Dynamic learning from adaptive neural control for flexible joint robot with tracking error constraints using high-gain observer

Zhiguang Chen, Min Wang and Yongtao Zou

School of Automation Science and Engineering, South China University of Technology, Guangzhou, People's Republic of China

ABSTRACT
This paper presents dynamic learning from adaptive neural control with prescribed tracking error performance for flexible joint robot (FJR) included unknown dynamics. Firstly, a system transformation method is introduced to convert the original FJR system into a normal system. As a result, only one neural network (NN) approximator is used to identify the uncertain system nonlinear dynamics and the verification on the convergence of neural weights is simplified extremely. To further solve the predefined performance issue, a performance function is introduced to describe a tracking error constraint and a error transformation technique is used to convert the constrained tracking control problem into the unconstrained stabilization of error system. By combining a high-gain observer and the backstepping method, the adaptive neural controller is presented to stabilize the unconstrained error system. Under the satisfaction of the partial persistent excitation condition, the adaptive neural controller is shown to be capable of achieving unknown dynamics acquisition, expression and storage. Furthermore, a neural learning control with using the stored NN weights is proposed for the same or similar control task so that a time-consuming NN online adjustment process can be avoided and a better control performance can be obtained. Simulation results demonstrate the effectiveness of the proposed control method.

ARTICLE HISTORY
Received 4 April 2018
Accepted 1 November 2018

KEYWORDS
Adaptive neural control; neural learning; tracking error constraints; PE condition; flexible joint robot

1. Introduction
Over the past two decades, the dynamics behaviour analysis, modelling and control of robotic manipulators with joint flexibilities has received a plenty of attention (Benosman & Le, 2004; Jiang, Liu, Chen, & Zhang, 2015; Nicosia & Tomei, 1990; Ozgoli & Taghirad, 2006; Rahimi & Nazemizadeh, 2014; Spong, Hutchinson, & Vidyasagar, 2006). In comparison with rigid robots, the flexible joint robots (FJR) have a number of advantages such as low power consumption, light weight, good security, large working space (Rahimi & Nazemizadeh, 2014). However, due to the presence of elastic gearboxes, it is necessary to consider the motor mass and inertia in the FJR modelling. Therefore, for an \( n \)-link FJR, \( 2n \) generalized coordinates are usually required to describe its whole dynamics behaviours, and thus FJR model is more complex than a rigid robot. Additionally, the model uncertainties, including parametric uncertainties and nonparametric dynamic uncertainties, caused by modelling errors and external disturbances, make the controller design for FJR challenging extremely. Based on the singular perturbation approach, Spong (1989) proposed an adaptive control scheme for FJR. In Al-Ashoor, Patel, and Khorasani (1993), a robust adaptive controller was developed for a reduced-order flexible-joint model using on-line identification of the manipulator parameters. Subsequently, many efficient methods, such as backstepping approach (Krstic, Kanellakopoulos, and Kokotovic, 1995; Oh & Lee, 1999; Soukkou & Labiod, 2015), sliding mode control (Huang & Chen, 2004), neural network (NN) control (Ge, Lee, & Tan, 1998) and adaptive fuzzy control (Li, Tong, & Li, 2013; Tang, Chen, & Lu, 2001), have been widely applied to deal with the control problem for FJR. By combining backstepping design and the function approximation technique, an adaptive sliding controller was proposed in Huang and Chen (2004) for a single-link flexible-joint robot with mismatched uncertainties. The \( H_\infty \) disturbance attenuation design and recurrent NN adaptive control technique were proposed in Miao and Wang (2013) to achieve the desired \( H_\infty \)-infinity tracking performance of uncertain FJR. Although the backstepping design has extensive applications in FJR control fields, it is limited in its inherent computational complexity by the repeated differentiations of virtual controllers. To address this problem, Swaroop, Hedrick, Yip, and Gerdes (2000) presented a dynamic surface control technique by introducing a first-order low-pass filter in each recursive step, such that the repeated differentiation...
of the virtual control inputs can be eliminated. Using dynamic surface control design, a robust adaptive neural control (ANC) design method was developed in Yoo, Park, and Choi (2008) for FJR with model uncertainties using the self-recurrent wavelet NN. The result in Yoo et al. (2006) was further extended to FJR with position measurements (Yoo, Park, & Choi, 2008). It should be pointed out that as a recursive design similar to backstepping, dynamic surface control (DSC) design still needs to employ many function approximators when unmodelled dynamics is presented in the considered systems. This not only requires large computation, but also makes the convergence of estimate parameters verified difficulty.

Meanwhile, so far, most existing intelligent control schemes for FJR only focused on the stability of closed-loop systems without guaranteeing that the estimated parameters converge to their optimal values. Therefore, the optimal parameter values are not capable to be stored and reused, so that for repeating the same or similar control task, the estimated parameters still have to be recalculated online (Ge, Hang, & Zhang, 1999; Ge & Wang, 2002; He, Chen, & Yin, 2016; Liu, Li, Tong, & Chen, 2016; Wang, Liu, Chen, & Zhou, 2014; Yu, Yu, Chen, Gao, & Qin, 2012). From this view of point, the learning ability of FJR is very limited. However, driven by practical specifications, learning and adaptation has become an essential abilities for FJR control systems to improve the control performance in uncertain dynamic environments. Learning in a dynamic environment for control systems is an extremely challenging problem. As stated in Narendra and Annaswamy (2012), the obstruction to achieve the parameter convergence is mainly caused by the verification of a persistent excitation (PE) condition. Recently, a deterministic learning mechanism (Wang & Hill, 2006) was proposed to verify that a partial PE condition can be satisfied by localized radial basis function (RBF) NNs along recurrent trajectories. Subsequently, the proposed method (Wang & Hill, 2006) achieves the convergence of partial NN weight estimates and accurate approximation of unknown dynamics. By presenting an extension of a recent result on stability analysis of linear time-varying (LTV) systems, the deterministic learning mechanism was further developed for nth-order affine/nonaffine systems in a normal form (Dai, Wang, & Wang, 2014; Liu, Wang, & Hill, 2009). To address the learning and control problem of unknown cascaded nonlinear systems, a few elegant dynamic learning methods were proposed by combining a recursive design with a system decomposition strategy (Wang, Wang, Liu, & Hill, 2012; Wang & Wang, 2015a, 2015b). Furthermore, the learning mechanism was also applied into some physical systems such as marine surface vessels (Dai, Wang, & Luo, 2012; Dai, Zeng, & Wang, 2016), robot manipulators (Wang, Ye, and Chen, 2017b), and gait recognition (Zeng & Wang, 2016). For FJR with unknown system dynamics, two or three NN approximators will be used in a recursive design process. However, in order to store the unknown knowledge and reuse the stored information in the same or similar control task, the convergence of neural weight estimates need to be verified. When multiple NN approximators are employed to approximate the unknown dynamics, the NN convergence of the next subsystem relies strictly on the NN convergence of the previous subsystem. The verification of all neural weight convergence still is a complex and challenging task.

Although deterministic learning can effectively recall the learned knowledge to improve the transient performance of the closed-loop systems, it is not capable to guarantee that the tracking error satisfies the predefined transient performance, such as maximum overshoot, convergence speed. However, a large number of practical systems are subject to predefined transient performance. The violation of the transient performance may cause unsafety problem or system damage. The predefined performance issue is an extremely challenging problem (Peng, Wang, and Wang, 2018). Recently, Bechlioulis and Rovithakis (2008) proposed a robust adaptive neural controller for multi-input multi-output feedback linearizable nonlinear systems with unknown nonlinearities, which achieved prescribed performance. The key idea is to use a performance function for describing the prescribed performance and transform the constrained system into an equivalent unconstrained one. The prescribed performance algorithm was further extended to the tracking control of many nonlinear systems, such as strict feedback nonlinear systems (Han & Lee, 2013; Tong, Chen, and Wang, 2005), robot-tracking control systems (Chen, Wu, Jiang, & Jiang, 2014; Kostarigka, Doulgeri, & Rovithakis, 2013; Teng, Yang, Dai, & Wang, 2016; Wang & Yang, 2017), a servo system (Na, Chen, Ren, & Guo, 2013), and permanent magnet synchronous motors (Chang & Tong, 2017). Using the existing schemes (Bechlioulis & Rovithakis, 2008; Chang & Tong, 2017; Chen et al., 2014; Han & Lee, 2013; Kostarigka et al., 2013; Na et al., 2013; Teng et al., 2016; Tong, 2005; Wang & Yang, 2017), the resulting closed-loop error systems is complex when the studied systems include unmodelled dynamics, so that it is difficult to achieve the convergence of estimate parameters and accurate approximation of unknown system dynamics.

In this paper, we address the dynamic learning control for FJR with prescribed performance. As stated above, the use of multiple NN approximators might break the achievement of neural learning control because the verification of the multiple NN weights convergence is difficult or impossible. To overcome such problem, a novel
system transformation with a set of new state variables is
designed to convert the original FJR system into a normal
system and then a high-gain observer is employed to esti-
mate the unavailable new state variables. The main bene-
fit of the transformation is that only one NN approximator
is used to identify uncertain system nonlinear dynamics,
which extremely simplifies the controller design and the
verification on the convergence of neural weights. Con-
sequently, a stable ANC scheme is proposed for the FJR
system and then a high-gain observer is employed to esti-
mate the states. So, we design an observer to estimate the states
in the transformed system (2) are not available since
the system (1) are all measurable, but the states
are chosen such that the polynomial \( s^3 + 1 \) is Hurwitz. According to the conditions of Lemma 2
and \( h_k \) holds.

**Remark 2.1:** For the considered flexible joint manipu-
lator (1), multiple NN approximators have to be employed
to identify the unknown system dynamics using the existing ANC methods. This makes the controller design
become a complicated procedure. Simultaneously, it is a

To simplify the controller design, we define \( x_1 = q_1, \)
\( x_2 = q_1, x_3 = q_2, x_4 = \dot{x}_1 \) as the new state variables.
Therefore, the system (1) can be transformed into a normal
system as follows:

\[
\begin{align*}
\dot{x}_1 &= x_{i+1} & i = 1, 2, 3 \\
\dot{x}_4 &= F(\ddot{q}) + Gu \\
y &= x_1,
\end{align*}
\]

(2)

where \( G = K/\nu \), \( \ddot{q} = [q_1, q_1, q_2, q_2] \in \mathbb{R}^4 \), and

\[
F(\ddot{q}) = M G L q_1^2 \sin(q_1)/l + K^2/(J)(q_1 - q_2) \\
+ [M G L \cos(q_1) + K][M G L \sin(q_1) \\
+ K(q_1 - q_2)^2]/l^2.
\]

It should be pointed out that \( F(\ddot{q}) \) is unknown smooth
function and \( G \) is unknown constant due to the unknown
system dynamics including these parameters \( I, M, L, J \) and
and \( K \).

The system states \( q_1, q_1, q_2, q_2 \) in robot manipulator
system (1) are all measurable, but the states \( x_3 \) and \( x_4 \) in
the transformed system (2) are not available since the
dynamics of the considered system (1) are uncertain. So, we design an observer to estimate the states \( x_3 \) and \( x_4 \). According to the following Lemma, the high-gain
observer used in Behtash (1990) is chosen in this paper.

**Lemma 2.1:** Consider the following high-gain observer system:

\[
\begin{align*}
p\dot{v}_i &= v_{i+1} & i = 1, 2 \\
p\dot{v}_3 &= -c_1 v_3 - c_2 v_2 - v_1 + \psi(t) \\
\psi(t) &= x_2,
\end{align*}
\]

(3)

where \( p \) is a small and positive designed constant, \( v = [v_1, v_2, v_3] \in \mathbb{R}^3 \) are observer states, and the parameters
\( c_i(i = 1, 2) \) are chosen such that the polynomials \( s^3 + c_1 s^2 + c_2 s + 1 \) is Hurwitz. According to the conditions of Lemma 2
in Wang and Wang (2015a), the following condition can be
guaranteed:

\[
\frac{v_{i+1}}{p^i} - \psi^{(i)} = -p \phi^{(i+1)} & \quad i = 1, 2, 3,
\]

(4)

where \( \phi = v_3 + c_1 v_2 + c_2 v_1, \psi^{(i)} \) and \( \phi^{(i)} \) denote the derivative of \( \psi(t) \) and \( \phi \), respectively. There exist positive
constants \( t^* \) and \( h_k \), such that for all \( t > t^* \) and \( |\phi| \leq h_k \) holds.

**Remark 2.1:** For the considered flexible joint manipula-
tor (1), multiple NN approximators have to be employed
to identify the unknown system dynamics using the existing ANC methods. This makes the controller design
become a complicated procedure. Simultaneously, it is a

**2. Problem formulation and preliminaries**

**2.1. System description**

In this paper, we consider a single-link flexible joint
manipulator which can rotate in vertical plane. The
dynamic equations for this system are given by the fol-
lowing differential equations (Slotine & Li, 1991):

\[
\begin{align*}
lq_1 + M G L \sin(q_1) + K(q_1 - q_2) &= 0 \\
Jq_2 - K(q_1 - q_2) &= u,
\end{align*}
\]

(1)

where \( q_1 \in \mathbb{R}, q_1 \in \mathbb{R}, q_1 \in \mathbb{R} \) are the angular position,
velocity and acceleration of manipulator, respectively.
Similarly, \( q_2 \in \mathbb{R}, q_2 \in \mathbb{R}, q_2 \in \mathbb{R} \) are the shaft angle,
velocity and acceleration of motor, respectively. \( u \in \mathbb{R} \) is the
control input used to denote the motor torque. The parameter \( J \) denotes the rotary inertia of motor. \( M \) and
\( I \) are the mass and inertia of manipulator, respectively. \( L \)
describes the distance from the centroid of the manipu-
lator to the end of the manipulator. \( K \) is the param-
eter which depicts the joint flexibility. In this paper, the
dynamics (including these system parameters) of the
robot manipulators (1) are uncertain.
challenge to obtain the convergence of NN weights for high-order flexible joint manipulator (1). To solve such problem, the original FJR system (1) is transformed into a normal system (2) by designing the new state variables. The transformed normal system can effectively reduce the number of NN approximators, but it contains partial unmeasurable states $x_3$ and $x_4$. As a result, a three-order observer (3), instead of the all-state observer, is employed to estimate these unmeasurable states.

The desired reference trajectory $y_d$ is chosen as a recurrent signal, which is obtained by the following reference model:

\[
\begin{align*}
    \dot{x}_{d1} &= x_{d1} + 1, \
    \dot{x}_{d2} &= f_d (\bar{x}_d) \
    y_d &= x_{d1},
\end{align*}
\]

where $f_d(\bar{x}_d) \in \mathbb{R}^n$ is a known smooth function and $\bar{x}_d = [x_{d1}, x_{d2}, x_{d3}, x_{d4}] \in \mathbb{R}^4 \subset \Omega_d$ is the state vector of the reference model with $\Omega_d$ being a compact set.

### 2.2. RBF network and spatially localized approximation

The RBF NNs have been widely employed as function approximators due to some significant merits including the linear structure, the universal approximation property and the good learning ability (Sanner & Slotine, 1992). Therefore, for any smooth function $Q(X): \mathbb{R}^l \rightarrow \mathbb{R}$ over a compact set $\Omega \subset \mathbb{R}^l$, there exists an arbitrary small value $\eta$ such that $\sup_{X \in \Omega} |Q(X) - W^T S(X)| < \eta$ holds, where $W \in \mathbb{R}^N$ being weights vector with NN nodes $N$ and $S(X) = [s_1(X), s_2(X), \ldots, s_N(X)]^T \in \mathbb{R}^N$ being the RBF vector. The RBF $s_i(X)$ is chosen as:

\[
s_i(X) = \exp[-(X - \xi_i)^T (X - \xi_i)/\zeta_i^2],
\]

where $\xi_i$ and $\zeta_i$ are the centre and width of the receptive field, respectively. With sufficiently large node number $N$ and suitable values $\xi_i$, $\zeta_i$, the RBF NN $W^T S(X)$ can approximate any smooth function $Q(X)$ to any arbitrary accuracy as

\[
Q(X) = W^T S(X) + \eta(X) \quad \forall X \in \Omega_X,
\]

where $W^*$ are the ideal weights, $\eta(X)$ is the approximation error, and $|\eta(X)| \leq \eta^*$ with $\eta^* > 0$ being a small value for all $X \in \Omega_X$.

In Wang and Hill (2006), it has been proven that for any bounded trajectory $X(t)$ over the compact set $\Omega_X$, the continuous function $Q(X)$ can be approximated using a limited number of neurons, which is located in a local region close to the trajectory:

\[
Q(X) = W^T S(X) + \eta(X) = W^* \zeta c S_c(X) + \eta_c(X),
\]

where $\zeta$ represents the region around the trajectory $X(t)$, and the approximation error $\eta_c(X)$ satisfies that $|\eta_c(X) - \eta(X)|$ is small.

The convergence of estimate parameters in adaptive system (Gorinevsky, 1995) is a challenging work since the PE condition is difficultly verified a priori. According to the previous results in Wang and Hill (2006), it has been proven that the regressor subvector $S_c(X)$, along any recurrent orbit $X(t)$, satisfies the PE condition. The following lemma shows the PE property of RBF NN.

**Lemma 2.2 (Partial PE for RBF NN (Wang & Hill, 2006)):**

Consider any continuous recurrent orbit $X(t)$, and assume that $X(t)$ remains in a bounded compact set $\Omega_X$. Then, for the RBF NN $W^T S(X)$ with centres laid on a regular lattice which is large enough to cover the compact set $\Omega_X$, the regression subvector $S_c(X)$ consisting of RBFs close to the recurrent orbit $X(t)$ is persistently exciting.

### 3. ANC with prescribed performance

In this section, our main work is to develop a novel adaptive neural controller which guarantees the convergence of the tracking error and achieves the prescribed tracking performance at the same time. In order to avoid using a large number of NN approximators and verify the convergence of NN weights easily, the flexible joint robot system (1) is transformed into the normal form (2). For the transformed system (2), the control objectives are not changed. As a result, we introduce a performance function which describes the prescribed performance such as convergence rate, maximum overshoot and steady-state error. To solve the restricted tracking control problem, a performance transformation technique is introduced to transform the constrained tracking error into equivalent unconstrained one. Finally, combining the backstepping method and employing RBF NN approximators, we design an adaptive neural controller and present the corresponding stability analysis.

#### 3.1. Constrained error transformation

According to Lemma 2.1 and Equation (4), it can be included that $v_2/p, v_3/p^2$ can well estimate the unmeasurable states $x_3, x_4$. Then, to facilitate analysis, define the following error vectors:

\[
e = [e_1, e_2, e_3, e_4]^T = [x_1 - x_{d1}, x_2 - x_{d2}, x_3 - x_{d3}, x_4 - x_{d4}]^T.
\]
\[ \dot{e} = [e_1, e_2, \dot{e}_3, \dot{e}_4]^T = [x_1 - x_{d1}, x_2 - x_{d2}, v_2/p - x_{d3}, v_3/p^2 - x_{d4}]^T, \quad (10) \]
\[ \dot{\hat{e}} = \hat{e} - e = [\hat{\dot{e}}_1, \hat{\dot{e}}_2, \hat{\dot{e}}_3, \hat{\dot{e}}_4] = [0, 0, v_2/p - x_3, v_3/p^2 - x_4]^T, \quad (11) \]
where \( e \) is an ideal error vector because of the unmeasurable \( x_3, x_4 \), \( \hat{e} \) is an actual error vector by using observer states, and \( \hat{e} \) refers to the observer error vector.

In this paper, the tracking error \( e_1 = x_1 - x_{d1} \) is required to satisfy the following prescribed performance condition:
\[ -\sigma \rho(t) < e_1(t) < \overline{\sigma} \rho(t), \quad (12) \]
where \( \sigma \) and \( \overline{\sigma} \) are positive design constants, \( \rho(t) \) denotes a performance function chosen as
\[ \rho(t) = (\rho_0 - \rho) e^{-st} + \rho_\infty, \quad (13) \]
where \( \rho_0 > \rho_\infty \) and \( s > 0 \) are predefined parameters. From (13), the performance function \( \rho(t) \) is positive, smooth and exponential decreasing with \( \lim_{t \to \infty} \rho(t) = \rho_\infty > 0 \).

According to the performance function (13) and the error restricted condition (12), the prescribed performance can be detailed as follows:

1. \( \overline{\sigma} \rho_0 \) and \( -\sigma \rho_0 \) denote the upper bound and the lower bound of the overshoot of tracking error \( e_1 \), respectively.
2. The lower boundness on the convergence rate of \( e_1 \) is confirmed by a decreasing rate \( s \) which introduces by the exponential term \( e^{-st} \).
3. The upper bound and the lower bound of the allowable steady-state tracking error \( e_1 \) are defined as \( \overline{\sigma} \rho_\infty \) and \( -\sigma \rho_\infty \), respectively.

Through tuning the parameters \( \sigma \) and \( \overline{\sigma} \), \( s \), \( \rho_0 \), and \( \rho_\infty \) probably, we can obtain different prescribed tracking error performances to satisfy the real industrial demands.

To achieve trajectory tracking control with prescribed performance, a performance transformation method is introduced to convert the constrained error \( e_1(t) \) into an equivalent unconstrained error \( z_1(t) \). Define a smooth and strictly increasing function \( \Phi(z_1) \) and satisfying the following conditions
\[ -\sigma < \Phi(z_1) < \overline{\sigma}, \quad \forall z_1 \in L_\infty \]
\[ \lim_{z_1 \to +\infty} \Phi(z_1) = \overline{\sigma}, \quad \lim_{z_1 \to -\infty} \Phi(z_1) = \sigma, \quad (14) \]
In this paper, the transformed function \( \Phi(z_1) \) is constructed as
\[ \Phi(z_1) = \frac{\overline{\sigma} e^{z_1+\delta} - \sigma}{e^{z_1+\delta} + 1}, \quad (15) \]
where \( \delta = \ln(\sigma/\overline{\sigma}) \). And the illustration of function \( \Phi(z_1) \) is shown in Figure 1.

From (12), (14) and (15), the restricted error condition can be rewritten as:
\[ e_1(t) = \rho(t) \Phi(z_1). \quad (16) \]
It can be proven that the inverse function of \( \Phi(z_1) \) exists, because \( \Phi(z_1) \) is a strictly monotonic increasing function and \( \rho(t) \neq 0 \). So the restricted error \( e_1(t) \) is transformed as the unconstrained error \( z_1(t) \):
\[ z_1(t) = \Phi^{-1}(e_1(t)/\rho(t)) = \ln \left( \frac{\sigma + \overline{\sigma} e_1}{\overline{\sigma}} \right) - \ln \left( \frac{\sigma + \sigma e_1}{\sigma} \right). \quad (17) \]
Then, the derivative of the transformed error is calculated as:
\[ \dot{z}_1 = \frac{\partial \Phi^{-1}(e_1(t)/\rho(t))}{\partial (e_1(t)/\rho(t))} \frac{d(e_1(t)/\rho(t))}{dt} \]
\[ = \gamma (\dot{e}_1 - \beta e_1), \quad (18) \]
where \( \beta = \rho/\rho_0 \) and \( \gamma = 1/(\sigma \rho + e_1) + (1/(\overline{\sigma} \rho - e_1)) > 0 \) since \( -\sigma \rho < e_1 < \sigma \rho \).

Remark 3.1: Note that the designed function \( \Phi(z_1) \) in (15) passes through the coordinate origin from Figure 1. This is a significant feature for the asymptotic convergence of tracking error \( e_1 \). From (15) and (16), it can be guaranteed \( e_1 = 0 \) when \( \Phi(0) = 0 \). In other words, the tracking error \( e_1 \) converges to zero rather than a neighbourhood of zero when the transformed error \( z_1 \) converges to zero.

3.2. ANC and stability analysis

In this subsection, the backstepping design method is introduced based on the unconstrained error variable \( z_1(t) \), and the unknown dynamics of the flexible-joint robot system is approximated by RBF NN.
From (18), the derivative of \( z_1(t) \) can be re-written as
\[
\dot{z}_1 = \gamma (e_2 - \beta e_1) = \gamma (x_2 - x_{d2} - \beta e_1). \tag{19}
\]
Define \( z_2 = x_2 - \alpha_1 \) with \( \alpha_1 \) being chosen as a virtual control, based on backstepping design method, the virtual control \( \alpha_1 \) can be constructed as
\[
\alpha_1 = -k_1 z_1 + x_{d2} + \beta e_1, \tag{20}
\]
where \( k_1 \) is a positive design constant to be specified later. Then, from (19) and (20), it can be obtained that
\[
\dot{z}_1 = \gamma (-k_1 z_1 + z_2). \tag{21}
\]
Combining the definition \( z_2 = x_2 - \alpha_1 \) with the virtual control (20), the derivative of \( z_2 \) is calculated as \( z_2 = x_2 - \dot{\alpha}_1 = x_3 - \dot{\alpha}_1 \). Since the state variable \( x_3 \) is unmeasurable, it cannot be used to design virtual control. Invoking the state estimated error vector \( \tilde{e} \) in (11), the unmeasurable state \( x_3 \) can be replaced with \( x_3 = v_2/p - \tilde{e}_3 \). Then, \( z_2 \) can be re-written as
\[
\dot{z}_2 = v_2/p - f_2(\cdot) + g_2 \tilde{e}_3, \tag{22}
\]
where \( f_2(\cdot) \triangleq f_2(e_1, e_2, \rho, \tilde{\rho}) = (k_1 \gamma \beta + \beta) e_1 - (k_1 \gamma - \beta) e_2 + x_{d3}, g_2 = -1 \) with \( f_2(\cdot) \) being a computable intermediate variable.

In a similar way, define \( z_3 = v_2/p - \alpha_2 \) with \( \alpha_2 \) being chosen as a virtual control, we construct \( \alpha_2 \) as follows
\[
\alpha_2 = -k_2 z_2 - z_1 + f_2(\cdot), \tag{23}
\]
the derivative of \( z_2 \) along (22) is
\[
\dot{z}_2 = -k_2 z_2 - z_1 + z_3 + g_2 \tilde{e}_3, \tag{24}
\]
where \( k_2 \) is a positive design constant to be specified later. Based on (23), the derivative of \( z_3 \) can be calculated as
\[
\dot{z}_3 = v_3/p^2 - \dot{\alpha}_2 = v_3/p^2 - f_3(\cdot) + g_3(\cdot) \tilde{e}_3, \tag{25}
\]
where \( f_3(\cdot) \triangleq f_3(e_1, e_2, \rho, \tilde{\rho}) = x_{d4} - (k_1 \gamma - \beta + k_2) \tilde{e}_3 + [(k_1 k_2 + 1) \gamma + k_2 \beta + k_1 \gamma \beta + 2 \beta - k_1 \gamma] e_2 + [(k_1 k_2 + 1) \gamma \beta + k_2 \beta + k_1 \gamma \beta + 2 \beta - k_1 \gamma ] e_1, g_3(\cdot) \triangleq g_3(e_1, \rho, \tilde{\rho}) = \beta - k_1 \gamma - k_2, g_3(\cdot) \) is a bounded function by choosing the appropriate design parameters \( k_1, k_2 \) and \( \beta \).

At last, define the error \( z_4 = v_3/p^2 - \alpha_3 \) and design a virtual control \( \alpha_3 \) as follows:
\[
\alpha_3 = -k_3 z_3 - z_2 + f_3(\cdot), \tag{26}
\]
where \( k_3 \) is a positive design constant to be specified later. By invoking the virtual control (26), the equation (25) can be re-written as
\[
\dot{z}_3 = -k_3 z_3 - z_2 + z_4 + g_3(\cdot) \tilde{e}_3. \tag{27}
\]
According to Lemmas 2.1 and (4), we have that \( v_4/p^3 = \dot{x}_4 - p \phi^{(4)} \) holds with \( \phi^{(4)} \) being a bounded function. So the derivative of \( z_4 \) is calculated as
\[
\dot{z}_4 = v_4/p^3 - \dot{\alpha}_3 = \dot{x}_4 - p \phi^{(4)} - \dot{\alpha}_3
\]
\[
= F(\tilde{q}) + \dot{\tilde{u}} - p \phi^{(4)} - f_4(\cdot) + g_4(\cdot) \tilde{e}_3
\]
\[
= G[u + Q(X) + z_3] - p \phi^{(4)} + g_4(\cdot) \tilde{e}_3 - Gz_3, \tag{28}
\]
where
\[
Q(X) = \frac{F(\tilde{q}) - f_4(\cdot)}{G} \tag{29}
\]
and \( F(\tilde{q}) \) and \( G \) defined in (2) are unknown, \( g_4(\cdot) \triangleq g_4(e_1, e_2, \rho, \tilde{\rho}) \) is a bounded function by choosing the appropriate design parameters, and \( f_4(\cdot) \triangleq f_4(e_1, e_2, \tilde{e}_3, \tilde{e}_4, \rho, \tilde{\rho}) \) is a computable intermediate variable.

According to (28), the desired control law is designed by
\[
u^* = -k_4 z_4 - z_3 - Q(X), \tag{30} 
\]
where \( k_4 > 0 \) is a design constant. Noting that the system dynamics \( Q(X) \) is unknown, an RBF NN \( W^T S(X) \) is employed to approximate it. Then, we have
\[
Q(X) = W^T S(X) + \eta, \tag{31} 
\]
where the NN input vector is \( X = [q_1, \tilde{q}_1, q_2, \tilde{q}_2, f_4(\cdot)]^T \in \mathbb{R}^S \), \( W^* \) is an ideal constant weight vector and \( |\eta| \leq \eta^* \) is the NN approximation error with an any small positive constant \( \eta^* \). Let \( \hat{W} \) be the estimated weights of \( W^* \), and define the estimate error \( \hat{W} = \hat{W} - W^* \). Then, the adaptive neural controller can be designed as
\[
u = -k_4 z_4 - z_3 - \hat{W}^T S(X) \tag{32} 
\]
with the updated law of NN weights \( \hat{W} \) given by
\[
\dot{\hat{W}} = \dot{\hat{W}} = \Gamma \left[ S(X) z_4 - \sigma \hat{W} \right], \tag{33} 
\]
where \( \Gamma > 0 \) is the designed diagonal matrix, and \( \sigma > 0 \) is a small value for improving the robustness of the adaptive controller.

Then, substituting (32) into (28) yields
\[
\dot{z}_4 = G[-k_4 z_4 - \hat{W}^T S(X) + \eta] + \vartheta, \tag{34} 
\]
where \( \vartheta = g_4(\cdot) \tilde{e}_3 - p \phi^{(4)} - Gz_3 \) is a small perturbation by choosing small enough \( p \). Subsequently, we obtain the
Theorem 3.1 (Stability and Tracking): Consider the closed-loop system consisting of the flexible joint manipulator system (1) with the prescribed performance restriction (12), the reference recurrent model (5), the error transformation (17), the ANC law (32) together with neural weights updating laws (33). If the bounded initial conditions satisfy the prescribed performance (12), there exists a finite time $T$ such that the proposed control scheme ensures:

- all the signals in the closed-loop system are uniformly ultimately bounded (UUB);
- the tracking error $e(t)$ converges to a small neighborhood of zero while satisfying the prescribed performance restriction (12).

Proof: See Appendix 1.

4. Learning from ANC

Based on the proposed stable ANC scheme in Section 3, this section will discuss how to achieve the knowledge acquisition, expression, and storage of the unknown system dynamic using only a limited number of neurons located in a local region along the trajectory. Without the accurate NN weights convergence, it is hard to acquire and store the knowledge of the time-varying NN weights. It has been shown in Wang and Hill (2006) that the convergence of NN weights is related to PE condition of regression vector $S(X)$. From Lemma 1, the PE condition can be achieved if we can verify the recurrent property of the NN input variables $X$. Subsequently, the converged NN weights can be obtained and stored as experience knowledge.

Using the spatially localized approximation property of the RBF NN presented in (7), the closed-loop system consisting of (33) and (34) can be expressed by

$$
\begin{align*}
\dot{z}_4 &= G[-k_4z_4 - \bar{W}^T \xi(X) + \eta_{\xi}] + \vartheta \\
\dot{\tilde{W}}_\xi &= \bar{\Gamma}_{\xi} [S_{\xi}(X)z_4 - \sigma W_{\xi}] \\
\dot{\bar{W}}_\xi &= \bar{\Gamma}_{\xi} [S_{\xi}(X)z_4 - \sigma W_{\xi}],
\end{align*}
$$

where $S_{\xi}(X)$ is the subvector of $S(X)$, consisting of RBFs that are close to the reference orbit $X(t)$, $\bar{W}_\xi \in R^\ell$ is the corresponding estimate weights subvector with $0 < \xi < N$, $\bar{\Gamma}_{\xi}$ denotes the region which is far away from the orbit $X(t)$, and $\eta_{\xi} = \bar{\xi}(X)$ is the approximation error along the reference orbit, $||\eta_{\xi}|| < |\eta_{\xi}|$ is close to zero because of the small value of $\bar{W}_\xi S_{\xi}(X)$.

Theorem 4.1 (Knowledge acquisition, representation and storage): Consider the closed-loop system consisting of the flexible joint manipulator system (1) with the prescribed performance restriction (12), the reference recurrent model (5), the error transformation (17), the anc law (32) together with neural weights updating laws (33). If the bounded initial conditions satisfy the prescribed performance (12), it can be obtained that the NN estimate weights $\bar{W}$ converge to small neighborhoods of their optimal values $W^*$ in a finite time $T$, and locally-accurate approximation for the unknown dynamics $Q(X)$ along the recurrent tracking orbit is achieved by $\bar{W}^T S(X)$ and $\bar{W}^T \dot{S}(X)$, where

$$
\bar{W} = \text{mean}_{[t_a,t_b]} \dot{W}(t) = \frac{1}{t_b - t_a} \int_{t_a}^{t_b} \dot{W}(t) \, dt
$$

with $t_b > t_a > T$.

Proof: In order to obtain the exponential stability of the closed-loop system, the system (36) needs to be described as a class of LTV system with very small perturbations whose exponential stability has been proven with some necessary condition (Liu et al., 2009).

$$
\begin{align*}
\begin{bmatrix}
\dot{z}_4 \\
\dot{\tilde{W}}_\xi \\
\dot{\bar{W}}_\xi
\end{bmatrix} &= 
\begin{bmatrix}
A(t) & B^T(t) \\
-C(t) & 0
\end{bmatrix} 
\begin{bmatrix}
z_4 \\
W_{\xi} \\
\bar{W}_{\xi}
\end{bmatrix} + 
\begin{bmatrix}
\bar{G}\eta_{\xi} + \vartheta \\
-\sigma \bar{\Gamma}_{\xi} \dot{W}_{\xi}
\end{bmatrix},
\end{align*}
$$

where $A(t) = -k_4G \in R$, $B(t) = -GS_{\xi}(X) \in R^\ell$, $C(t) = -\bar{\Gamma}_{\xi} S_{\xi}(X) \in R^\ell$.

However, for the actual flexible manipulator, the large joint flexibility parameter $K$ and the inertia parameters $I_j$ may cause the large term $G$, and the disturbance $G\eta_{\xi}$ may be large due to the possible large $G$. The existing result of the exponential convergence becomes invalid. Therefore, the following discussion is given in this paper:

1. When the term $G$ is a small value, $G\eta_{\xi}$ will become a small perturbation. As a result, the system (38) can be regarded as a class of LTV system with small perturbations.

2. When the term $G$ is a large value, a state transformation $z = z_4/G$ is introduced to make the perturbation $G\eta_{\xi}$ small. Then, the system (38) can be
transformed as
\[
\begin{bmatrix}
  \dot{z}_t \\
  \dot{W}_t
\end{bmatrix} = \begin{bmatrix}
  A(t) & B^T(t) \\
  -C(t) & 0
\end{bmatrix} \begin{bmatrix}
  z_t \\
  W_t
\end{bmatrix} + \begin{bmatrix}
  \eta_t + \theta/G \\
  -\sigma \Gamma \dot{W}_t
\end{bmatrix},
\]
where \( A(t) = -k_4 G \in R \), \( B(t) = -S_t(X) \in R^2 \), \( C(t) = -G_\Gamma S_t(X) \in R^r \).

The LTV system with small perturbations is exponentially stable if the following conditions are satisfied:

- **C1:** \( B(t) \) satisfies PE condition;
- **C2:** There exists a symmetric and positive matrix \( P(t) \) such that \( P(t) B(t) = C(t) \) holds;
- **C3:** and \( A^T(t) P(t) + P(t) A(t) + \dot{P}(t) < 0 \) holds.

Condition C1 means that \( S_t(X) \) needs to be verified the satisfaction of the PE condition. As stated in Lemma 2.2, the regression vector \( S_t(X) \) is persistently exciting if the NN inputs \( X = [q_1, q_2, q_3, f_4(\cdot)]^T \) are recurrent. From Theorem 3.1 and the stability analysis in Appendix 1, \( z_t \) and \( e_1 \) converge exponentially to small neighbourhoods of zero for \( t > T \). Since \( e_1 = x_1 - x_{d1} \) and \( z_2 = x_2 - \alpha_1 \), the states \( x_1 = q_1 \) and \( x_2 = q_1 \) follow the periodic signals \( x_{d1} \) and \( \alpha_1 \) for all \( t > T \), respectively. Noting \( z_3 = v_2/p - \alpha_2 \), \( v_2/p \) is recurrent with the same period as \( \alpha_2 \). From Lemma 2.1, state \( x_3 \) is also recurrent as \( v_2/p \). By combining the original system (1) and the transformed system (2), it can be concluded that \( q_2 \) is recurrent. Recursively, we can also obtain the recurrent property of \( f_4(\cdot) \). So, the regression vector \( S_t(X) \) satisfies PE condition due to the recurrent inputs \( X(t) \). Meanwhile, a positive \( P(t) \) can be easily found to satisfy the condition C2 and condition C3. For example, by choosing \( P(t) = \Gamma \) or \( P(t) = G \Gamma \) (for large \( G_\Gamma \)), we obtain \( P(t) B(t) = C(t) \) and \( A^T(t) P(t) + P(t) A(t) + \dot{P}(t) < 0 \) if appropriately designing the positive constant \( k_4 \).

Based on the perturbation theory from Lemma 4.6 in Khalil (1996) and \( \dot{W}_t = W_t - \hat{W}_{\infty} \), \( \dot{W}_t \) converge exponentially to small neighbourhoods of optimal weights \( \hat{W}_{\infty} \) for \( t > T \). Therefore, the constant convergent weights \( \hat{W}_{\infty} \) can be obtained according to (37), and the unknown system dynamics can be approximated by
\[
Q(X) = \dot{W}_t^T S_t(X) + \eta_{t1} = \hat{W}_{\infty}^T S_t(X) + \eta_{t2},
\]
where \( \eta_{t1} \) and \( \eta_{t2} \) are close to the ideal approximation error \( \hat{\eta}^o \) because the small convergence errors \( \dot{W}_t \). Noting that \( \hat{\xi} \) denotes the region which is far away from the orbit \( X(t) \), the NN weights \( \dot{W}_t \) are not activated and updated when \( \dot{W}(0) \) is chosen, in other words \( \dot{W}_t \) remain small (close to zero). So (40) can be rewritten as
\[
Q(X) = \dot{W}_t^T S_t(X) + \eta_{t2} = \hat{W}_{\infty}^T S(X) + \eta_{t2},
\]
where \( \eta_{t2} = \eta_{t2} - \hat{W}_{\infty}^T S(X) \) with \( ||\eta_{t2}|| = ||\eta_{t2}|| \) being small. This completes the proof.

5. Learning control using learned knowledge

In the previous section, it has been proven how to achieve the knowledge acquisition, expression, and storage of unknown dynamics \( Q(X) \). In this section, the stored NN weights \( \dot{W} \) in (37) will be reused to construct a neural learning controller for the same control task. The control performance will be improved with the following neural learning controller
\[
u = -k_4 z_4 - z_3 - \hat{W}_t S(X),
\]
where the learned constant NN weights \( \dot{W} \) are obtained from (37), the definition of \( z_3 \) and \( z_4 \) are same as the ANC.

---

Figure 2. The framework of the flexible robot controller.
process in Section 3. For clarity purposes, the schematic diagram of the proposed neural learning control scheme framework is shown in Figure 2. In what follows, the following results can be obtained based on the proposed neural learning controller (42).

**Theorem 5.1 (Knowledge reuse):** Consider the closed-loop system consisting of the flexible joint manipulator system (1) with the prescribed performance restriction (12), the reference recurrent model (5), the error transformation (17), and the neural learning control law (42) together with the constant NN weights obtained by (37). If the recurrent reference orbit $y_d$ is as similar as the orbit in Theorem 3.1, and the corresponding initial conditions satisfy the prescribed bound (12), we have that all signals in the closed-loop system remain UUB, and the tracking error $e_1$ exponentially converges to a small neighbourhood of zero with subjecting to the prescribed performance constraints (12).

**Proof:** See Appendix 2.

### 6. Simulation results

In this section, the simulation of a fourth-order one-link flexible manipulator is performed to test the effectiveness of the proposed ANC scheme and the NN learning control scheme. Considering the dynamic model of the one-link flexible manipulator system in (1), the actual parameters in the system are chosen as $M = 2.3$ kg, $L = 1$ m, $g = 9.8$ m/s², $I = 2.3$ kg m², $J = 0.5$ kg m², $k_m = 15$. And the recurrent reference trajectory $y_d$ is chosen as

$$y_d = x_{d1} = \frac{45\pi}{360} \sin(t) + \frac{70\pi}{360} \cos\left(\frac{\pi}{4} t\right),$$

which is shown in Figure 3.

In this paper, the tracking error $e_1$ is required to satisfy the prescribed performance which is described as

$$-\rho(t) < e_1(t) < \rho(t)$$

$$\rho(t) = (1.2 - 0.05)e^{-t} + 0.05,$$

where $\sigma = 1$ and $\varpi = 1.2$.

#### 6.1. Learning from stable ANC

Based on the dynamic form (29) of $Q(X)$, the NN input variables are chosen as $X = [q_1, \ldots, q_1, q_2, \ldots, q_2]$, and the RBF network $\hat{W}(X)$ is constructed with 6561 nodes (i.e. $N = 6561$) whose centres $\xi_i$ are evenly spaced on $[-1.2, 1.2] \times [-1.2, 1.2] \times [-2, 2] \times [-2, 2]$ with the widths $\varsigma_1 = \varsigma_2 = 0.375$ and $\varsigma_3 = \varsigma_4 = 0.625$ in this simulation. The parameters of the high-gain observer (3) are selected as $\rho = 0.008$, $c_1 = 3$ and $c_2 = 2$. The controller parameters are designed as $k_1 = k_2 = k_3 = 0.5$, $k_4 = 2$, $\Gamma = 12$, $\sigma = 0.0001$. The initial system states are $[q_1(0), q_1(0), q_2(0), q_2(0)] = [1.5, 0.3, 0.3, 0.3]$, the initial observer states are $[v_1(0), v_2(0), v_3(0)] = [0.3, 0.0, 0.0]$, and NN initial weights values are $\hat{W}(0) = 0$.

In order to show the better transient performance of the proposed adaptive neural controller (32)–(33), the simulation comparison is performed with the existing adaptive DSC method in Wang and Wang (2015a) without prescribed performance. For contrast, we choose the flexible manipulator (1) with the same system parameters and the same reference trajectory (43). The transient and steady performance of $e_1$ with both control schemes is shown in Figure 4. It can be seen that the faster convergence speed and the smaller overshoot of the tracking error $e_1$ can be obtain with the proposed ANC method than the DSC method in Wang and Wang (2015a). From Figure 4, the tracking error obtained (Wang & Wang, 2015a) not only violates the prescribed performance (44), but also shows damped oscillations. On the other hand, according to Appendix 1, the tracking error $e_1$ of the proposed ANC method can be made arbitrarily small by choosing the large $k_2$, $k_4$ and the small $\sigma$. In particular, two NN approximators have

Figure 3. Reference trajectory $y_d$ (‘-’) and system state output $q_1$ (‘—’).

Figure 4. Tracking error $e_1$ with prescribed performance in this paper (‘-’), DSC without prescribed performance in (Wang & Wang, 2015a) (‘--’), and the prescribed error bounds(‘-’).
Figure 5. Unmeasurable system states (‘-’), and estimated states (‘--’).

to been used to identify uncertain system dynamics of the flexible manipulator (1), which makes time consumption very larger than the proposed method in this paper. For example, based on the same computer setup, the actual time consumption is 2548 and 4005 s, respectively, using the proposed method in this paper and the existing method in Wang and Wang (2015a). Figure 5 indicates that the state observer outputs can well estimate unmeasurable system states. The bounded control input $u$ is shown in Figure 6. From Figures 3–6, the proposed adaptive neural controller (32) – (33) guarantees all the signals of the closed-loop systems are bounded and the tracking error remains in the predefined bound (44) all the time.

6.2. Learning control with experience reuse

After achieving stable ANC in Section 6.1, the main objectives of this section are to show that the neural weight estimates $\hat{W}$ can exponentially converge to constant weights $\bar{W}$; the unknown system dynamics $Q(X)$ in (29) can be accurately identified by the constant RBF NNs $\bar{W}^T S(X)$, and the constant RBF NNs can be reused to design a neural learning controller for the improved control performance of the same control task.

As stated in Section 4, we can verify the regression vector $S(x)$ satisfies the partial PE condition when system output $y_d$ tracks the given periodic reference trajectory (see Figure 3 for the output tracking performance). Under the PE condition, the NN weight estimates $\hat{W}$ can converge to the ideal NN weights $\bar{W}$, which is shown in Figure 7. From Figure 7, the NN weight estimates achieves the good convergence within the time interval [450s, 500s]. Therefore, based on Theorem 4.1, the converged constant NN weights $\bar{W}$ can be calculated by

$$\bar{W} = \text{mean}_{t \in [450, 500]} \hat{W}(t) = \frac{1}{50} \int_{450}^{500} \hat{W}(\tau) d\tau. \quad (45)$$

The constant NN weights $\bar{W}$ in (45) is easily stored for the experience knowledge instead of the time-varying NN weight estimates $\hat{W}$. Invoking the stored constant NN weights $\bar{W}$ in (45), we can design a neural learning controller (42). To show the improved control performance for the same control task using the experience knowledge, a simulation comparison is given between the proposed ANC (32)–(33) and the proposed learning controller (42) with the experience knowledge (45). For comparison, in the simulation, the flexible manipulator (1) is used again as the plant to be controlled by the neural learning controller (42). The reference trajectory and the system initial condition are taken to be the same as those used in Section 6.1. Simulation results are shown

Figure 6. Control input $u$. 
in Figures 8–9. From Figure 8, it can be seen that the unknown system dynamics $Q(X)$ can be well approximated by the constant RBF NN $\bar{W}^T S(X)$ with the experience knowledge (45). From Figure 9, it is clear that the tracking error transient performance in learning control phase is better than in ANC phase. Moreover, comparing with the ANC (32)–(33), the proposed neural learning controller effectively recalls the experience knowledge (45), so that the running time is saved nearly 2/3 and the computational burden is also greatly eased due to avoiding to recalculate NN weight estimates tuned in ANC process.

7. Conclusions

In this paper, we addressed learning from ANC of FJR with the predefined performance constraint. By combining a system transformation method and the high-gain observer, the proposed ANC scheme guaranteed that the tracking error converged to a small residual set of zero in a finite time $T$ and all the signals in the closed-loop system were UUB. Simultaneously, the prescribed performance could be guaranteed by the error transformation method, which converted the original constrained tracking error control issue into the equivalent unconstrained error stabilization issue. Specially, the proposed adaptive neural control scheme only use one NN approximator, so that we could easily verify the partial PE condition of RBF NN. Under the partial PE condition, the proposed ANC achieved the knowledge acquisition, representation and storage of unknown system dynamics. Using the stored experience knowledge without the online NN adjustment, the neural learning control scheme was proposed
to obtain a better transient and steady-state performance for the same or similar control task. Some further actions can be taken to extend the proposed scheme to solve different kinds of control problems including the output-feedback control (Wang, Liu, Li, & Wang, 2018; Wang, Liu, & Shi, 2017a) and the leader–follower formation control (He, Wang, Dai, and Luo, 2018).

**Disclosure statement**

No potential conflict of interest was reported by the authors.

**Funding**

This work was supported in part by the National Natural Science Foundation of China [grant numbers 61773169 and 61473121], the Guangdong Natural Science Foundation [grant numbers 2017A030313381, 2017A030313369 and 2014A030312005], the Science and Technology Program of Guangzhou [grant number 201607010006], and the Fundamental Research Funds for the Central Universities.

**ORCID**

Min Wang http://orcid.org/0000-0001-7025-7651

**References**

Al-Ashoor, R. A., Patel, R. V., & Khorasani, K. (1993). Robust adaptive controller design and stability analysis for flexible-joint manipulators. *IEEE Transactions on Systems, Man and Cybernetics*, 23(2), 589–602.

Bechlioulis, C. P., & Rovithakis, G. A. (2008). Robust adaptive control of feedback linearizable mimo nonlinear systems with prescribed performance. *IEEE Transactions on Automatic Control*, 53(9), 2090–2099.

Behtash, S. (1990). Robust output tracking for non-linear systems. *International Journal of Control*, 51(6), 1381–1407.

Benosman, M., & Le Vey, G. (2004). Control of flexible manipulators: A survey. *Robotica*, 22(5), 533–545.

Chang, W., & Tong, S. (2017). Adaptive fuzzy tracking control design for permanent magnet synchronous motors with output constraint. *Nonlinear Dynamics*, 87(1), 291–302.

Chen, M., Wu, Q. X., Jiang, C. S., & Jiang, B. (2014). Guaranteed transient performance based control with input saturation for near space vehicles. *Science China Information Sciences*, 57(5), 1–12.

Dai, S. L., Wang, C., & Luo, F. (2012). Identification and learning control of ocean surface ship using neural networks. *IEEE Transactions on Industrial Informatics*, 8(4), 801–810.

Dai, S.-L., Wang, C., & Wang, M. (2014). Dynamic learning from adaptive neural network control of a class of nonaffine nonlinear systems. *IEEE Transactions on Neural Networks and Learning Systems*, 25(1), 111–123.

Dai, S. L., Wang, M., & Wang, C. (2016). Neural learning control of marine surface vessels with guaranteed transient tracking performance. *IEEE Transactions on Industrial Electronics*, 63(3), 1717–1727.

Ge, S. S., Hang, C., & Zhang, T. (1999). A direct method for robust adaptive nonlinear control with guaranteed transient performance. *Systems and Control Letters*, 37(5), 275–284.

Ge, S. S., Lee, T. H., & Tan, E. (1998). Adaptive neural network control of flexible joint robots based on feedback linearization. *International Journal of Systems Science*, 29(6), 623–635.

Ge, S. S., & Wang, C. (2002). Adaptive NN control of uncertain nonlinear pure-feedback systems. *Automatica*, 38(4), 671–682.

Gorinevsky, D. (1995). On the persistency of excitation in radial basis function network identification of nonlinear systems. *IEEE Transactions on Neural Networks*, 6(5), 1237–1244.

Han, S. I., & Lee, J. M. (2013). Improved prescribed performance constraint control for a strict feedback non-linear dynamic system. *IET Control Theory & Applications*, 7(14), 1818–1827.

He, W., Chen, Y., & Yin, Z. (2016). Adaptive neural network control of an uncertain robot with full-state constraints. *IEEE Transactions on Cybernetics*, 46(3), 620–629.

He, S., Wang, M., Dai, S.-L., & Luo, F. (2018). Leader–follower formation control of USVs with prescribed performance and collision avoidance. *IEEE Transactions on Industrial Informatics*. doi:10.1109/TII.2018.2839739

Huang, A.-C., & Chen, Y.-C. (2004). Adaptive sliding control for single-link flexible-joint robot with mismatched uncertainties. *IEEE Transactions on Control Systems Technology*, 12(5), 770–775.

Jiang, Y., Liu, Z., Chen, C., & Zhang, Y. (2015). Adaptive robust fuzzy control for dual arm robot with unknown input dead-zone nonlinearity. *Nonlinear Dynamics*, 81(3), 1301–1314.

Khalil, H. K. (1996). *Nonlinear systems*. New Jersey: Prentice-Hall.

Kostarigka, A. K., Doulgeri, Z., & Rovithakis, G. A. (2013). Prescribed performance tracking for flexible joint robots with unknown dynamics and variable elasticity. *Automatica*, 49(5), 1137–1147.

Krstic, M., Kanellakopoulos, I., & Kokotovic, P. V. (1995). *Nonlinear and adaptive control design*. New York, Wiley.

Li, Y., Tong, S., & Li, T. (2013). Adaptive fuzzy output feedback control for a single-link flexible robot manipulator driven DC motor via backstepping. *Nonlinear Analysis: Real World Applications*, 14(1), 483–494.

Liu, Y.-J., Li, J., Tong, S., & Chen, C. P. (2016). Neural network control-based adaptive learning design for nonlinear systems with full-state constraints. *IEEE Transactions on Neural Networks and Learning Systems*, 27(7), 1562–1571.

Liu, T., Wang, C., & Hill, D. J. (2009). Learning from neural control of nonlinear systems in normal form. *Systems and Control Letters*, 58(9), 633–638.

Miao, Z., & Wang, Y. (2013). Robust dynamic surface control of flexible joint robots using recurrent neural networks. *Journal of Control Theory and Applications*, 11(2), 222–229.

Na, J., Chen, Q., Ren, X., & Guo, Y. (2013). Adaptive prescribed performance motion control of servo mechanisms with friction compensation. *IEEE Transactions on Industrial Electronics*, 61(1), 486–494.

Narendra, K. S., & Annaswamy, A. M. (2012). *Stable adaptive systems*. New York: Courier Corporation.

Nicosia, S., & Tomei, P. (1990). Robot control by using only joint position measurements. *IEEE Transactions on Automatic Control*, 35(9), 1058–1061.

Oh, J. H., & Lee, J. S. (1999). Control of flexible joint robot system by backstepping design approach. *Intelligent Automation and Soft Computing*, 5(4), 267–278.

Ozgoli, S., & Taghirad, H. (2006). A survey on the control of flexible joint robots. *Asian Journal of Control*, 8(4), 332–344.
Choose the Lyapunov function candidate as follow:

$$V_z = \frac{z_1^2}{2\gamma} + \frac{z_2^2}{2} + \frac{z_3^2}{3} + \frac{z_4^2}{2G}.$$  \hspace{1cm} (A1)

According to (35), the derivative of $V_z$ is calculated as

$$\dot{V_z} = \frac{z_1z_1}{\gamma} - \frac{\gamma}{2\gamma} z_2^2 + z_2z_2 + \frac{z_3^2}{z_2} - \frac{z_4z_4}{G} = -\sum_{i=1}^{4} k_i z_i^2 + \frac{g_4 z_4^3}{G} - \frac{z_4 g_4 (\cdot) z_4 (\cdot) z_4 z_4 (\cdot) z_4}{G} - \frac{z_4 g_4 (\cdot) z_4 x}{G} - \frac{z_4 g_4 (\cdot) x}{G}.$$  \hspace{1cm} (A2)

According to Lemma 2.1 and $\dot{\epsilon}_3 = v_2 - p$, the following inequalities holds that

$$\begin{align*}
g_i (\cdot) z_i \leq & g_i (\cdot) z_i \rho \phi(\cdot) \leq k_2 z_i^2 + \frac{g_i^2 \rho^2}{4k_2} z_i^2 + \frac{g_i^2 \rho^2}{4k_2} z_i^2 + \frac{g_i^2 \rho^2}{4k_2} z_i^2 \leq k_{42} z_i^2 + \frac{g_i^2 \rho^2}{4k_2} z_i^2 + \frac{g_i^2 \rho^2}{4k_2} z_i^2 + \frac{g_i^2 \rho^2}{4k_2} z_i^2,
\end{align*}$$  \hspace{1cm} (A3)

where $k_{42} > 0$, $k_{11} > 0$, $k_{32} > 0$, and $k_{43} > 0$ are design constants with $i = 1, 2, 3$.

Then, substituting (A3) into (A2), we obtain

$$\begin{align*}
\dot{V_z} \leq & -\sum_{i=1}^{4} k_i z_i^2 + \frac{g_i^2 \rho^2}{4k_2} z_i^2 + \frac{g_i^2 \rho^2}{4k_2} z_i^2 + \frac{g_i^2 \rho^2}{4k_2} z_i^2 + \frac{g_i^2 \rho^2}{4k_2} z_i^2
+ \frac{p^2 h_2}{4k_3 G^2} + \frac{g_i^2 \rho^2}{4k_2} z_i^2 - z_4 \tilde{W} \dot{S}(X),
\end{align*}$$  \hspace{1cm} (A4)

where $\dot{k}_i = k_{11} - (\dot{\gamma}/2\gamma)^2$, $k_2 = k_{21} + k_{22}$, $k_3 = k_{31} + k_{32}$, $k_4 = k_{41} + k_{42} + k_{43} + k_4$ and $k_i$ can be designed to guarantee $k_{11} > 0$.

Consider the NN weights errors $\tilde{W}$, we define the Lyapunov function as

$$V = V_z + \frac{1}{2} \tilde{W}^T \Gamma^{-1} \tilde{W}.$$  \hspace{1cm} (A5)
Base on the fact that $-\dot{W}^T \sigma \dot{W} \leq -\frac{\sigma}{2} \| \dot{W} \|^2 + \frac{\sigma}{2} \| W \|^2$, the derivative of $V$ along (33) and (A4) yields

$$\dot{V} = \dot{V}_2 + \dot{W}^T \Gamma^{-1} \dot{W}$$

$$\leq -\sum_{i=1}^{4} k_i z_i^2 + \frac{g_2^2 p_2^2 h_2^2}{4k_2} + \frac{g_3^2 p_2^2 h_2^2}{4k_3} + \frac{g_4^2 p_2^2 h_2^2}{4G^2k_4}$$

$$+ \frac{p^2 h_4^2}{4k_4 G^2} + \frac{\eta^2}{4k_4} - \frac{\sigma}{2} \| \dot{W} \|^2 + \frac{\sigma}{2} \| W \|^2.$$  \hfill (A6)

Consider the form of Lyapunov functions $V$ and $V_2$, the following inequality can be obtained

$$\dot{V} \leq -\kappa V + \varrho,$$  \hfill (A7)

where

$$\kappa = \min\{2\gamma k_{11}, 2k_{21}, 2k_{31}, 2Gk_{41}, \lambda_{\text{min}}(\Gamma) \sigma \} \hfill (A8)$$

$$\varrho = \frac{g_2^2 p_2^2 h_2^2}{4k_2} + \frac{g_3^2 p_2^2 h_2^2}{4k_3} + \frac{g_4^2 p_2^2 h_2^2}{4G^2k_4}$$

$$+ \frac{p^2 h_4^2}{4k_4 G^2} + \frac{\eta^2}{4k_4} - \frac{\sigma}{2} \| W \|^2.$$  \hfill (A9)

Let $\tau = \varrho/\kappa$, then the solution of $V(t)$ satisfies

$$V(t) \leq [V(0) - \tau]e^{-\kappa t} + \tau.$$  \hfill (A10)

Considering the Lyapunov function $V$ along (18), (24), (27) and (A15) can be calculated as

$$\dot{V} = \frac{z_1 z_1^*}{2\gamma} - \frac{\gamma}{2} x_2^2 + z_2^2 + z_3^2 + \frac{z_4^2}{G}$$

$$= -\frac{\sum_{i=1}^{4} k_i z_i^2}{2\gamma} + \frac{g_2 z_2 \tilde{e}_3 + g_3 z_3 \tilde{e}_3 + z_4 \eta_2}{G}$$

$$= -\frac{\sum_{i=1}^{4} k_i z_i^2}{2\gamma} + \frac{g_2 z_2 \tilde{e}_3 + g_3 z_3 \tilde{e}_3 + z_4 \eta_2}{G}$$

$$\leq -\sum_{i=1}^{4} k_i z_i^2 + \frac{g_2^2 p_2^2 h_2^2}{4k_2} + \frac{g_3^2 p_2^2 h_2^2}{4k_3} + \frac{g_4^2 p_2^2 h_2^2}{4G^2k_4}$$

$$+ \frac{p^2 h_4^2}{4k_4 G^2} + \frac{\eta^2}{4k_4}.$$  \hfill (A11)

Using the compact set can be found

$$\Omega_{z_i} = \{ z_i \mid z_i \leq \kappa, x_d \in \Omega_d \}.$$  \hfill (A13)

For example, if we choose $\mu = \sqrt{2\gamma T}$, there exists a finite time $T$ such that for all $t > T$, the transformed error $z_1$ satisfies

$$|z_1| < \mu,$$  \hfill (A12)

where the following compact set can be found

$$\Omega_{z_1} = \{ z_1 \mid z_1 < \mu, x_d \in \Omega_d \}.$$  \hfill (A13)

For example, if we choose $\mu = \sqrt{2\gamma T}$, there exists a finite time

$$T = -\frac{1}{\kappa} \ln \frac{\tau}{V(0) - \tau},$$ \hfill (A14)

that $|z_1| < \mu$ holds for all $t > T$. Note that $\mu$ can be made arbitrarily small by choosing the large $k_2$, $k_4$, $k_1$ and the small $\sigma$ with $i = 2, 3, 4$.

From (A10), it is obvious that the errors $z_1$ to $z_4$ and NN weights errors $W$ are bounded. According to the error transformation $e_1(t) = \rho(t) \Phi(z_1)$, the boundedness of $z_1$ guarantees that the tracking error $e_1$ converges to a small residual set of zero with satisfying the constrained error bound (12). The boundedness of $e_1$ is guaranteed by the bounded $e_1, z_1, x_d$. Based on the bounded $\alpha_1, z_2$ and $x_d$, the system state $x_1, x_2$ are bounded. It follows from $z_3 = v_2/p - \alpha_2$ that $v_2/p$ is bounded because of the bounded $\alpha_2$ and $z_3$. From Lemma 2.1, the observer output $v_2/p$ asymptotically converges to the state $x_3$, so $x_3$ is also bounded. In the same way, the boundedness of $x_4$ and $u$ are also guaranteed. Therefore, all the signals in the closed-loop system are UUB, and the constrained tracking error $e_1$ converges to a small neighbourhood of zero with satisfying the prescribed performance restriction (12). This completes the proof.

### Appendix 2. Proof of Theorem 3

Substituting the neural learning control law (42) into Equation (28), then we have

$$\dot{z}_4 = G[-k_4 z_4 + \eta_2] + g_4(\cdot) \tilde{e}_3 - p \phi^{(S)} - Gz_3.$$  \hfill (A15)

The form of Lyapunov function is chosen as

$$V = \frac{z_1^2}{2\gamma} + \frac{z_2^2}{2} + \frac{z_3^2}{2} + \frac{z_4^2}{G^2}.$$  \hfill (A16)

Similarly to the proof procedure of Theorem 3.1, the derivative of $V$ along (18), (24), (27) and (A15) can be calculated as

$$\dot{V} = \frac{z_1 z_1^*}{2\gamma} + \frac{\gamma}{2} x_2^2 + z_2^2 + z_3^2 + \frac{z_4^2}{G}$$

$$= -\frac{\sum_{i=1}^{4} k_i z_i^2}{2\gamma} + \frac{g_2 z_2 \tilde{e}_3 + g_3 z_3 \tilde{e}_3 + z_4 \eta_2}{G}$$

$$= -\frac{\sum_{i=1}^{4} k_i z_i^2}{2\gamma} + \frac{g_2 z_2 \tilde{e}_3 + g_3 z_3 \tilde{e}_3 + z_4 \eta_2}{G}$$

$$\leq -\sum_{i=1}^{4} k_i z_i^2 + \frac{g_2^2 p_2^2 h_2^2}{4k_2} + \frac{g_3^2 p_2^2 h_2^2}{4k_3} + \frac{g_4^2 p_2^2 h_2^2}{4G^2k_4}$$

$$+ \frac{p^2 h_4^2}{4k_4 G^2} + \frac{\eta^2}{4k_4}.$$  \hfill (A11)

Considering the Lyapunov function $V$ in (A16), we have

$$\dot{V} \leq k_1 V + \varrho,$$  \hfill (A18)

where $k_1 = \min\{2\gamma k_{11}, 2k_{21}, 2k_{31}, 2Gk_{41}\}$ and $\varrho = (g_2^2 p_2^2 h_2^2)/4k_2 + (g_3^2 p_2^2 h_2^2)/4k_3 + (g_4^2 p_2^2 h_2^2)/4G^2k_4 + (p^2 h_4^2)/4k_4$ and $G^2 + (\eta^2)/4k_4$. Then, solving the inequality (A18) and defining $\tau_1 = \varrho/k_1$, we have

$$V(t) \leq [V(0) - \tau_1]e^{-\kappa t} + \tau_1,$$  \hfill (A19)

Based on the similar analysis as Theorem 3.1, the proposed neural learning controller (42) with the learned NN weights $\tilde{W}$ in (37) guarantees that all the signals in the closed-loop system are UUB, and the tracking error $e_1$ also converges to a small neighbourhood of zero with subjecting to the prescribed performance constrains.