training trajectory. Fig.4(b) illustrates the experimental results for the evaluation trajectory. In addition, the comparison with the modified taught data method, which was presently used in IARA, was made. In the modified taught data method, the linear dynamic model was used to control the nonlinear dynamic part of the nonlinear separation structure of IARA without considering uncertainties [2].

In Fig.4(b), the solid line represents the objective evaluation trajectory. The dashed line represents the actual trajectory of IARA controlled by the modified taught data method, named by "MTD". The dash dot line represents the actual trajectory controlled by the proposed control method, named by "GNN+UC". From the comparison, it is quite distinct that the actual trajectory controlled by the proposed method is more precise than that controlled by the modified taught data method.

4 Conclusion

A general hybrid neuro-inverse control approach with the knowledge-based nonlinear separation for industrial nonlinear systems was proposed and it was successfully applied for the high-precision contour control of industrial articulated robot arm. This method provides a simple approach to design a high-performance controller for the industrial nonlinear system. The original system is separated into the input-output nonlinear static part and the nonlinear dynamic part based on the physical model or actual field data. An inverse controller is designed for the control of the input-output nonlinear static part and a neurocontroller is used for the control of the nonlinear dynamic part contaminated by uncertainties. With this method, high system performances of different systems can be achieved even with different situation of prior knowledge. The compensation of uncertainties becomes much easy and accurate by the neurocontroller, because the neurocontroller is only responsible for the control of the nonlinear dynamic part instead of the entire nonlinear system. In addition, a powerful neurocontroller with a simple structure and lesser time in training can be implemented. As a general method, the proposed method has a great potential for realizing the high-performance control for various kinds of industrial nonlinear systems.

References

[1] K. Funahashi, "On the approximate realization of continuous mapping by neural networks", *Neural Network*, vol. 2, no. 3, pp. 359-366, 1989.

[2] S. Goto, M. Nakamura and N. Kyura, "Modified taught data method for industrial mechatronic servo controller to achieve accurate contour control performance", *IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics (AIM)*, 525B, 1997.

[3] C. L. Hwang and C. H. Lin, "A discrete-time multivariable neuro-adaptive control for nonlinear unknown dynamic systems", *IEEE Trans. on System, Man and Cybernetics, Part B*, vol. 30, no. 6, pp. 865-877, 2000.

[4] T. A. Johansen, "Identification of non-linear systems using empirical data and prior knowledge - an optimization approach", *Automatica*, vol. 32, no. 3, pp. 337-356, 1996.

[5] S. Lee and R. M. Kil, "A gaussian potential function network with hierarchically self-organizing learning", *Neural Networks*, vol. 4, pp. 207-224, 1991.

[6] M. Nakamura, S. Goto and T. Sugi,"A methodology for designing controllers for industrial systems based on nonlinear separation model and control", *Control Engineering Practice*, vol. 7, pp. 347-356, 1999.

[7] B. Widrow and M. M. Lamego, "Neurointerface", *IEEE Trans. on Control Systems Technology*, vol. 10, no. 2, pp. 221-228, 2002.

[8] Y. Zhang, P. Sen and G. E. Hearn, "An on-line trained adaptive neural controller", *IEEE Control Systems Magazine*, vol. 15, no. 5, pp. 67-75, 1995.

[9] T. Zhang, M. Nakamura, S. Goto and N. Kyura, "Neural-network controller for articulated robot arm with interference for high-precision contour control", *Proceedings of the 6th International Symposium on Artificial Life and Robotics* (AROB'01), vol. 2, pp. 474-477, 2001.