Available at www.MATHEMATICSWEB.ORG
POWERED BY SCIENCE @ DIRECT•

ELSEVIER

Fuzzy Sets and Systems 134 (2003) 117–133

FUZZY sets and systems

www.elsevier.com/locate/fss

Neuro-fuzzy adaptive control based on dynamic inversion for robotic manipulators

Fuchun Sun[a,b,*], Zengqi Sun[a], Lei Li[c], Han-Xiong Li[d]

[a] *Department of Computer Science and Technology, State Key Lab of Intelligent Technology and Systems, Beijing 100084, People's Republic of China*

[b] *Robotics Laboratory, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110015, People's Republic of China*

[c] *School of Public Policy and Management of Tsinghua University, Institute of Software, Chinese Academy of Science, Beijing 100084, People's Republic of China*

[d] *Department of Manufacturing Engineering and Engineering Management, City University of Hong Kong, Hong Kong, People's Republic of China*

Abstract

This paper presents a stable neuro-fuzzy (NF) adaptive control approach for the trajectory tracking of the robotic manipulator with poorly known dynamics. Firstly, the fuzzy dynamic model of the manipulator is established using the Takagi–Sugeno (T–S) fuzzy framework with both structure and parameters identified through input/output data from the robot control process. Secondly, based on the derived fuzzy dynamics of the robotic manipulator, the dynamic NF adaptive controller is developed to improve the system performance by adaptively modifying the fuzzy model parameters on-line. The dynamic NF system aims to approximate the whole robot dynamics rather than its nonlinear components as is done by static neural networks. The dynamic inversion introduced for the controller design is constructed by the dynamic NF system and will help the NF controller design because it does not require the assumption that the robot states should be within a compact set. It is generally known that the compact set cannot be specified before the control loop is closed. Thirdly, the system stability and the convergence of tracking errors are guaranteed by Lyapunov stability theory, and the learning algorithm for the dynamic NF system is obtained thereby. Finally, simulation studies are carried out to show the viability and effectiveness of the proposed control approach.

Keywords: Robotics; Neuro-fuzzy systems; Dynamic inversion; Fuzzy clustering; Adaptive control

* Corresponding author. Tel.: +86-10-6278-8939; fax: +86-10-6277-1138.
E-mail address: sfc@s1000e.cs.tsinghua.edu.cn (F. Sun).

PII: S0165-0114(02)00233-6

1. Introduction

Research efforts for synthesizing neural networks (NNs) with fuzzy logic have been very intensive in the last several years and have generated three major integrated systems: neuro-fuzzy (NF) systems, fuzzy NNs and fuzzy neural hybrid systems [10]. These integrated systems possess advantages of both NNs and fuzzy systems, where NNs provide an essentially low-level learning and computational power to process large amounts of data while fuzzy logic provides a structural framework that utilizes and exploits those low-level capabilities of NNs. This property makes the integrated systems to be more powerful than the pure fuzzy systems or NNs in the modeling and control of nonlinear dynamical systems.

Dynamic NF systems, the NN realization of dynamic fuzzy systems, have the same structure as recurrent NNs. Their inherent memories with feedback connections in dynamics make them suitable for dynamic system modeling and control. Like dynamic NNs, dynamic NF systems not only can simulate some dynamic behaviors such as limit cycles and chaos, but also may provide fast convergence speed with small network size as compared with recurrent NNs [7]. Unlike static NF systems, dynamic NF systems are used to approximate the whole system dynamics, instead of only their nonlinear components. There are two types of dynamic NF systems used recently. One is constructed using dynamic systems in the form of Takagi–Sugeno (T–S) fuzzy model [23], and the other using static NF system with dynamic element [9].

Since dynamic NF systems are essentially fuzzy systems with the kind of automatic tuning methods from NNs, stable dynamic NN-based adaptive control approaches are suitable for all the control frameworks using dynamic NF systems. Recently, some important works in this field have been presented, representative examples are direct and indirect stable adaptive control schemes based on a recurrent NN model of the unknown system by Rovithakis and Christodoulou [13, 12], and Hopfield-type dynamic NN-based identification and control of nonlinear systems [11]. However, there is few works in the stable adaptive control based on dynamic NF systems so far. Dynamic NF systems constructed with the dynamic T–S fuzzy model are most widely used in robust and optimal controller design [3, 2], while researches on stable adaptive control of nonlinear systems almost focus on using static NF networks in the form of T–S fuzzy model. These works include stable adaptive fuzzy control for feedback linearizable dynamical systems [5, 16] and NF adaptive controller design for robotic manipulators based on sliding mode [17]. Furthermore, the dynamic NF system constructed by static NF system with dynamic element, which has been used in the identification and control of flexible-link manipulators by Lee [9], has a similar topological structure to recurrent NNs by Rovithakis [13, 12]. However, in either the dynamic NF systems [9] or the dynamic NNs used in [13, 12], inputs to nonlinear components in their dynamic NF systems are states of the original plant instead of states of dynamic NF system itself. Though these structures are good for mathematical tractability in the derivation of the control law and learning algorithm, and also helpful for the stability of the system in the initial learning of the NF system, the dynamic NF system they proposed is not a true dynamic system. Another problem concerned with the NF adaptive control is that most adaptive control approaches based on NF systems require the system states to be within a compact set for the well-defined approximation. However, the approximation equation may not be true during on-line learning because states of system could be outside the approximation region before the stability of the whole system is achieved.

This paper is to develop the stable adaptive control based on dynamic NF systems for the trajectory tracking of the manipulator with poorly known dynamics. Gustafson–Kessel fuzzy clustering algorithm and least-squares optimization method are used to obtain the dynamic T–S fuzzy model of robot dynamics. By using fuzzy composition operation, the dynamic T–S fuzzy model of the robotic manipulator can be represented in some forms of the dynamic NN. The input to the nonlinear function components in the dynamic NF system is its own state, which makes our dynamic NF system be a true dynamic one. Furthermore, the dynamic inversion constructed by the dynamic NF system is used as the control input to the practical plant, which makes the dynamic NF system determined through design in advance and excludes the assumption that the input to the dynamic NF system should be within a compact set. Finally, the proposed dynamic NF-based adaptive controller is illustrated with simulations of a two-link manipulator.

The rest of the paper is organized as follows. In Section 2, some fundamentals of the robot model and controller design are reviewed. In Section 3, the adaptive controller design using dynamic NF systems is introduced together with a complete control structure and the learning algorithms for parameter adaptation. The proof of the stability and tracking error convergence is also presented in this section. Section 4 shows a simulated example. Finally, the features of the proposed dynamic NF adaptive controller are concluded.

2. Problem statement

2.1. Notation

Standard notation is used in this paper. Let R be the real number set, R^+ be the positive real number set, R^n be the n-dimensional vector space, and $R^{n\times n}$ be a real matrix space. In particular, the norm of a vector $\boldsymbol{x}=(x_1,\dots,x_n)\in R^n$ and that of a matrix $A=(a_{i,j})\in R^{n\times n}$ are defined, respectively, as

$$\|x\| = \sqrt{x^{\mathrm{T}}x}, \quad \|A\| = \sqrt{tr(A^{\mathrm{T}}A)} \tag{1}$$

with $tr(\cdot)$ the trace of a matrix. Moreover, for any positive definite symmetric matrix $A(x)$ and for any x, we denote the minimum and maximum eigenvalues of $A(x)$ by A_m and A_M, respectively. Let $f(t)=(f_1,\dots,f_n)^{\mathrm{T}}$ be a vector function of time, define

$$\|f(t)\|_\infty = \operatorname{ess}\sup_{t\in R} |f(t)|, \tag{2}$$

where $|\cdot|$ denotes the norm in R^n. We say $f(t)\in L_\infty$ if $\operatorname{ess}\sup_{t\geqslant 0}|f(t)|<0$. Finally, we recall from [21] the following definition.

Definition 1 (Vidyasagar [21]). Consider the nonlinear system, $\dot{x}=f(x,u)$, $y=h(x)$ where x is a state vector, u is the input vector and y is the output vector. The solution is uniformly ultimately bounded (UUB) if for all $x(t_0)=x_0$, there exists $\varepsilon>0$ and $T(\varepsilon,x_0)$ such that $\|x(t)\|<\varepsilon$ for all $t\geqslant t_0+T$.

2.2. Robot dynamic equation and properties

The general equation describing the dynamics of an n-degree of freedom rigid robot is given by

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + F(q,\dot{q}) = u(t), \tag{3}$$

where $q, \dot{q} \in R^n$ are the vectors of generalized coordinates and velocities, $M(q) \in R^{n\times n}$ the positive inertia matrix, $C(q,\dot{q})\dot{q} \in R^n$ the Coriolis and centrifugal torques, $G(q) \in R^n$ the gravitational torque, $u(t) \in R^n$ the applied torque. $F(q,\dot{q}) \in R^n$ is the unstructured uncertainty of the dynamics including friction and other disturbances. The following properties of the robot dynamics are required for the subsequent development.

Property 1. $M(q)$ is a positive symmetric matrix defined by $M_m \leqslant \|M(q)\| \leqslant M_M$ with M_m, $M_M > 0$ being known constants.

Property 2. $C(q,\dot{q})$ defined by using the Christoffel symbols, satisfies that

- $\dot{M}(q) - 2C(q,\dot{q})$ is skew symmetric;
- $\|C(q,\dot{q})\| \leqslant C_M \|\dot{q}\|$ and $C(q,y)x = C(q,x)y$, $x, y \in R^n$, $C_M > 0$.

Eq. (3) can be represented in the form of state equation

$$\dot{x} = \bar{A}x + f(x) + B(x)u(x), \tag{4}$$

where $x = (q^{\mathrm{T}}, \dot{q}^{\mathrm{T}})^{\mathrm{T}}$ denotes the system state vector, and

$$\bar{A} = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix}, \quad f(x) = \begin{bmatrix} 0 \\ -M^{-1}(q)(C(q,\dot{q})\dot{q} + G(q) + F(q,\dot{q})) \end{bmatrix}, \quad B(x) = \begin{bmatrix} 0 \\ M^{-1}(q) \end{bmatrix}. \tag{5}$$

2.3. Dynamic NF systems

The robot dynamic equation can be represented by the following dynamic T–S fuzzy model which is described by fuzzy IF–THEN rules.

Rule i:

If $z_1(t)$ is F_{i1}, $z_2(t)$ is $F_{i2}, \ldots, z_p(t)$ is F_{ip}

then

$$\dot{x} = \bar{A}x + \bar{f}_i(x) + \bar{B}_i(x)u(t), \quad i = 1, \ldots, m, \tag{6}$$

where F_{ij} $(i = 1, 2, \ldots, m; j = 1, 2, \ldots, p)$ are fuzzy sets described by membership function $A_{ij}(z_j(t))$, $x(t) \in R^{2n}$ the state vector, $u(t) \in R^n$ the input vector. Besides, it is assumed that $\bar{B}_i(x) = \bar{B}_i$, $\bar{f}_i(x) = \bar{A}_i x + \bar{b}_i$, where $\bar{A}_i \in R^{2n\times 2n}$, $\bar{B}_i \in R^{2n\times n}$ and $\bar{b}_i \in R^{2n}$ are constant matrices and vector, respectively. m is the number of fuzzy rules, and $z_1(t) \sim z_p(t)$ are some measurable variables, i.e. the premise variables.

2.4. Fuzzy model identification

Fuzzy modeling or identification aims at finding a set of fuzzy if-then rules with well-defined parameters, which can describe the given input/output (I/O) behavior of the process. For dynamic T–S fuzzy model (6), Gustafson–Kessel fuzzy clustering algorithm described in [1] is used to determine fuzzy space partition, initial positions of membership functions, while consequent parameters can be obtained under the given structure on the measured I/O data using the following global least-squares optimization method.

The dynamic T–S fuzzy model (6) can be computed using fuzzy composition operation as below

$$\dot{x}(t) = \left(\sum_{i=1}^{m} \theta_i(z(t))\check{A}_i\right) x(t) + \left(\sum_{i=1}^{m} \theta_i(z(t))\bar{B}_i\right) u(t) + \sum_{i=1}^{m} \theta_i(z(t))\bar{b}_i, \tag{7}$$

where $\check{A}_i = \bar{A} + \bar{A}_i$ $(i = 1,\dots,n)$. Obviously, (7) is in some forms of dynamic NN, where the input to nonlinear components in the dynamic NF system is its own state such that a real dynamic NF system is constructed. Furthermore, (7) can also be written as

$$\begin{aligned}\dot{x}(t) &= (\check{A}_1,\dots,\check{A}_m,\bar{B}_1,\dots,\bar{B}_m,\bar{b}_1,\dots,\bar{b}_m)(\theta(z(t))\otimes x^{\mathrm{T}}(t),\theta(z(t))\otimes u^{\mathrm{T}}(t),\theta(z(t)))^{\mathrm{T}}\\ &= \hat{\alpha}v(x(t),u(t),\theta(z(t))),\end{aligned} \tag{8}$$

where "$\otimes$" denotes Kronecker product, and

$$\begin{aligned}&\hat{\alpha} = (\check{A}_1,\dots,\check{A}_m,\bar{B}_1,\dots,\bar{B}_m,\bar{b}_1,\dots,\bar{b}_m) \in R^{2n\times n_a}, \quad n_a = (3n+1)m,\\ &v(x(t),u(t),\theta(z(t))) = (\theta(z(t))\otimes x^{\mathrm{T}}(t),\theta(z(t))\otimes u^{\mathrm{T}}(t),\theta(z(t)))^{\mathrm{T}} \in R^{n_a},\\ &\mu_i(z(t)) = \prod_{j=1}^{p} A_{ij}(z_j(t)), \quad \theta_i(z(t)) = \mu_i(z(t)) \Big/ \sum_{j=1}^{m} \mu_j(z(t)),\\ &z(t) = (z_1(t),z_2(t),\dots,z_p(t)), \quad \theta(z(t)) = (\theta_1(z(t)),\dots,\theta_m(z(t))).\end{aligned} \tag{9}$$

By using least-squares optimization method, parameter vector $\hat{\alpha}$ can be solved as follows:

$$\hat{\alpha}^{\mathrm{T}} = (\Phi^{\mathrm{T}}(\theta,x,u)\Phi(\theta,x,u))^{-1}\Phi^{\mathrm{T}}(\theta,x,u)X$$

with $\Phi(x,u,\theta) = (v(x(1),u(1),\theta(z(1))),\dots,v(x(N),u(N),\theta(z(N))))^{\mathrm{T}}$, and

$$X = (\dot{x}(1),\dot{x}(2),\dots,\dot{x}(N))^{\mathrm{T}}. \tag{10}$$

After the consequent parameters of (6) are determined by global least-squares optimization method, premise parameters can be easily obtained using μ_{it} given in Section 4.1 of [1]. For bell-typed membership functions [8]

$$A_{ij}(z_j(t)) = EXP\{-(z_j - c_{ij})^2/\delta_{ij}^2\}, \quad i=1,\dots,m; \ j=1,\dots,p \tag{11}$$

and the center and width of the jth membership function of the ith fuzzy rule can be determined as

$$c_{ij} = \sum_{t=1}^{N} \mu_{it}z_j(t) \Big/ \sum_{t=1}^{N} \mu_{it}, \quad \delta_{ij} = \kappa\sqrt{\sum_{t=1}^{N} \mu_{it}(z_j(t)-c_{ij})^2 \Big/ \sum_{t=1}^{N} \mu_{it}}, \tag{12}$$

where κ denotes a scale factor, which can be selected according to the fitting precision of least-squares optimization method.

3. Dynamic NF adaptive controller design for a robot

In this section, a design approach of NF adaptive controller based on dynamic inversion will be developed for robot control. The goal is to design a control law $u(t)$ that ensures that the robot joint displacement $q(t)$ follows the desired state trajectory $q_{\mathrm{d}}(t)$. The adaptive NF controller is developed in the following steps. First, the dynamics of tracking error metric is derived for a robot manipulator that is described in the form of dynamic T–S fuzzy model. Then a dynamic NF system is constructed to approximate the robot dynamics, and the dynamic inversion control is obtained thereby. Finally, a composite controller composed of the dynamic inversion control and a PD type control is proposed, and its learning algorithm and global stability proof of the closed-loop control system are also given.

3.1. The dynamics of tracking error metric for a robot modeled by T–S fuzzy model

The following tracking error metric is defined for the NF adaptive controller design:

$$S = C(x - x_{\mathrm{d}}), \tag{13}$$

where $S=(s_1,\dots,s_n)^{\mathrm{T}}, x(t)=(q^{\mathrm{T}},\dot{q}^{\mathrm{T}})^{\mathrm{T}}, x_{\mathrm{d}}=(q_{\mathrm{d}}^{\mathrm{T}},\dot{q}_{\mathrm{d}}^{\mathrm{T}})^{\mathrm{T}}$ is the desired state trajectory to be tracked, and $C=[r,I]\in R^{n\times 2n}$, $r=r^{\mathrm{T}}>0$. Usually, r is chosen as $r=diag(\lambda_1,\dots,\lambda_n)>0$. If the only source of high-frequency unmodeled dynamics is assumed to be the finite sampling, it is shown by Slotine [15] that λ_i $(i=1,\dots,n)$ can be determined by $\lambda_i \leqslant 0.25/\delta$, where δ is the sampling period.

Substituting the state equation presented by the ith fuzzy rule in (6) into the derivative of (13), we have

$$\dot{S} = C(\dot{x} - \dot{x}_{\mathrm{d}}) = -rS + h + C\bar{A}_i x + C\bar{b}_i + C\bar{B}_i u, \tag{14}$$

where

$$h = C\bar{A}x + rS - C\dot{x}_{\mathrm{d}}. \tag{15}$$

It is easy to verify that $G_i=C\bar{B}_i\in R^{n\times n}$ is a positive and symmetric matrix (see (5)), so multiplying G^{-1} to both sides of (14) gives

$$G_i^{-1}\dot{S} = -G_i^{-1}rS + G_i^{-1}h + G_i^{-1}C\bar{A}_i x + G_i^{-1}C\bar{b}_i + u. \tag{16}$$

Define $\tilde{S}=\dot{S}+rS, A_i=G_i{}^{-1}C\bar{A}_i\in R^{n\times 2n}$, $b_i=G_i^{-1}C\bar{b}_i\in R^n$, then (16) can be written

$$G_i^{-1}\tilde{S} = G_i^{-1}h + A_i x + b_i + u. \tag{17}$$

Eq. (17) represents the tracking error dynamics in the subspace defined by the ith fuzzy rule. By using center-average defuzzifier, product inference, and singleton fuzzifier, the error dynamics in the whole state space can be obtained using fuzzy composition as follows:

$$\sum_{i=1}^{m} \theta_i(z(t))G_i^{-1}\tilde{S} = \sum_{i=1}^{m} \theta_i(z(t))G_i^{-1}h + \sum_{i=1}^{m} \theta_i(z(t))(A_i x + b_i) + u, \tag{18}$$

where $\theta_i(z(t))$ $(i=1,\dots,m)$ are defined in (9) and $A_{ij}(z_j(t))$ is in the form of bell-typed Gaussian function shown in (11).

Define $G^{-1}=\sum_{i=1}^{m}\theta_i(z(t))G_i^{-1}$, we have the tracking error dynamics of the robotic manipulator as

$$G^{-1}\tilde{S}=\sum_{i=1}^{m}\theta_i(z(t))G_i^{-1}h+\sum_{i=1}^{m}\theta_i(z(t))(A_ix+b_i)+u. \tag{19}$$

3.2. The dynamics of tracking error metric for a dynamic NF system

Usually, the dynamic T–S fuzzy model constructed by fuzzy model identification method given in Section 2.4 is only a rough approximation to the robot dynamics. For accurate trajectory tracking of the robotic manipulator, a stable adaptive control approach using dynamic NF system will be developed, where the NF system can be adapted by learning algorithms for updating its free parameters such that the approximation performance of the NF system can be improved on-line. The following dynamic NF system with the same antecedents as that in (6) is constructed to approximate the robot dynamics shown in (3) as

Rule i:
If $z_1(t)$ is F_{i1}, $z_2(t)$ is $F_{i2},\dots,z_p(t)$ is F_{ip}
then

$$\dot{\hat{x}}=\bar{A}\hat{x}+\hat{\bar{f}}_i(\hat{x})+\hat{\bar{B}}_i(\hat{x})(u-u_p), \tag{20}$$

where F_{ij} $(i=1,2,\dots,m;j=1,2,\dots,p)$ are fuzzy sets described by membership function $A_{ij}(z_j(t))$, $\hat{x}=(\hat{q}^{\mathrm{T}},\dot{\hat{q}}^{\mathrm{T}})^{\mathrm{T}}\in R^{2n}$ is the state vector of the NF system. Besides, assume that $\hat{\bar{B}}_i(\hat{x})=\hat{\bar{B}}_i$, $\hat{\bar{f}}_i(\hat{x})=\hat{\bar{A}}_i\hat{x}+\hat{\bar{b}}_i$, and $\hat{\bar{A}}_i\in R^{2n\times 2n}$ $\hat{\bar{B}}_i\in R^{2n\times n}$ and $\hat{\bar{b}}_i\in R^{2n}$ denote the estimates of $\bar{A}_i$ $\bar{B}_i$ and $\bar{b}_i$. u_p is a robust control component which will be defined later, and as $x\to x_{\mathrm{d}}$, $u_p\to 0$ then the NF system constructed is equivalent to the robot system in terms of state and control input.

The tracking error metric for the dynamic NF system (20) can be defined as

$$S_0=C(\hat{x}-x_{\mathrm{d}}). \tag{21}$$

By the same derivation as that of (19), S_0 dynamics in the whole fuzzy space can be written as

$$G_F^{-1}\tilde{S}_0=\sum_{i=1}^{m}\theta_i(z(t))\hat{G}_i^{-1}\hat{h}+\sum_{i=1}^{m}\theta_i(z(t))(\hat{A}_i\hat{x}+\hat{b}_i)+u-u_p, \tag{22}$$

where $G_F^{-1}=\sum_{i=1}^{m}\theta_i(z(t))\hat{G}_i^{-1}$ with $\hat{G}_i=C\hat{\bar{B}}_i$, $\tilde{S}_0=\dot{S}_0+r_0S_0$, $\hat{A}_i=\hat{G}_i^{-1}C\hat{\bar{A}}_i$, $\hat{b}_i=\hat{G}_i^{-1}C\hat{\bar{b}}_i$, $r_0>0$ is a design parameter used to assign the dynamics of S_0, and $\hat{h}=C\bar{A}\hat{x}+r_0S_0-C\dot{x}_{\mathrm{d}}$.

Remark 1. The universal approximation power of the dynamic NF system in the form of (6) has been proved in [22]. Furthermore, Johansen and Shorten [6, 14] also consider its interpretation and identification. Evidently, for approximating the dynamic NF fuzzy model (6) by using dynamic NF system (20), the sufficient and necessary condition will be $\hat{\bar{A}}_i\to\bar{A}_i$, $\hat{\bar{B}}_i\to\bar{B}_i$, $\hat{\bar{b}}_i\to\bar{b}_i$ $(i=1,\dots,m)$ if the membership functions are chosen appropriately. For these, $\hat{A}_i$, $\hat{G}_i$ and $\hat{b}_i$ $(i=1,\dots,m)$ are termed as the estimates of parameter matrices A_i, G_i and vector b_i.

3.3. NF dynamic inversion

Dynamic inversion used in the controller design is defined as the inverse model of the dynamic NF system shown in (20) with state specified by desired dynamics, which is similar to the stable inversion proposed by Chen and Zhao [4]. Stable inversion is the inverse model of the multivariable system with desired state trajectory as its model input, which guarantees that the inverse model is always bounded. For a dynamic NF system (20), if the desired dynamics is chosen as

$$\dot{S}_0 = -r_0 S_0 \tag{23}$$

then the dynamic inversion of the dynamic NF system is obtained from (22) as

$$u_I(t) = u_s(t) - u_p(t) = -\sum_{i=1}^{m} \theta_i(z(t))\hat{G}_i^{-1}\hat{h} - \sum_{i=1}^{m} \theta_i(z(t))(\hat{A}_i\hat{x} + \hat{b}_i). \tag{24}$$

Since $\hat{x}$ will converge to the desired trajectory specified by dynamics (23), the only difference between dynamic inversion and stable inversion lies in the initial stage of the control process. The advantage from dynamic inversion is that the initial state trajectory $\hat{x}$ can be designed in advance such that good dynamic performance of the closed-loop system is guaranteed. Besides, dynamics relation (23) ensures the state of the dynamic NF system to be in the compact set strictly such that the dynamic inversion always exists. It will be proven that the dynamic NF system (20) will approximate the robot dynamics (6), with appropriate control law included dynamic inversion and the dynamic NF learning algorithm, and thus the robot state will track the desired trajectory $x_{\rm d}$. As a result, the dynamic inversion (24) will finally approximate the inverse dynamics of the robot.

3.4. Dynamic NF controller design

The following control law is considered for the robot trajectory tracking:

$$u = u_I - \bar{K}\tilde{x}, \tag{25}$$

where u_I is defined in (24), $\bar{K} = KC \in R^{n\times 2n}$, and $K \in R^{n\times n}$ is assumed to be diagonal.

The whole NF adaptive control diagram is shown in Fig. 1, where the dynamic NF system which acts as a feedforward controller u_I, is used to compensate for the robot inverse dynamics defined in (3). In the feedback control loop, there is a proportional-derivative (PD) control u_p, which is used to enhance the stability and robustness of the robot control system. Unlike the adaptive control structure using static NNs [19] with only one loop, the control structure using dynamic NF system shown in Fig. 1 has two loops, one by the robot control loop, the other by dynamic NF system.

Define the state deflection metric of the robot from the dynamic NF system as

$$S_e = C\tilde{x} \tag{26}$$

with $\tilde{x} = x - \hat{x} = (q_e^{\rm T}, \dot{q}_e^{\rm T})^{\rm T}$, and C defined as before.

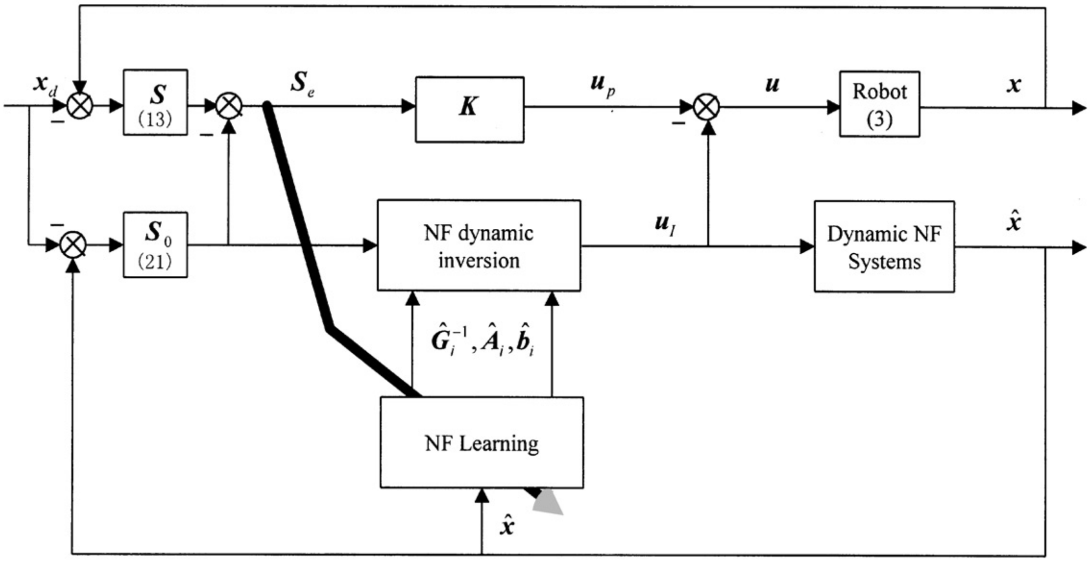


Fig. 1. NF adaptive control structure.

Subtracting (22) from (19) and using (23), we have

$$
\begin{aligned}
G^{-1}\tilde{S}_e &= \sum_{i=1}^{m} \theta_i(z(t))(G_i^{-1} - \hat{G}_i^{-1})\hat{h} + \sum_{i=1}^{m} \theta_i(z(t))((A_i - \hat{A}_i)\hat{x} + b_i - \hat{b}_i) \\
&\quad + \sum_{i=1}^{m} \theta_i(z(t))(V_i - \bar{K})\tilde{x} \\
&= \tilde{W} Y(\theta(z(t)), \hat{x}, \hat{h}) + \sum_{i=1}^{m} \theta_i(z(t))(V_i - \bar{K})\tilde{x}, \qquad (27)
\end{aligned}
$$

where

$$
\begin{aligned}
&\tilde{S}_e = \dot{S}_e + rS_e, \quad V_i = G_i^{-1}(C\bar{A} + rC) + A_i, \quad V = \sum_{i=1}^{m} \theta_i(z(t))V_i, \\
&\tilde{W} = W - \hat{W}, \quad W = (G_1^{-1}, \ldots, G_m^{-1}, A_1, \ldots, A_m, b_1, \ldots, b_m) \in R^{n \times n_a}, \\
&\hat{W} = (\hat{G}_1^{-1}, \ldots, \hat{G}_m^{-1}, \hat{A}_1, \ldots, \hat{A}_m, \hat{b}_1, \ldots, \hat{b}_m) \in R^{n \times n_a}, \qquad (28) \\
&Y(\theta(z(t)), \hat{x}, \hat{h}) = (\theta(z(t)) \otimes \hat{x}^{\mathrm{T}}(t), \theta(z(t)) \otimes \hat{h}^{\mathrm{T}}, \theta(z(t)))^{\mathrm{T}} \\
&\qquad = (y_1(k), \ldots, y_{n_a}(k)), \quad n_a = (3n+1)m.
\end{aligned}
$$

The following theorem gives stable adaptive control law and learning algorithm for dynamic NF systems.

Theorem. *Consider the n-link rigid robot manipulator described by the dynamic model given in* (3), *by applying the control law* (25) *with the dynamic NF learning algorithm*

$$\dot{\hat{W}} = \eta(S_e Y^{\mathrm{T}}(\theta(z(t)), \hat{x}, \hat{h}) + \sigma(W_0 - \hat{W})), \tag{29}$$

where $\eta = diag(\eta_1, \ldots, \eta_n) > 0$ *is a learning rate matrix*, $\sigma > 0$ *and* W_0 *are design parameters. If the following relation*:

$$K_m > M_G + \tfrac{3}{2} V_{\max}(1 + r_m^{-1}) \tag{30}$$

holds, where $M_G = \frac{1}{2}\sup_q \|\dot{M}(q)\|$, $V_{\max} = \sup_t \geqslant \|V\|$, K_m and r_m *are the minimum eigenvalues of* K *and* r, *respectively, then the tracking error metric of the robot is uniformly ultimately bounded.*

Proof. See the appendix. □

Remark 2. $M(q)$ only contains trigonometric functions of q, hence the derivative of each element with respect to q is bounded so that boundedness of M_G is guaranteed.

4. Application example

In this section, the above developed control approach is employed in the position control of a two-link manipulator. The dynamic equation and parameters of a two-link manipulator are the same as those given in [19]. The desired joint angle trajectory for the robot to follow is

$$q_{1\mathrm{d}}(t) = 0.5(\sin t + \sin 2t), \quad q_{2d}(t) = 0.5(\cos 3t + \cos 4t). \tag{31}$$

And the initial simulation condition for robot motion is chosen as

$$q_1(0) = 1.0, \quad \dot{q}_1(0) = -0.5, \quad q_2(0) = 0.5, \quad \dot{q}_2(0) = -2.0 \tag{32}$$

where $q_i, q_{i\mathrm{d}}$ $(i = 1, 2)$ are the ith robot joint angle and the corresponding desired trajectory, respectively.

4.1. Derivation of the T–S fuzzy model

The dynamic T–S fuzzy model of a robot can be derived using Gustafson–Kessel clustering approach through I/O data from the robot control process with sampling interval 0.02 s. There are 400 sets of data in all used for fuzzy clustering. It has been shown in the example that the dynamic NF system with 8 rules is enough to approximate the robot dynamic process. The design parameters are chosen as $w = 2, \kappa = 9.8$. Partial results are presented as follows:

Rule i:

If x is about cen_i

then

$$\dot{x} = \bar{A}x + \hat{\bar{A}}_i x + \hat{\bar{B}}_i u + \hat{\bar{b}}_i. \tag{33}$$

The membership function for the ith fuzzy rule can be represented as

$$\mu_i(x) = EXP((x - cen_i)^{\mathrm{T}} M_i^{-1}(x - cen_i)) \tag{34}$$

with

$$Cen = \begin{bmatrix} cen_1 \\ cen_2 \\ \vdots \\ cen_7 \\ cen_8 \end{bmatrix} = \begin{bmatrix} -0.8347 & -0.1304 & -0.6528 & -0.9633 \\ 0.08582 & 0.1938 & -0.1182 & 0.08137 \\ \vdots & \vdots & \vdots & \vdots \\ -0.5365 & 1.114 & 0.1427 & 2.882 \\ 0.3884 & -0.9429 & 0.4343 & 0.5851 \end{bmatrix},$$

$$M_1 = diag(18.17 \;\; 82.87 \;\; 2.466 \;\; 139.8),\; M_2 = diag(2.472 \;\; 78.82 \;\; 1.993 \;\; 214.5),$$
$$M_3 = diag(35.81 \;\; 24.20 \;\; 12.46 \;\; 261.6),\; M_4 = diag(12.62 \;\; 1.785 \;\; 8.314 \;\; 436.3),$$
$$M_5 = diag(7.776 \;\; 18.50 \;\; 1.479 \;\; 161.2),\; M_6 = diag(3.389 \;\; 36.78 \;\; 1.682 \;\; 247.9),$$
$$M_7 = diag(1.775 \;\; 33.65 \;\; 13.64 \;\; 342.0),\; M_8 = diag(0.6163 \;\; 1.243 \;\; 4.572 \;\; 4.713).$$

After having determined cluster centers and widths for 8 fuzzy rules using Gustafson–Kessel clustering approach, parameters $\hat{G}_i^{-1}, \hat{A}_i$, and $\hat{b}_i$ $(i=1,\dots,8)$ in (22) are calculated by global least square optimization method, which are used to assign the initial values for dynamic NF systems. The estimated values of $\hat{G}_i^{-1}, \hat{A}_i, \hat{b}_i$ $(i=1,\dots,8)$ are omitted here for saving the paper length.

4.2. NF adaptive controller design

In the previous fuzzy modeling of dynamic NF system, constant vectors $\bar{b}_i$ $(i=1,\dots,8)$ have not been considered in the identification model for the time being, which will be considered during on-line adaptive learning for compensating constant disturbances. The basis functions of NF system can be chosen by (28), there are 56 neurons in all required, which is <142 neurons used in the static NN-based adaptive controller design [20]. The initial weights for the dynamic NF adaptive controller are chosen as

$$\hat{W}(0) = (\hat{G}_1^{-1}(0),\dots,\hat{G}_8^{-1}(0),\hat{A}_1^{-1}(0),\dots,\hat{A}_8^{-1}(0),\hat{b}_1^{-1}(0),\dots,\hat{b}_8^{-1}(0)), \tag{35}$$

where $\hat{G}_i^{-1}(0)$ and $\hat{A}_i^{-1}(0)$ $(i=1,\dots,8)$ come from the model identification in Section 4.1, $\hat{b}_i(0)$ $(i=1,\dots,8)$ are chosen as zero vectors or as random numbers.

The design parameters are chosen as

$$r = diag(12,12), \quad C = (r, I), \quad K = diag(100, 80), \quad I = diag(1,1) \tag{36}$$

and parameter vector $r_0 = 0.12I$ is chosen to assign the desired dynamics of the dynamic NF system for good tracking responses of the robot state q in the initial control process. The learning law for dynamic NF adaptive control is chosen as

$$\eta_1 = 222.5/(\|Y(\theta(z(t)),\hat{x},\hat{h})\|_\infty + 0.5), \quad \eta_2 = 221.3/(\|Y(\theta(z(t)),\hat{x},\hat{h})\|_\infty + 0.5). \tag{37}$$

Fig. 2(a) and (b) present the angle tracking errors for two joints of the robot during the first 10 s and the last 10 s of operation using dynamic NF controller with payload $m_2 = 6.25$ kg. To evaluate the robustness of the dynamic NF controller against constant disturbances and payload variations,

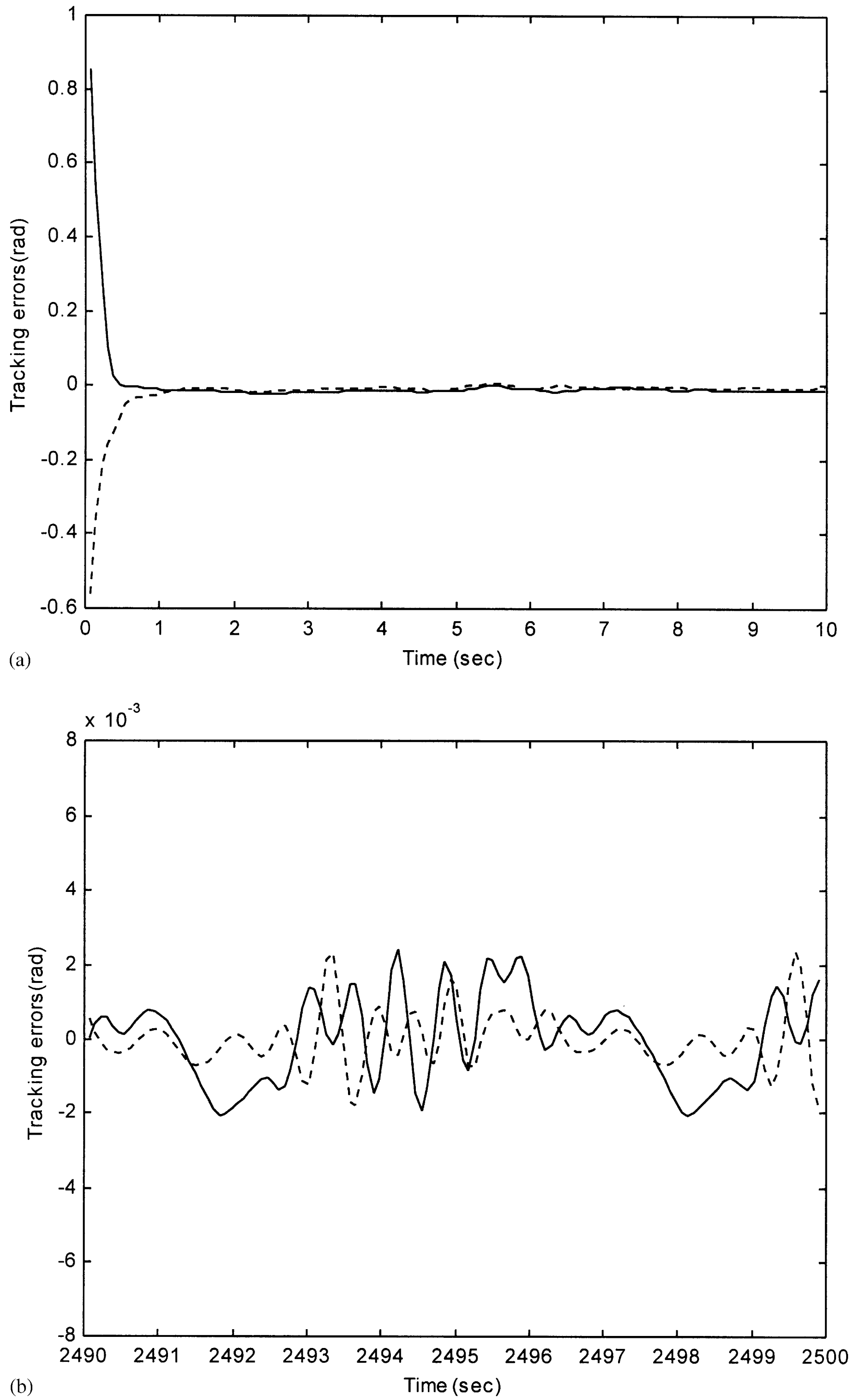
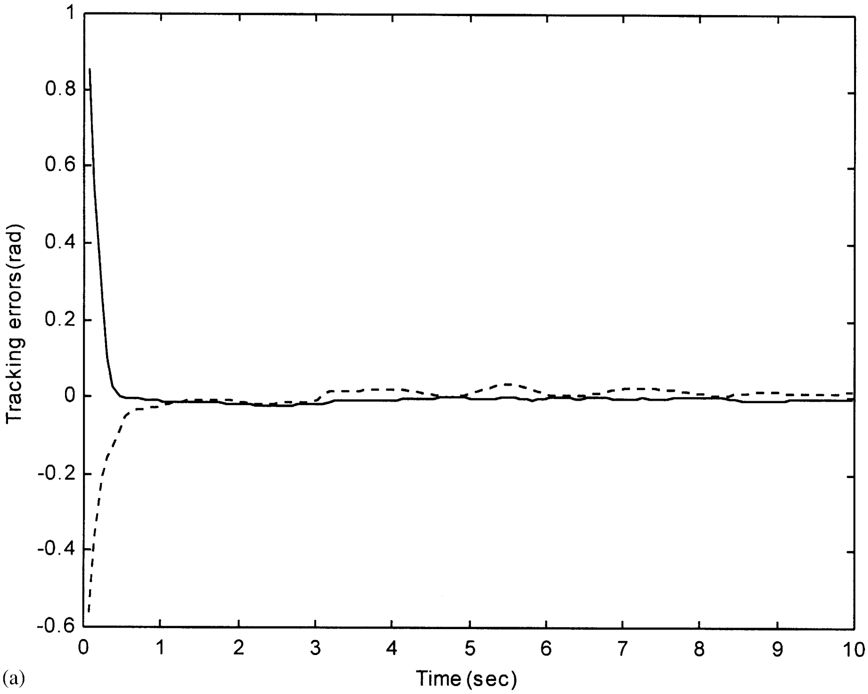
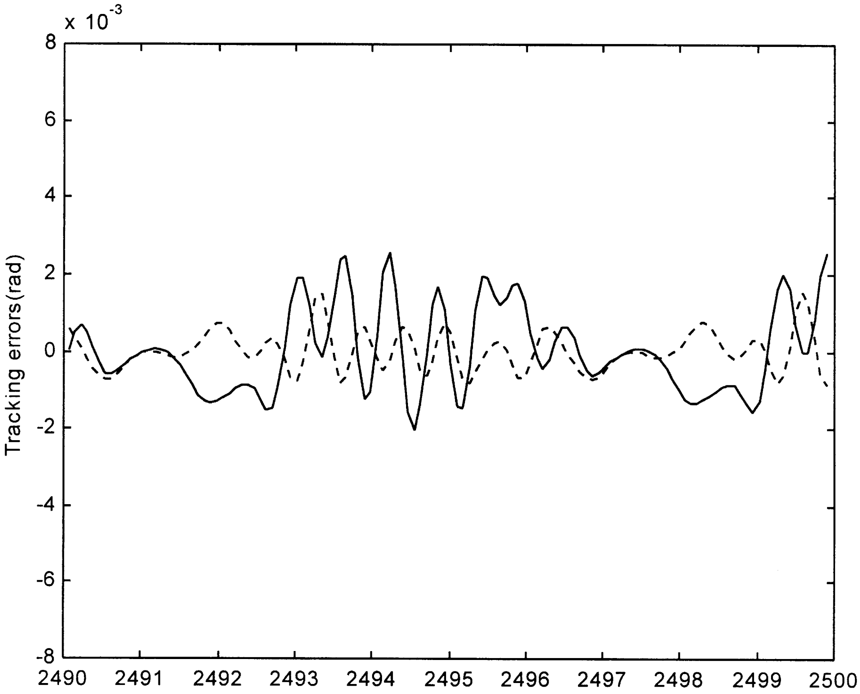


Fig. 2. Robot angle tracking error responses using dynamic NF control with payload $m_2 = 6.25$ kg for q_1 (solid) and q_2 (dotted): (a) during the first 10 s of operation; (b) during last 10 s of operation.



(a)



(b)

Fig. 3. Robot angle tracking error responses using dynamic NF control with disturbances for q_1 (solid) and q_2 (dotted): (a) during the first 10 s of operation; (b) during last 10 s of operation.

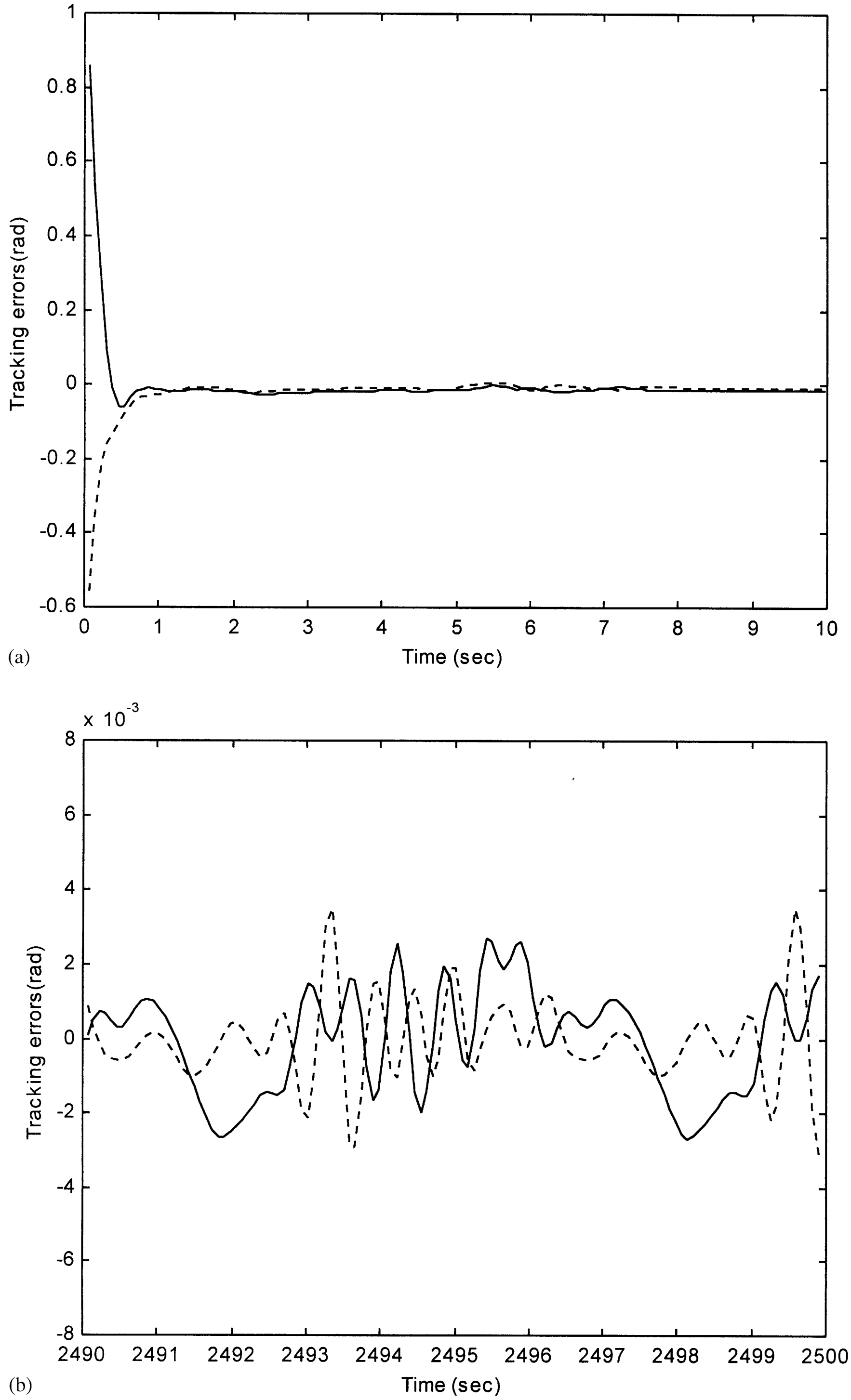


Fig. 4. Robot angle tracking error responses using dynamic NF control with payload $m_2 = 10.25$ kg for q_1 (solid) and q_2 (dotted): (a) during the first 10 s of operation; (b) during last 10 s of operation.

disturbance control torques and payload variation are considered in the robot dynamics for simulation studies, respectively. Fig. 3(a) and (b) show the angle tracking errors for a robot with disturbance torques 100 N m, which are added to two joints of the robot since $t \geqslant 3$ s. Fig. 4(a) and (b) present the angle tracking errors for two joints of a robot, with payload $m_2 = 10.25$ kg instead of $m_2 = 6.25$ kg. It has been shown in Figs. 3 and 4 evidently that dynamic NF controller has a better robustness against constant disturbances and payload variations. Besides, the control results shown in Fig. 2(a) and (b) are superior to these given by the adaptive control approaches of robotic manipulators using static and dynamic NNs [20, 18].

We have shown in previous simulations that dynamic NF adaptive control has a better tracking performance than both static and dynamic NN-based adaptive ones in the whole control process and has robustness against constant disturbances and payload variations. The reason for this is two folds. One is attributed to the modeling capability of the dynamic NF system, where the good NN structure and initial weights are determined for the dynamic NF system while the initial weights in the static and dynamic NN-based controllers are totally unknown. The other is introduction of the PD type control in the control structure of Fig. 1 for enhancing the stability and robustness of the dynamic NF control system.

5. Conclusions

A stable NF adaptive control approach integrating the merits of dynamic NN approach and the modeling power of the fuzzy logic system has been developed for the trajectory tracking of a robotic manipulator with poorly known dynamics. With the modeling power of fuzzy logic, the structure of NF system can be determined using I/O data and linguistic information from the robot control process. In the example, the controller obtained is simpler in structure with only 56 neurons and shows a better approximation capability than the dynamic NNs with 154 neurons [18] and the static NNs with 142 neurons [20]. Besides, by appropriately choosing r_0 for assigning the desired dynamics of the NF system, the dynamic performance of robot trajectory tracking is guaranteed in the initial stage of control process. The control law developed contains the dynamic inversion constructed by the dynamic NF system, a PD control component that is used to compensate for uncertainty and improve the dynamic performance of the robot manipulator. Using Lyapunov stability theory, the complete stability and the tracking error convergence are proven, and the learning algorithm is obtained thereby. Simulations for the trajectory tracking of a two-link manipulator show the superiority of the stable dynamic NF adaptive control approach to those using dynamic and static NNs.

Acknowledgements

This work was jointly supported by the National Science Foundation of China under Grant 60084002, the National Excellent Doctoral Dissertation Foundation of China under Grant 200041, 863 High-Tech Project under Grant 863-704-2-18, the Science Foundation of the College of Information Science and Technology of Tsinghua University, and Robotics Laboratory, Shenyang Institute of Automation, Chinese Academy of Sciences Foundation under Grant RL200001.

Appendix A. The Proof of Theorem

Let the Lyapunov function is defined by

$$V = \tfrac{1}{2} S_e^{\mathrm{T}} G^{-1} S_e + \tfrac{1}{2} S_0^{\mathrm{T}} S_0 + q_e^{\mathrm{T}} K r q_e + \tfrac{1}{2}\, tr(\tilde{W}^{\mathrm{T}} \eta^{-1} \tilde{W}). \tag{A.1}$$

Differentiating V_0 defined in (A.1) with respect to time leads to

$$\dot{V} = \tfrac{1}{2} S_e^{\mathrm{T}} \dot{G}^{-1} S_e - S_e^{\mathrm{T}} G^{-1} r S_e + S_e^{\mathrm{T}} \tilde{W} Y(\theta(z(t)), \hat{x}, \hat{h}) + S_e^{\mathrm{T}} (V - \bar{K}) \tilde{x}$$
$$- S_0^{\mathrm{T}} r_0 S_0 + 2\dot{q}_e^{\mathrm{T}} K r q_e - tr(\dot{\hat{W}}^{\mathrm{T}} \eta^{-1} \tilde{W}). \tag{A.2}$$

Since

$$S_e^{\mathrm{T}} V \tilde{x} \leqslant \tfrac{1}{2} V_{\max}(3 + r_m^{-1}) \|\dot{q}_e\|^2 + \tfrac{1}{2} V_{\max}(1 + 3 r_m^{-1}) \|r q_e\|^2, \tag{A.3}$$

$$- S_e^{\mathrm{T}} \bar{K} \tilde{x} + 2\dot{q}_e^{\mathrm{T}} K r q_e \leqslant -K_m(\|\dot{q}_e\|^2 + \|r q_e\|^2). \tag{A.4}$$

Besides, it is easy to verify from (5) that $G^{-1} = M(q)$. For this, we have

$$\tfrac{1}{2} S_e^{\mathrm{T}} \dot{G}^{-1} S_e \leqslant M_G(\|\dot{q}_e\|^2 + \|r q_e\|^2). \tag{A.5}$$

Substituting (A.3) through (A.5) into (A.2) and using (29) lead to

$$\dot{V} \leqslant -r_m M_m \|S_e\|^2 - r_{0m} \|S_0\|^2 - \alpha \|r q_e\|^2 - \sigma\, tr((W_0 - W)^{\mathrm{T}} \tilde{W}), \tag{A.6}$$

where $\alpha = K_m - \frac{3}{2} V_{\max}(1 + r_m^{-1}) - M_G > 0$ with condition given in (30). Since

$$- \sigma\, tr((W_0 - W)^{\mathrm{T}} \tilde{W}) = -\tfrac{1}{2} \sigma \|\tilde{W}\|^2 - \tfrac{1}{2} \sigma \|W_0 - \hat{W}\|^2 + \tfrac{1}{2} \sigma \|W - W_0\|^2. \tag{A.7}$$

Substituting (A.7) into (A.6), we have

$$\dot{V} < -r_m M_m \|S_e\|^2 - r_{0m} \|S_0\|^2 - \alpha \|r q_e\|^2 - \tfrac{1}{2} \sigma \|\tilde{W}\|^2 - \tfrac{1}{2} \sigma \|W_0 - \hat{W}\|^2 + \tfrac{1}{2} \sigma \|W - W_0\|^2$$
$$\leqslant -\frac{P}{P_{\max}} V + \gamma, \tag{A.8}$$

where $P_{\max} = max(M_M, 1.0, 2K_M r_m^{-1}, \eta_m^{-1})$, $P = min(2r_m M_m, 2r_{0m}, 2\alpha, \sigma) > 0$, $\gamma = \frac{1}{2} \sigma \|W - W_0\|^2$ with M_M, M_m defined in Section 2.2 by property 1, K_M is the maximum eigenvalue of K, and r_{0m}, η_m defined as the minimum eigenvalues of r_0 and η, respectively.

It is concluded from (A.8) that $\|\tilde{x}\|$ and $\|\tilde{W}\|$ will eventually fall into a residual set with the size $O(\gamma)$, and so will $\|\dot{q}_e\|$ and $\|q_e\|$. Moreover, $\hat{q}$ will converge to the desired state vector q_{d} exponentially fast as required (See (23)), so $\|S\|$ will also eventually fall into a residual set with the size $O(\gamma)$. By applying Definition 1 theorem is proved.

References

[1] R. Babuska, H.B. Verbruggen, Constructing fuzzy models by product space clustering, in: H. Hellendoorn, D. Driankov (Eds.), Fuzzy Model Identification—Selected Approaches, Springer, Berlin, 1997, pp. 53–90.

[2] S.G. Cao, N.W. Rees, G. Feng, Quadratic stability analysis and design of continuous-time fuzzy control systems, Internat. J. Systems Sci. 27 (1996) 193–203.
[3] S.G. Cao, N.W. Rees, G. Feng, H_∞ control of uncertain fuzzy continuous-time systems, Fuzzy Sets and Systems 115 (2000) 171–190.
[4] D.G. Chen, B. Paden, Stable inversion of nonlinear nonminimum phase systems, Internat. J. Control 64 (1996) 81–97.
[5] S. Jagannathan, Adaptive fuzzy logic control of feedback linearizable discrete-time dynamical systems under persistence of excitation, Automatica 34 (1998) 1295–1310.
[6] T.A. Johansen, R. Shorten, R. Murray-Smith, On the interpretation and identification of dynamic Takagi–Sugeno fuzzy models, IEEE Trans. Fuzzy Systems 8 (2000) 297–313.
[7] C.F. Juang, C.T. Lin, A recurrent self-organizing neural fuzzy inference network, IEEE Trans. Neural Networks 10 (1999) 828–845.
[8] E. Kim, M. Park, S. Ji, M. Park, A new approach to fuzzy modeling, IEEE Trans. Fuzzy Systems 5 (1997) 328–336.
[9] J.X. Lee, G. Vukovich, The dynamic fuzzy logic system: nonlinear system identification and application to robotic manipulators, J. Robotic Systems 14 (1997) 391–405.
[10] C.T. Lin, C.S.G. Lee, Neural Fuzzy Systems: A Neural-Fuzzy Synergism to Intelligent Systems, Prentice-Hall, Englewood Cliffs, NJ, 1996.
[11] A.S. Poznyak, W. Yu, E.N. Sanchez, J.P. Perez, Nonlinear adaptive trajectory tracking using dynamical neural networks, IEEE Trans. Neural Networks 10 (1999) 1402–1411.
[12] G.A. Rovithakis, Tracking control of multi-input affine nonlinear dynamical systems with unknown nonlinearities using dynamical neural networks, IEEE Trans. Systems Man Cybernet. Part B 29 (1999) 179–189.
[13] G.A. Rovithakis, M.A. Christodoulou, Adaptive control of unknown plants using dynamical neural networks, IEEE Trans. Systems Man Cybernet. 24 (1994) 400–412.
[14] R. Shorten, R. Murray-Smith, R. Bjorgan, H. Gollee, On the interpretation of local models in blended multiple model structure, Internat. J. Control 72 (1999) 620–628.
[15] J.J.E. Slotine, Sliding mode controller design for nonlinear systems, Internat. J. Control 40 (1984) 421–434.
[16] J.T. Spooner, K.M. Passino, Stable adaptive control using fuzzy systems and neural networks, IEEE Trans. Fuzzy Systems 4 (1996) 339–359.
[17] F.C. Sun, Z.Q. Sun, G. Feng, An adaptive fuzzy controller based on sliding mode for robot manipulators, IEEE Trans. Systems Man Cybernet. Part B 29 (1999) 661–667.
[18] F.C. Sun, Z.Q. Sun, N. Li, L.B. Zhang, Robot adaptive control based on dynamic inversion using dynamical neural networks, Mach. Intell. Robot Control 1 (1999) 71–78.
[19] F.C. Sun, Z.Q. Sun, P.Y. Woo, Stable neural network-based adaptive control for sampled-data nonlinear systems, IEEE Trans. Neural Networks 9 (1998) 956–968.
[20] F.C. Sun, Z.Q. Sun, R.J. Zhang, Y.B. Chen, Neural adaptive tracking controller for robot manipulators with unknown dynamics, IEEE Proc. Part D 147 (2000) 366–370.
[21] M. Vidyasagar, Nonlinear Systems Analysis, Prentice-Hall, Englewood Cliffs, NJ, 1993.
[22] H.O. Wang, J. Li, D. Niemann, K. Tanaka, T–S model with linear rule consequence and PDC controller: a universal framework for nonlinear control systems, Proc. IEEE Internat. Conf. on Fuzzy Systems, 2000, pp. 549–554.
[23] H.O. Wang, K. Tanaka, M.F. Griffin, An approach to fuzzy control of nonlinear systems: stability and design issues IEEE Trans. Fuzzy Systems 4 (1996) 14–23.