Neural Network-Based Adaptive Controller Design of Robotic Manipulators with an Observer

Fuchun Sun, Member, IEEE, Zengqi Sun, Senior Member, IEEE, and Peng-Yung Woo, Member, IEEE

Abstract—A neural network (NN)-based adaptive controller with an observer is proposed in this paper for the trajectory tracking of robotic manipulators with unknown dynamics nonlinearities. It is assumed that the robotic manipulator has only joint angle position measurements. A linear observer is used to estimate the robot joint angle velocity, while NNs are employed to further improve the control performance of the controller system through approximating the modified robot dynamics function. The adaptive controller for robots with an observer can guarantee the uniform ultimate bounds of the tracking errors and the observer errors as well as the bounds of the NN weights. For performance comparisons, the conventional adaptive algorithm with an observer using linearity in parameters of the robot dynamics is also developed in the same control framework as the NN approach for online approximating unknown nonlinearities of the robot dynamics. Main theoretical results for designing such an observer-based adaptive controller with the NN approach using multilayer NNs with sigmoidal activation functions, as well as with the conventional adaptive approach using linearity in parameters of the robot dynamics are given. The performance comparisons between the NN approach and the conventional adaptation approach with an observer is carried out to show the advantages of the proposed control approaches through simulation studies.

Index Terms—Adaptive control, neural networks (NNs), observer, robot, stability.

I. INTRODUCTION

ROBOTIC manipulators are complicated nonlinear dynamical systems with inherent unmodeled dynamics and unstructured uncertainties. These dynamical uncertainties make the controller design for manipulators a difficult task in the framework of classical adaptive and nonadaptive control. Design of ideal controllers for such systems is one of the most challenging tasks in control theory today, especially when manipulators are asked to move very quickly while maintaining good dynamic performance. Conventional control methods such as proportional, integral, and derivative (PID) scheme [1], the computed torque scheme (CTM) [2] and the adaptive control scheme (ACM) [3], [4], etc., have been in discussions for over twenty years. The traditional PID control with a simple structure and implementation has been the predominant method used for industrial manipulator controllers. Though the static precision is good if the gravitational torques are compensated, the dynamic performance of PID controllers leave much to be desired. CTM and ACM give very good performance, if manipulator dynamics are exactly known or the linearity in parameters of the robot dynamics holds. However, they suffer from three difficulties. First, they require explicit a priori knowledge of individual manipulators, which is very difficult to acquire in most practical applications. Second, uncertainties existing in real manipulators seriously devalue the performance of both methods. Although ACM has the ability to cope with structured uncertainties, it does not solve the problem of unstructured uncertainties. Third, the computational load of both methods is high. Since the control-sampling period must be at the millisecond level, this high computational load requires very powerful computing platforms that result in a high implementation cost.

A class of computational model known as neural networks (NNs) has been applied to robot control, which provides robotic manipulators with just such enhanced adaptive capability. Justification for using NNs for robot control lies in their excellent capability in learning any complicated mapping from training examples and generalizing what it has learned such that the robot controller is able to respond to an unexpected situation. Moreover, the parallel processing capability, when NNs have been implemented in hardware using very large scale integration (VLSI) technology, enables NNs to respond quickly in generating timely control actions.

Much research effort has been put into the design of NN applications for robot control. The early applications of NNs in the control of robotic manipulators include Albus and Miller’s CMAC Controller [5], [6], Iiguni’s linear optimal control techniques with backpropagation NNs [7], Kawato and Ozaki’s feedback compensators using backpropagation NNs [8], [9] for improving the control performance, etc. These NN-based control approaches could give good simulations or even experimental results. However, lack of theoretical analysis and stability security makes industrialists wary of using the results in real industrial environments. To cope with these problems, stable NN-based adaptive control both in continuous and discrete time for robots has been recently investigated by many researchers [10]–[16]. Representatives of these researches are nonlinarily parameterized NN-based adaptive controllers [10]–[12] and linearly parameterized NN-based adaptive ones [13]–[16] for robotic manipulators. In the proposed control schemes above, NNs are used to approximate the nonlinear components in the robot dynamic system, and Lyapunov stability theory or passive theory is employed to design a closed-loop control system with stability, convergence and improved robustness. As a result, the designed systems are stable, and online NN weight updating laws yield the function approximations. All these results have showed that stable NN-based
control approaches do have the potential to overcome the difficulties in robot control experienced by conventional adaptive and nonadaptive controllers [17]. However, most of the existing NN-based control approaches require the measurements of robot joint angle velocity, which may significantly deteriorate the control performance of these methods, because the velocity measurements are often contaminated by a considerable amount of noise. Furthermore, velocity sensors such as tachometers increase the weight and volume of the moving parts of the robot, thereby decreasing the robot’s efficiency. Therefore, it is desired to achieve good control performance by using only joint position measurements [18].

In order to solve the NN-based adaptive tracking control problem for those manipulators using the position measurements only, an NN-based output feedback controller with an observer is proposed by Kim [19] for rigid robotic manipulators, which contains two NNs, one for the observer and the other for the controller. The controller design requires accurate knowledge of the robot inertia matrix, and the controller structure and the computing algorithms are very complicated. In this paper, a novel hybrid control design is investigated by incorporating the merits of the NN-based adaptive control with the output feedback control of a robot. The output feedback control is used to stabilize the robot system with a linear observer, while the NN approach is employed to further improve the control performance of the controlled system by approximating the modified robot dynamics function. The whole NN-based controller design, with a linear observer to estimate the velocity of the robot, only requires one NN. At the same time, the robot dynamics is assumed to be unknown. This paper gives the main results for designing such an observer-based adaptive controller for robots using multilayer NNs with sigmoid activation functions. For performance comparison with the conventional adaptive control algorithm as on-line approximator, the adaptive control algorithm proposed by Bayard and Wen [20] is expanded with an observer in the same control framework as the NN approach for robot trajectory tracking. The effectiveness and efficiency of the proposed observer-based controller using multilayer NNs are demonstrated in comparison studies with the conventional adaptive control algorithm by simulations of a two-link manipulator.

This paper is organized as follows. In Section II, some basics for the robot model and its properties as well as those for controller design are reviewed. Then in Section III, main results for designing an NN-based adaptive controller and a conventional adaptive controller design are reviewed. Then in Section III, main results for designing such an observer-based adaptive controller with an observer is proposed by Kim [19] for rigid robotic manipulators, which contains two NNs, one for the observer and the other for the controller. The controller design requires accurate knowledge of the robot inertia matrix, and the controller structure and the computing algorithms are very complicated. In this paper, a novel hybrid control design is investigated by incorporating the merits of the NN-based adaptive control with the output feedback control of a robot. The output feedback control is used to stabilize the robot system with a linear observer, while the NN approach is employed to further improve the control performance of the controlled system by approximating the modified robot dynamics function. The whole NN-based controller design, with a linear observer to estimate the velocity of the robot, only requires one NN. At the same time, the robot dynamics is assumed to be unknown. This paper gives the main results for designing such an observer-based adaptive controller for robots using multilayer NNs with sigmoidal activation functions. For performance comparison with the conventional adaptive control algorithm as on-line approximator, the adaptive control algorithm proposed by Bayard and Wen [20] is expanded with an observer in the same control framework as the NN approach for robot trajectory tracking. The effectiveness and efficiency of the proposed observer-based controller using multilayer NNs are demonstrated in comparison studies with the conventional adaptive control algorithm by simulations of a two-link manipulator.

This paper is organized as follows. In Section II, some basics for the robot model and its properties as well as those for controller design are reviewed. Then in Section III, main results for designing an NN-based adaptive controller and a conventional adaptive controller with an observer for robot trajectory tracking are given, where a complete control structure and the learning algorithms for the free adaptive parameters are presented. Stability and tracking error convergence proof is also given in this section. An application example is given in Section IV. Finally, Section V concludes the paper by highlighting the feature properties of the proposed NN-based controller.

II. PRELIMINARIES

A. Notation

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

$$\|x\| = \sqrt{x^T x}, \quad \|A\| = \sqrt{\lambda_{\text{max}}(A^T A)}$$

with $\lambda_{\text{max}}(\cdot)$ the maximum eigenvalue. Moreover, for any positive definite symmetric matrix $A(x)$ and for any $x$, we denote the minimum and maximum eigenvalue of $A(x)$ by $\lambda_m$ and $\lambda_M$, respectively. Let $f(t) = (f_1, \ldots, f_n)^T$ be a vector function of time, define

$$\|f(t)\|_\infty = \text{ess sup}_{t \in \mathbb{R}} |f(t)|$$

where $| \cdot |$ denotes the norm in $R^n$. We say $f(t) \in L_\infty$ if $\text{ess sup}_{t \geq 0} |f(t)| < \infty$. $\text{sgn}(a)$ function is defined as follows:

$$\text{sgn}(a) = \begin{cases} 1 & \text{if } a > 0 \\ -1 & \text{if } a \leq 0. \end{cases}$$

Finally, we recall from [21], [22] the following definitions.

**Definition 1** [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 $t \geq t_0 + T$.

**Definition 2** [22]: Consider the same nonlinear system as described in Definition 1. If there exists a function $V : R_+ \times R^n \rightarrow R$, and constants $a, b, c, r > 0, p > 1$, such that

$$q |x|^p \leq V(t, x) \leq b |x|^p$$

$$\dot{V}(t, x) \leq -c |x|^p, \quad \forall t \geq 0, \forall x \in B_r.$$ (4)

Then the system is locally exponentially stable in space $B_r$, including the equilibrium $x = 0$.

B. Robot Dynamics and Its Properties

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

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

(5)

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 torques, $u(t) \in R^n$ the applied torque. $F(q, \dot{q})$ is the unstructured uncertainty of the dynamics including friction and other disturbances, and usually is assumed to be in a particular form

$$F(q, \dot{q}) = F_v \dot{q} + F_s(q)$$

(6)

where $F_v \dot{q}$ is the viscous friction, in which $F_v$ is a constant positive definite matrix defined by $F_v, M \leq ||F_v|| \leq F_v, M$, and $F_s(q) \in R^n$, the remaining part of the unstructured uncertainty, is assumed to be the continuous function of the robot.
joint angles. The following properties of the robot dynamics are required for the subsequent development.

**Property 1:** \( \mathbf{M}(q) \) is a positive symmetric matrix defined by \( M_m \leq ||\mathbf{M}(q)|| \leq M_M \) with \( M_M > 0 \) being known constants.

**Property 2:** \( \mathbf{C}(q, \dot{q}) \) defined by using the Christoffel symbols, satisfies that

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

**Property 3:** There exists a vector \( \mathbf{\theta} \in \mathbb{R}^{n_0} \) with components depending on robot parameters (masses, moments of inertia, etc.), such that

\[
\mathbf{M}(\mathbf{v}) \dot{\mathbf{v}} + \mathbf{C}(\mathbf{v}, \dot{\mathbf{v}}) \dot{\mathbf{v}} + \mathbf{G}(\mathbf{v}) = \Phi(\mathbf{v}, \dot{\mathbf{v}}, \ddot{\mathbf{v}}) \tag{7}
\]

where \( \Phi(\mathbf{v}, \dot{\mathbf{v}}, \ddot{\mathbf{v}}) \) is a vector of smooth functions, \( \Phi(\mathbf{v}, \dot{\mathbf{v}}, \ddot{\mathbf{v}}) \in \mathbb{R}^{n_0} \) is a coefficient matrix consisting of the known functions of joint position, velocity, and acceleration, which is called the regressor [3].

This property means that the dynamic equation can be linearized with respect to a specially selected set of robot parameters, which leads to the linear parameterization approach.

### C. Multilayer Feedforward NNs

Multilayer feedforward NNs are most commonly used in the NN-based controller design, which are composed of an input layer, an output layer, and at least one layer of nonlinear processing elements, which sum incoming signals and generate output signals according to some predefined function. An \( m \)-layer network with the same activation function \( \phi(\cdot) \) at each layer shown in Fig. 1, can be described by [23]

\[
\mathbf{y}(\mathbf{x}, \mathbf{W}) = \mathbf{W}_m^T \phi(\mathbf{W}_{m-1}^T \phi(\ldots \phi(\mathbf{x}))) \tag{8}
\]

where

- \( \mathbf{y} \in \mathbb{R}^p \); \( \mathbf{x} = (x_0, x_1, \ldots, x_2)^T \); \( \mathbf{W}_i \in \mathbb{R}^{(N_i+1) \times N_{i+1}} \); \( (i = 1, \ldots, m) \)
- \( \mathbf{W} \) is the output vector;
- \( \mathbf{x} \) is the input vector with \( x_0 = 1 \);
- \( \mathbf{W}_i \) is the weight matrix which include the threshold vector associated with the \( i \)th layer as its first column of \( \mathbf{W}_i^T \), where \( q = N_1, p = N_{m+1}, \mathbf{\phi}^T = (\gamma_0(\cdot), \gamma_1(\cdot), \ldots, \gamma_{N_i}(\cdot))^T \in \mathbb{R}^{N_i+1} \) is a nonlinear operator at the \( i \)th layer, \( \gamma_0(\cdot) = 1 \) is defined to allow one to include the threshold vector;
- activation functions, which are usually continuous, bounded, nondecreasing, nonlinear functions.

The usual choice is the sigmoidal function, defined as \( \gamma(a) = 1/(1 + e^{-a\xi}) \), where \( \xi \) is a constant.

For notational convenience, the vector of activation functions of the input layer is denoted as \( \mathbf{\phi}_1 = \phi(\mathbf{x}) \), then the vectors of hidden and the output layer activation functions are denoted by

\[
\mathbf{\phi}_{i+1} = \phi(\mathbf{W}_i^T \mathbf{\phi}_i), \quad i = 1, \ldots, m - 1 \tag{9}
\]

and the following fact holds for activation functions such as sigmoid, Tanh, RBF, etc.

\[
||\mathbf{\phi}_i|| \leq \phi_{i,\text{max}}, \quad i = 1, \ldots, m \tag{10}
\]

where \( \phi_{i,\text{max}}(i = 1, \ldots, m) \) are known positive values [23].

One of the most interesting properties of the NNs is that they are universal approximators, that is, they can approximate any real-valued continuous function or one with a countable number of discontinuities between two compact sets [24], [25]. Accordingly, we make the following assumption.

**A1:** Given a positive constant \( \varepsilon_N \) and a continuous function \( f : C \rightarrow \mathbb{R}^p \), where \( C \subset \mathbb{R}^q \) is a compact set, there exits a weight vector \( \mathbf{W} = \mathbf{W}^* \) such that the nonlinear function \( f(\mathbf{x}) \) can be approximated by the output \( \mathbf{y}(\mathbf{x}, \mathbf{W}) \) of the NN architecture (8) with \( m \)-layers

\[
f(\mathbf{x}) = \mathbf{y}(\mathbf{x}, \mathbf{W}^*) + \varepsilon \tag{11}
\]

where \( \varepsilon \in \mathbb{R}^p \) is the NN approximation error vector satisfying \( ||\varepsilon|| \leq \varepsilon_N \); the number of hidden layers in a multilayer NN may depend on \( \varepsilon_N \) and \( f(\mathbf{x}) \). The ideal weights \( \mathbf{W}^* \) are usually defined as those that minimize the supremum norm over \( C \) of \( \varepsilon \) [16], [23].

### III. OBSERVER-BASED CONTROLLER DESIGN USING NNs

#### A. Observer-Based Controller Design for Robots

To solve the tracking control problem for robots using position measurement only, Berghuis and Nijmeijer [26] consider the following controller–observer design based on passivity theory where the unstructured uncertainty \( \mathbf{F}(\mathbf{q}, \dot{\mathbf{q}}) \) defined in (6) is not considered for the time being.
Controller:
\[
\begin{align*}
\dot{q}_r &= \dot{q}_d - \Lambda_1 (q - q_d) \\
\ddot{q}_r &= \ddot{q}_d - \Lambda_2 (q - \dot{q})
\end{align*}
\]  
where \( K_d > 0, \Lambda_1 > 0, \Lambda_2 > 0 \) are assumed to be diagonal, \( L_d = \Lambda_2 I + \Lambda_2, L_D = \Lambda_2 \Lambda_2 \), and \( \lambda \) is the estimate of the robot joint angle, \( q_d \in \mathbb{R}^n \) is the desired path to be tracked, and it is assumed that \( V_M = \sup_t ||\dot{q}_d(t)||, \quad A_M = \sup_t ||\ddot{q}_d(t)||. \)  

Remark 1: \( \dot{q}_r \) and \( \ddot{q}_r \) are two auxiliary signals in the control law (12a). \( \dot{q}_r \) is usually called reference joint velocity in the standard adaptive control [3], while \( \ddot{q}_r \) is formed by modifying the estimated joint velocity \( \dot{q} \) using the observer position estimation error \( q - \dot{q} \) is introduced to guarantee the convergence of the observer errors. Intuitively, \( \ddot{q}_r \) decreases if the estimated joint angle \( \dot{q} \) lags behind the actual joint angle \( q \). For these, \( \ddot{q}_r \) is also called reference acceleration velocity of the robot joint angle. \( \Lambda_1 \) is usually chosen as \( \text{diag}(\Lambda_1, \ldots, \Lambda_n) \). If the only source of high-frequency unmodeled dynamics is assumed to be the finite sampling, it is shown by Slotine [27] that \( \lambda \) can be determined by \( \lambda \leq 0.25/T \), where \( T \) is the sampling period.

Remark 2: The observer with a similar structure as the pseudovelocity filter [28] consists of two dynamic equations shown in (12b). The auxiliary variable \( w(t) \in \mathbb{R}^n \) is introduced to make the equations implementable. \( \dot{w} \) denotes the reference acceleration input, which is formed by modifying the desired joint acceleration using the observer position estimation error. Integrating \( \dot{w} \) and further modifying it by observer position estimation error yield the estimated joint angle velocity. Such a simple linear observer has been used in other observer-based controller design for robots, and also verified by experiment [26], [29].

The following result is given by Berghuis and Nijmeijer [26].

Lemma 1: Consider the passivity-based output-feedback controller (12a) in a closed loop with a robotic manipulator (5). Define \( e = q - \dot{q} \), \( \ddot{e} = q - \ddot{q} \), \( x^T = [e^T, (\Lambda_1 e)^T, e^T, (\Lambda_2 e)^T] \). Under the conditions
\[
\begin{align*}
K_{d,m} > \Lambda_1, M M_M + \frac{5}{2} C_M V_M + 2 C_M \|x_0\| \frac{P_M}{P_m} \\
\Lambda_i > 0, M M_M + \frac{5}{2} C_M V_M + 2 C_M \|x_0\| \frac{P_M}{P_m} \\
\end{align*}
\]  
where \( x_0 = x(0) \) and \( P_m = \min \{ \frac{1}{5} M_m, \frac{5}{2} \Lambda_{1,m} \Lambda_1, M M_M \}, \quad P_M = \max \{ \lambda \Lambda_{1,m} K_{d,m}, \lambda \Lambda_{2,m} K_{d,m} \} \end{align*} \]  
the closed-loop system is locally exponentially stable.

If the unstructured uncertainty \( F(q, \dot{q}) \) is considered in the robot dynamics, then the controller is assumed to be in the following form:
\[
\begin{align*}
\dot{q}_r &= \dot{q}_d - \Lambda_1 (q - q_d) \\
\ddot{q}_r &= \ddot{q}_d - \Lambda_2 (q - \dot{q})
\end{align*}
\]  
and the observer is in the form of (12b). Define
\[
S_1 = \dot{q} - \dot{q}_r, \quad S_2 = \ddot{q} - \ddot{q}_r, \quad S = S_1 + S_2
\]

Then the following can be proven.

Theorem 1: Consider the output-feedback controller (17) in a closed loop with a robotic manipulator (5). Under the conditions
\[
\begin{align*}
K_{d,m} > \Lambda_1, M M_M + \frac{5}{2} C_M V_M + 2 C_M \|x_0\| \frac{P_M}{P_m} \\
\Lambda_i > 0, M M_M + \frac{5}{2} C_M V_M + 2 C_M \|x_0\| \frac{P_M}{P_m} \\
\end{align*}
\]  
then in the region of attraction
\[
B = \left\{ x \in \mathbb{R}^n \left| ||x|| < \frac{1}{2} \frac{P_M}{P_m} \right. \right\}
\]  
the closed-loop system is locally exponentially stable.

Proof: See Appendix.

In the controller design of robotic manipulators, one available technology is to use the desired joint angle values to take place of the actual joint angle values in the control law [8], [30]. This is important from the viewpoint of the universal approximation feature of NNs, since the desired joint angle values are normally bounded. Therefore, the following controller design is considered.

\[
\begin{align*}
\dot{q}_r &= \dot{q}_d - \Lambda_1 (q - q_d) \\
\ddot{q}_r &= \ddot{q}_d - \Lambda_2 (q - \dot{q})
\end{align*}
\]  
and the observer is in the form of (12b), and the following can be proved.
Theorem 2: Consider the output-feedback controller (21) in a closed loop with a robotic manipulator (5). Under the conditions
\begin{align*}
\begin{cases}
K_{d,m} > A_{1,1}MM + 6C_MM + 2M\|x_0\|\sqrt{\frac{F_m}{P_m}} \\
+ \frac{3}{2} F_{v,M} + \frac{3}{2} a_4 - F_{u,m} \\
L_d > 2M^{-1}K_{d,M}
\end{cases}
\end{align*}
(22)
where
\begin{align*}
a_4 &= (a_2A_M + (3/2)a_2V_M + a_3)A_{1,1}^{-1},
\end{align*}
and
\begin{align*}
a_1 &= \sup_{q \in R^n} \sum_{j=1}^n \left| \frac{\partial \text{Col}_j[M(q)]}{\partial q} \right|,
\end{align*}
\begin{align*}
a_2 &= \sup_{q \in R^n} \sum_{j=1}^n \sum_{j=1}^n \left| \frac{\partial \text{Col}_j[M_i(q)]}{\partial q} \right|,
\end{align*}
\begin{align*}
a_3 &= \sup_{q \in R^n} \left| \frac{\partial G(q)}{\partial q} \right| + \sup_{q \in R^n} \left| \frac{\partial F_{e}(q)}{\partial q} \right|,
\end{align*}
\begin{align*}
M_f(q) &= \frac{\partial M(q)}{\partial q_k}
\end{align*}
with \(q_i\) being the \(i\)th component of the vector \(q\). Then in the region of attraction
\begin{align*}
B = \left\{ x \in R^n \left| \|x\| < \frac{1}{2} \frac{\sqrt{P_m}}{P_M} \left[ \begin{array}{c}
K_{d,m} + F_{v,m} - A_{1,1}MM - \frac{3}{2} F_{v,M} - \frac{3}{2} a_4 \\
-6V_M
\end{array} \right] \right. \right\}
\end{align*}
(23)
the closed-loop system is locally exponentially stable.

Proof: See Appendix.

The controller given in (21) consists of a linear estimated state feedback part and a nonlinear part that is in a special form of full dynamics compensation. The controller–observer combination (21), (12b) is based on the requirement that exact knowledge of the robot dynamics is available. Obviously, this is a rather strong requirement that generally can not be met in practice. For robotic manipulators with partially known dynamics, even unknown dynamics, Berghuis and Nijmeijer have continued their research on the robust controller–observer design [29], [31]. The proposed robust controller with partially known robot dynamics is composed of the estimated robot dynamics compensation and a linear estimated state feedback control. If the robot dynamics is unknown, the controller will reduce to a linear estimated state feedback [29]. By using stability analysis techniques that are similar to the ones in [26], it is proved that the proposed controller with partially known or unknown robot dynamics can provide uniform ultimate bound of the closed-loop error dynamics for arbitrary initial condition \(x_0\) by increasing the gains \(K_d\) and \(L_d\) [29], [31].

Therefore, we use Theorem 2 to develop the observer-based adaptive controllers using linear parameterization of robot dynamics (property 3) and multilayer NNs given in Section II-C, respectively. Adaptive approaches here are used to approximate the following modified robot dynamics in the control law (21)
\begin{align*}
M(q_d)\ddot{q}_d + C(q_d, \dot{q}_d)\dot{q}_d + G(q_d) = \Phi(q_d, \dot{q}_d, \ddot{q}_d)\theta
\end{align*}
(24)

B. Observer-Based Controller Design Using Linear Parameterization Adaptation

With Section II-B, Property 3, in the linear parameterization adaptation of the robot dynamics enables us to have the following expression [see (7)]
\begin{align*}
M(q_d)\ddot{q}_d + C(q_d, \dot{q}_d)\dot{q}_d + G(q_d) = \Phi(q_d, \dot{q}_d, \ddot{q}_d)\theta
\end{align*}
(25)
where \(\theta \in R^{n_\alpha}\) is a constant unknown parameter vector from a suitable selected set of robot dynamic parameters, \(\Phi(q_d, \dot{q}_d, \ddot{q}_d) \in R^{n_\alpha \times n_\alpha}\) is the regressor matrix independent of the dynamic unknown parameters. The vector \(\theta\) is unknown, since the manipulator parameters are unknown. Therefore \(\hat{\theta}\) is used as the actual parameters. Define
\begin{align*}
S'_1 = \dot{\hat{q}} - \dot{q}_r, \quad S'_2 = \dot{\hat{q}} - \dot{q}_0 \quad \text{and} \quad S' = S'_1 + S'_2.
\end{align*}
(26)
If the modified robot dynamics in (24) is approximated by the linear parameterization of robot dynamics, the following theorem gives the stable adaptive control law and the parameter learning algorithm.

Theorem 3: Consider the robot dynamics defined in (5) with a control law
\begin{align*}
\mathbf{u}(t) &= -K_d(\ddot{\hat{q}}_d - \dot{q}_r) + \Phi(q_d, \dot{q}_d, \ddot{q}_d)\hat{\theta} - K_a \text{sgn}(S')
\end{align*}
(27)
and an adaptive law
\begin{align*}
\hat{\theta} = -\eta_a^{-1}(\Phi^T(q_d, \dot{q}_d, \ddot{q}_d)S + \sigma_a(\hat{\theta} - \theta_0))
\end{align*}
(28)
where
\begin{align*}
\eta_a &= \text{diag}(\eta_{a,1}, \ldots, \eta_{a,n_\alpha}) > 0 \quad \text{learning rate matrix;}
\sigma_a &> 0 \quad \text{and} \quad \theta_0 \quad \text{design constants;}
K_a &= \text{diag}(K_{a,1}, \ldots, K_{a,n_\alpha}) \quad \text{control gain matrix.}
\end{align*}
The observer is in the form of (12b). If \(K_d\) is a sufficiently large definite matrix, and \(L_d\) is a big enough positive constant, and
\begin{align*}
K_{a,i} \geq |F_i(q_d, \dot{q}_d)|, \quad i = 1, \ldots, n
\end{align*}
(29)
with \(F_i(q_d, \dot{q}_d)\) being the \(i\)th component of the vector \(F(q_d, \dot{q}_d)\). Then the closed-loop system is uniformly ultimately bounded.

Proof: See Appendix.

Remark 3: The control approach presented in Theorem 3 is the extension of the work by Bayard and Wen [20] to where a velocity observer is integrated in the conventional adaptive control loop. If no unstructured uncertainty is considered in the robot dynamics, the estimated joint angle values \(\hat{q}\) and \(\dot{\hat{q}}\) are replaced by actual ones \(\tilde{q}\) and \(\tilde{\dot{q}}\) in the control law (27), and let \(\sigma_a = 0\) in the learning algorithm (28). Then the adaptive control algorithm in Theorem 3 becomes the adaptive control algorithm 7a given in [20]. As such, the adaptive control algorithm 7a given in [20] is only a special case of the one presented in Theorem 3.
C. Observer-Based Controller Design Using Multilayer NNs

With A1, a multilayer NN $\mathbf{g}(\varphi, \mathbf{W})$ can approximate the modified dynamics function defined in (24), since the function is continuous with its bounded inputs. Then

$$M(q_d)\ddot{q}_d + C(q_d, \dot{q}_d)\dot{q}_d + G(q_d) + F_v\dot{q}_d + F_s(q_d) = \mathbf{g}(\varphi, \mathbf{W}) + \varepsilon(\varphi) \tag{30}$$

where $\mathbf{g}(\varphi, \mathbf{W}) \in R^m$ is defined in (8), $\varphi = (q_d^T, \dot{q}_d^T, \ddot{q}_d^T)^T$ is the corresponding input vector of the NN, and $\varepsilon(\varphi)$ is the approximation error vector satisfying $||\varepsilon(\varphi)|| \leq \varepsilon_m$, where $\varepsilon_m$ could be as small as possible by carefully choosing the NN structure and parameters. The NN weights are unknown, since the manipulator dynamics are unknown. Therefore, $\hat{\mathbf{W}}_i(i = 1, \ldots, m)$ are used as the actual NN weights. Then, the following can be proved.

Theorem 4: Consider the robot dynamics defined in (5) with a control law

$$u(t) = -K_d(\dot{q}_d - \hat{q}_d) + \hat{W}_m^T\phi_m - \varepsilon_m \text{sgn}(S') \tag{31}$$

and the following learning algorithms for the input and the hidden layers are

$$\hat{\dot{W}}_i = -\left(\hat{\phi}_i(\hat{\mathbf{W}}_i - \mathbf{W}_i^0)^T\phi_i - B_iS\right)^T + \sigma_i(\hat{\mathbf{W}}_i - \mathbf{W}_i^0)\eta_i \tag{32}$$

and for the output layer is

$$\hat{\dot{W}}_m = -\left(\hat{\phi}_mS^T + \sigma_m(\hat{\mathbf{W}}_m - \mathbf{W}_m^0)\right)\eta_m \tag{33}$$

where

- $\eta_i \in R^{N_{i+1} \times N_i+1}(i = 1, \ldots, m)$ learning rate matrix;
- $\sigma_i > 0$ and $\mathbf{W}_i^0 \in R^{(N_i+1) \times N_i+1}(i = 1, \ldots, m)$ design parameters;
- $\mathbf{F}_{i} = \text{tr}(\mathbf{W}_i\mathbf{W}_i^T)B_i \in R^{(N_{i+1}) \times m}$ satisfying $||\mathbf{F}_{i}|| \leq \kappa_i$;
- and $\varepsilon_m = \varepsilon_m + \Delta_m$.

The observer is in the form of (12b). If $\mathbf{K}_d$ is a sufficiently large definite matrix, and $L_d$ is a big enough positive constant, then the closed-loop system is uniformly ultimately bounded.

Proof: See Appendix.

A unified scheme diagram of the proposed controller is shown in Fig. 2. The NN controller (or adaptive controller) with $\varphi$ as an input vector, acts as a feedforward controller, which is used to approximate the modified robot dynamics function. In the feedforward control loop, there is a linear estimated state feedback control ($\hat{K}_d(\dot{\hat{q}}_d - \hat{\hat{q}}_d)$) and a sliding controller [32]. The sliding controller is added here to enhance the system robustness against unstructured uncertainties and the inherent NN approximation errors. The magnitude of the sliding control effort is the bound limit value on the NN approximation errors and the unstructured uncertainty.

In Theorem 4 (or Theorem 3), the design parameter $\mathbf{W}_i^0(i = 1, \ldots, m)$ (or $\theta_0$) can be considered as an initial estimate of the unknown weight $\mathbf{W}_i$ (or parameter $\theta$), allowing the designer to incorporate any prior parameter knowledge that may be available through off-line identification or other methods. As shown in (A.31) [or (A.23)], the closer $\mathbf{W}_i^0$ (or $\theta_0$) is to its true values, the smaller the residual tracking errors becomes. Besides, the weight learning laws (32) and (33) [or
incorporate a leakage term based on a variant of the \( \sigma \)-modification [33], which prevents parameter drift of the NN weights (adaptive parameters).

Remark 4: In the parameter learning laws (28), (32) and (33), \( S = 2\dot{q} - \dot{\hat{q}} - \dot{q} \), contains an unknown quantity \( \dot{q} \). By integrating both sides of (28), (32), and (33) over \([kT, (k+1)T]\), an equivalent version of (28), (32), and (33) are obtained, where \( \dot{q} \) is eliminated

\[
\hat{\theta}(k+1) \\
= \hat{\theta}(k) - \eta_\alpha \Phi^T(q_d(k), \dot{q}_d(k), \ddot{q}_d(k)) \\
\cdot \left( 2\Delta_q(k+1) - \Delta_q(k+1) - \Delta \dot{q}(k+1) \right) \\
- \eta_\alpha \int_{kT}^{(k+1)T} \left( \Phi(q_d, \dot{q}_d, \ddot{q}_d)(1+\Lambda_2 \dot{q}) + \sigma_\alpha (\hat{\theta} - \theta_0) \right) dt
\]

(34)

\[
\hat{W}_i(k+1) \\
= \hat{W}_i(k) + \hat{\theta}_i(k) \\
\cdot \left( 2\Delta_q(k+1) - \Delta_q(k+1) - \Delta \dot{q}(k+1) \right) \eta_k \\
+ \int_{kT}^{(k+1)T} \left( \hat{\theta}_i(L_1 e + \Lambda_2 \dot{q}) B_i^T + \eta_m \right) \cdot ( \dot{q}_i^{(m+1)} - \dot{q}_i^{(m)} ) dt \\
i = 1, \ldots, m
\]

(35)

\[
\hat{W}_m(k+1) \\
= \hat{W}_m(k) + \hat{\theta}_m(k) \\
\cdot \left( 2\Delta_q(k+1) - \Delta_q(k+1) - \Delta \dot{q}(k+1) \right) \eta_m \\
- \int_{kT}^{(k+1)T} \left( \hat{\theta}_m(L_1 e + \Lambda_2 \dot{q}) + \sigma_\alpha (\hat{W}_m - W_{m0}) \right) \eta_m dt
\]

(36)

where \( T \) is the sampling interval, and \( \Delta W_i(k+1) = W_i(k+1) - W_i(k) \). \( \Delta q(k+1) = q(k+1) - q(k) \). \( \Delta q_d(k+1) = q_d(k+1) - q_d(k) \). \( \Delta \dot{q}(k+1) = \dot{q}(k+1) - \dot{q}(k) \).

IV. SIMULATION RESULTS

In this section, the proposed observer-based adaptive control approach using multilayer NNs is used for the position control of a two-link manipulator with unknown dynamics, and its performance is illustrated as compared with the conventional adaptive control for robots with an observer given in Theorem 3. The dynamical equation and parameters of a two-link manipulator are the same as those in [26, Appendix] except that

\[
F(q, \dot{q}) = \begin{bmatrix}
\dot{q}_1 + 10 \cos(2q_1) \\
1.2 \dot{q}_2 + 5 \sin(2q_2)
\end{bmatrix}
\]

\[ |\dot{q}_i| \leq 360 \text{ (N.m), } \quad i = 1, 2. \quad (37) \]

The desired joint angle trajectories for a robot to track are

\[
q_d(t) = \cos(1.5t), \quad q_2d(t) = 3 \cos(3t). \quad (38)
\]

The controller–observer gains are chosen as

\[
K_d = \text{diag}(100, 50), \quad \Lambda_1 = \text{diag}(40, 40) \]

\[ \Lambda_2 = \text{diag}(0.1, 0.1), \quad \eta_\alpha = 400. \quad (39) \]

In the design of the adaptive tracking controller, a multilayer NN defined in (8) and a conventional adaptive approach based on the linear parameterization of robot dynamics in (7), are used to approximate the modified robot dynamics function, respectively. Since the desired joint acceleration vector \( \ddot{q}_d(t) \) is correlative to \( q_d(t) \) if the desired trajectories to be tracked are in the form of (38), the NN only requires vectors \( q_d(t), \ddot{q}_d(t) \) as its input vectors, and \( u(t) \) as its output vector. Simulations are done using a fourth-order Runge–Kutta algorithm with an integral step of 0.005 s, and the initial simulation condition is

\[
q_1(0) = 0.5, \quad \dot{q}_1(0) = 0, \quad q_2(0) = 2.0 \]

\[
\dot{q}_2(0) = 0, \quad \ddot{q}_1(0) = q_1(0), \quad \ddot{q}_2(0) = \ddot{q}(0). \quad (40)
\]

and the initial tracking errors of the robot joint state from the desired trajectories are

\[
\epsilon(0) = (-0.5, \quad -1.0)^T, \quad \epsilon(0) = (0.0, \quad 0.0)^T. \quad (41)
\]

In simulations, the design parameters of each controller are tuned to their best values, in terms of the conflicting requirements of tracking accuracy and controller stability, so that the best performances of these two types of controllers can be compared. In order to check the impact of the approximation power of these two different types of on-line approximators on the robot tracking performance, the sliding control components are all assumed to be 0, i.e., \( \mathbf{K}_a = \mathbf{0}, \epsilon_m = 0 \) in the following simulations.

A. Linearly Parameterized Adaptation as an On-line Approximator

With the robot dynamic equation given in the appendix of the reference [26], the equivalent parameter vector \( \theta \) can be written as

\[
\theta = (8.51, 1.26, 1.01, 8.1, 1.13)^T. \quad (42)
\]

Then the regressor matrix \( \Phi(q_d, \dot{q}_d, \ddot{q}_d) \) defined in (7) can be written as

\[
\Phi(q_d, \dot{q}_d, \ddot{q}_d) = \begin{bmatrix}
\dot{q}_1d & \dot{q}_2d & \dot{q}_1d + \dot{q}_2d & \alpha_3 \sin(q_1d) & \alpha_3 \sin(q_1d + q_2d) \\
0 & \dot{q}_1d + \dot{q}_2d & \alpha_3 & 0 & \alpha_3 \sin(q_1d + q_2d)
\end{bmatrix}
\]

(43)

with \( \alpha_1 = (2\dot{q}_1d + \dot{q}_2d) \cos(q_2d) - (\dot{q}_2d^2 + 2\dot{q}_1d \dot{q}_2d) \sin(q_2d) \), \( \alpha_2 = \dot{q}_1d \cos(q_2d) + \dot{q}_1d^2 \sin(q_2d) \).

The control algorithm (27) with parameter learning rule (28) is used to drive the robot joint angles to track the desired joint angle trajectories. The initial values of the parameter vector \( \hat{\theta} \) are taken to be 0, i.e., the parameters of the arm are assumed to be totally unknown. The adaptive controller therefore starts as a linear estimated state feedback controller and the nonlinear feedforward part constructed by parameter adaptation plays an increasingly effective role. The learning rates are chosen as \( \eta_\alpha = 0.85(1.0 - e^{-0.5t}) \) if \( i = 1, \ldots, 5 \), \( \eta_\alpha = 0.005 \), and \( \hat{\theta}_0 = 0, \mathbf{K}_a = 0 \).

Fig. 3(a) and (b) present the robot angle tracking errors during the first and the last 20 seconds of operation with \( F(q, \dot{q}) \) not
being considered and considered in the robot dynamics, respectively. Fig. 4(a) and (b) are the corresponding responses of the modified robot dynamics functions defined in (24), and outputs of the conventional adaptive algorithm using linear parameterization of robot dynamics defined in (7).

It is shown in Figs. 3(a)–4(b) that unstructured uncertainty devalues the approximation power of the conventional adaptive algorithm, the robot tracking performance deteriorates in such a case. It means that the conventional adaptive algorithm using linearity in parameters of robot dynamics could not deal with the unstructured uncertainty well in the robot dynamics.

B. Multilayer NNs as an Online Approximator

A multilayer NN with four neurons in the input layer, four neurons in the first hidden layer, three neurons in the second hidden layer, and two neurons in the output layer, is applied in the control law (31) for approximating the modified robot dynamics function. There are altogether 43 NN weights required to be determined. The activation function is chosen as $\gamma_i(\hat{a}) = 1/(1 + e^{-10\hat{a}})$, and the adaptive gain for the multilayer NN weight tuning are chosen as $\eta_3 = (1.0 - e^{-0.5t}) \text{diag}(2800, 2200)$, $\eta_2 = 1500(1.0 - e^{-0.5t}) \text{I}$, $\eta_1 = 20 \text{I}$, $\sigma_1 = \sigma_2 = 0.03$, $\sigma_3 = 3 \times 10^{-5}$, $\varepsilon_1 = 0$, $\text{B}_1 = 0.165(1.0 - e^{-0.5t}) \text{I}^{15 \times 2}$, $\text{B}_2 = 0.165(1.0 - e^{-0.5t}) \text{I}^{15 \times 2}$, where $\text{I}^{n_a \times n_b}$ denotes a $n_a \times n_b$ matrix with all the elements being one.

The same simulation parameters and initial conditions as in the previous case are chosen. Fig. 5(a) and (b) present the robot joint angle tracking errors during the first and second 20 s of operation with $\text{F}(\text{q}, \dot{\text{q}})$ not being considered and considered in the robot dynamics, respectively. Fig. 6(a) and (b) are the corresponding responses of the modified robot dynamics functions defined in (24), and multilayer NNs outputs defined in (8).

It is shown in Figs. 5(a)–6(b) that by online tuning laws given in (32) and (33), the multilayer NN provides a good approximation to the modified dynamics function. Its approximation power and tracking performance almost remain unchanged even with the unstructured uncertainty. Furthermore, the NN approach does not require the offline computation for determining the NN parameters, which is constructed by online learning, while the conventional adaptive algorithm requires the accurate offline computation of the regressor matrix in advance.
It is worth noting that the control performance of the linearly parameterized adaptive algorithm of robot dynamics can be improved by employing sliding control as shown in (27), if unstructured uncertainty exists in the robot dynamics. If the control gain matrix $K_0 = \text{diag}(15, 15)$ is chosen with boundary layer width 0.05 in Section III-B [27], the robot tracking performance shown in Fig. 3(b) can be improved. Fig. 7 shows the robot tracking error responses during the first and the last 20 s of operation.

Remark 5: In Sections IV-A and B, time-varying learning rates such as $\gamma_1$, $\gamma_2$, $\gamma_3$, etc., are chosen so as to improve the adaptation quality in the initial learning phrase. Since the derivative of the time-varying parameter $1 - e^{-0.2t}$ is not big and will approach zero as $t \to \infty$, the time-varying parameter will not influence the system stability if appropriate learning rates are chosen.

Remark 6: How to choose the NN structure for a prescribed bound on the NN approximation error is still a current topic of research. For our applications, an $m$-layer network with the same activation function $\phi(x)$ at each layer is chosen such that the work left for constructing the NN is only to determine the size of a hidden layer. The size of a hidden layer is usually determined experimentally. One experimental guideline is as follows. For a network of reasonable size, the size of hidden nodes needs to be only a relatively small fraction of the input layer. If the NN fails to converge to a solution, it is possible that more hidden
nodes are required. If it does converge, a few hidden nodes may be tried and then a size based on the overall system performance is settled.

Remark 7: The NN is simulated in Pentium PC-200 using the VC 6.0 language. It takes 1.4 ms to feedforward and feedback through the NN once, which is less than the sampling interval 5.0 ms. Hence, a real-time application of the proposed observer-based control scheme is possible by digital computers even without NN chips.

V. CONCLUSION

This paper presents an observer-based adaptive control scheme using multilayer NNs with only joint position measurements for the trajectory tracking of a robot with unknown dynamics nonlinearities. The main idea is the synthesis of the output feedback control with an observer and the NN-based adaptive control approach, where the output feedback control approach with an observer is used to control the robot system to move in the neighborhood of the desired path stably while the NN-based adaptive approach is used to further improve the system’s tracking performance by compensating for the modified robot dynamics nonlinearities as universal online approximators. Two different types of online approximators have been considered: 1) multilayer NNs using continuous, bounded, nondecreasing and nonlinear functions as activation units; 2) conventional adaptive algorithm using linear parameterization of robot dynamics. Although these two classes of on-line approximators are evidently constructed differently, they are examined in a common control framework for approximating the modified robot dynamics function.

This paper gives a unified control structure and the learning algorithms for the free adaptive parameters using these two classes of online approximators. The system stability and tracking error convergence are proved by Lyanapunov approach. The effectiveness and efficiency of the proposed observer-based controller using multilayer NNs are demonstrated in comparison studies with the conventional adaptive control algorithm by simulations of a two-link robot.

The proposed approach demonstrates important aspects when compared with related work in the fields of neural and conventional adaptive controllers for robots. In what follows we summarize the most significant advantages.

1) The proposed NN-based adaptive controller for robots only requires the joint position measurements. No offline computation of the NN parameters is required for robot trajectory tracking as compared with the conventional adaptive control algorithm by Bayard and Wen [20].

2) It is the first time in the NN literature for robot control, that a systematic approach is presented to deal with the trajectory tracking control for a robot with unknown dynamics nonlinearities using an observer. As compared with the existing work by Kim [19], the results given in this paper is simple, and suitable for any robotic manipulators with unknown dynamics nonlinearities.

3) The proposed control scheme excludes the assumption that is often used in the existing literature, i.e., the robot states are assumed to be within a compact set. Actually, without proving the stability of the whole system, the robot joint values may be unbounded. Therefore, the approximation equation is not necessarily true during online learning. By using the desired joint trajectory, velocity and acceleration to replace the actual ones, this problem is solved because desired joint signals are normally bounded without noise.

4) As compared with the conventional adaptive control using linear parameterization of robot dynamics. The NN-based control approach can tackle the unstructured uncertainties, and has a better approximation power and control performance than the conventional adaptive approach. Furthermore, it can solve the problem of high real-time computational requirements with NN chips, and is suitable for any manipulator.

5) The adaptive controller for robots with an observer given in Theorem 3 is a new result as the expansion of the adaptive control approach proposed by Bayard and Wen [20] to the case that a velocity observer is integrated in the conventional adaptive control loop. The control algorithm proposed in Theorem 3 can overcome the unstructured uncertainty in robot dynamics by augmenting a sliding control and only require the joint position measurements for the robot trajectory tracking. The adaptive control algorithm given in [20] is only a special case of the one proposed in Theorem 3 (also see Remark 3).

The above are achieved by the proposed adaptive controller for robotic manipulators with an observer. Investigations are necessary to further improve the performance of the proposed NN-based adaptive tracking controller.

APPENDIX

The Proof of Theorem 1

Refer to [26], the following Lyapunov function candidate is considered:

\[
V_0 = \frac{1}{2} S_1^T M(q) S_1 + \frac{1}{2} e^T \Lambda_1 (2 K_1 \Lambda_1^{-1} - M(q)) \Lambda_1 e \\
+ \frac{1}{2} S_2^T M(q) S_2 + \dot{q}^T K_d \dot{q}.
\]  \hfill (A.1)

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

\[
\dot{V}_0 = \frac{1}{2} S_1^T \dot{M}(q) S_1 + S_1^T M(q) \dot{S}_1 - \frac{1}{2} e^T \Lambda_1 \dot{M}(q) \Lambda_1 e \\
+ e^T \Lambda_1 (2 K_1 \Lambda_1^{-1} - M(q)) \Lambda_1 \dot{e} + \frac{1}{2} S_2^T \dot{M}(q) S_2 \\
+ S_2^T M(q) \dot{S}_2 + 2 \dot{q}^T K_d \dot{q}.
\]  \hfill (A.4)
Substituting (A.2) and (A.3) into (A.4), and using properties 1 and 2, it is easy to obtain

\[ \tilde{V}_0 = -\tilde{e}^T (K_d - \Lambda_1 M(q)) \tilde{e} - e^T \Lambda_1 K_d \Lambda e - \tilde{g}^T K_d \tilde{q} - \tilde{q}^T \Lambda_2 K_d \Lambda \tilde{e} - S_2^T (l_d M(q) - 2K_d) S_2 \\
- S_1^T C(q, S_2) \tilde{e} + e^T C(q, \tilde{q}) \Lambda e \\
+ S_2^T C(q, S_2 - \tilde{q}) \tilde{e} - (S_1 + S_2)^T F_v \tilde{e} \tag{A.5} \]

Since

\[ -(\tilde{e}^T (K_d - \Lambda_1 M(q)) \tilde{e} \leq -(K_d, m - \Lambda_1 M, M) ||\tilde{e}||^2 \\
- e^T \Lambda_1 K_d \Lambda e \leq K_d, m ||\Lambda e||^2 \\
- \tilde{q}^T K_d \tilde{q} \leq K_d, \tilde{m} ||\tilde{q}||^2 \\
- \tilde{q}^T \Lambda_2 K_d \Lambda \tilde{q} \leq K_d, \tilde{m} ||\Lambda \tilde{q}||^2 \\
- S_2^T (l_d M(q) - 2K_d) S_2 \leq -(l_d, M, m - 2K_d, m) ||S_2||^2 \tag{A.6} \]

Substituting (A.6)–(A.9) into (A.5) leads to

\[ \tilde{V}_0 \leq - (K_d, m - \Lambda_1 M, M) M - C_M \left( ||\Lambda e||^2 + ||\Lambda \tilde{q}||^2 + ||\Lambda \tilde{g}||^2 \right) \\
\cdot ||\tilde{e}||^2 - K_d, m ||\Lambda e||^2 - (K_d, m - C_M) ||\Lambda \tilde{q}||^2 \\
- (K_d, m - C_M) ||\Lambda \tilde{g}||^2 - (l_d, M, m - 2K_d, M) ||S_2||^2 \\
+ C_M V_M \left( ||\tilde{e}|| ||\tilde{q}|| + 2||\tilde{e}|| ||\Lambda \tilde{q}|| + ||\tilde{q}|| ||\Lambda e|| \right) \\
+ ||\Lambda e|| ||\Lambda \tilde{q}|| + ||\tilde{q}|| ||\Lambda \tilde{e}|| \right) \\
+ 2C_M ||\tilde{e}|| ||\tilde{q}|| ||\Lambda \tilde{q}|| - F_v, m ||\tilde{e}||^2 \\
+ F_v, M \left( ||\tilde{e}|| ||\Lambda e|| + ||\tilde{q}|| ||\tilde{q}|| + ||\tilde{e}|| ||\Lambda \tilde{e}|| \right) \tag{A.10} \]

Substituting (A.6)–(A.9) into (A.5) leads to

\[ \tilde{V}_0 \leq - (K_d, m - \Lambda_1 M, M) M - C_M \left( ||\Lambda e||^2 + ||\tilde{q}||^2 + ||\Lambda \tilde{g}||^2 \right) \\
\cdot ||\tilde{e}||^2 - K_d, m ||\Lambda e||^2 - (K_d, m - C_M) ||\tilde{q}||^2 \\
- (K_d, m - C_M) ||\Lambda \tilde{q}||^2 - (l_d, M, m - 2K_d, M) ||S_2||^2 \\
+ C_M V_M \left( ||\tilde{e}|| ||\tilde{q}|| + 2||\tilde{e}|| ||\Lambda \tilde{q}|| + ||\tilde{q}|| ||\Lambda e|| \right) \\
+ ||\Lambda e|| ||\Lambda \tilde{q}|| + ||\tilde{q}|| ||\Lambda \tilde{e}|| \right) \\
+ 2C_M ||\tilde{e}|| ||\tilde{q}|| ||\Lambda \tilde{q}|| - F_v, m ||\tilde{e}||^2 \\
+ F_v, M \left( ||\tilde{e}|| ||\Lambda e|| + ||\tilde{q}|| ||\tilde{q}|| + ||\tilde{e}|| ||\Lambda \tilde{e}|| \right) \tag{A.10} \]

Substituting (A.6)–(A.9) into (A.5) leads to

\[ \tilde{V}_0 \leq - (K_d, m - \Lambda_1 M, M) M - C_M \left( ||\Lambda e||^2 + ||\tilde{q}||^2 + ||\Lambda \tilde{g}||^2 \right) \\
\cdot ||\tilde{e}||^2 - K_d, m ||\Lambda e||^2 - (K_d, m - C_M) ||\tilde{q}||^2 \\
- (K_d, m - C_M) ||\Lambda \tilde{q}||^2 - (l_d, M, m - 2K_d, M) ||S_2||^2 \\
+ C_M V_M \left( ||\tilde{e}|| ||\tilde{q}|| + 2||\tilde{e}|| ||\Lambda \tilde{q}|| + ||\tilde{q}|| ||\Lambda e|| \right) \\
+ ||\Lambda e|| ||\Lambda \tilde{q}|| + ||\tilde{q}|| ||\Lambda \tilde{e}|| \right) \\
+ 2C_M ||\tilde{e}|| ||\tilde{q}|| ||\Lambda \tilde{q}|| - F_v, m ||\tilde{e}||^2 \\
+ F_v, M \left( ||\tilde{e}|| ||\Lambda e|| + ||\tilde{q}|| ||\tilde{q}|| + ||\tilde{e}|| ||\Lambda \tilde{e}|| \right) \tag{A.10} \]

By completing the square, we have

\[ \frac{2||\tilde{e}||}{2||\tilde{e}||^2 + ||\tilde{q}||^2} \leq ||\tilde{e}||^2 + ||\tilde{q}||^2 \\
\frac{2||\tilde{q}||}{2||\tilde{e}||^2 + ||\Lambda \tilde{q}||^2} \leq ||\tilde{e}||^2 + ||\Lambda \tilde{q}||^2 \\
\frac{2||\tilde{q}||}{2||\tilde{e}||^2 + ||\Lambda \tilde{q}||^2} \leq ||\tilde{e}||^2 + ||\Lambda \tilde{q}||^2 \\
\frac{2||\tilde{q}||}{2||\tilde{e}||^2 + ||\Lambda \tilde{q}||^2} \leq ||\tilde{q}||^2 \leq ||\Lambda \tilde{q}||^2 \leq \frac{1}{2} ||\Lambda e||^2 + \frac{1}{2} ||\Lambda \tilde{q}||^2 \\
\frac{2||\tilde{q}||}{2||\tilde{e}||^2 + ||\Lambda \tilde{q}||^2} \leq \frac{1}{2} ||\tilde{e}||^2 + \frac{1}{2} ||\Lambda \tilde{q}||^2 \leq \frac{1}{2} ||\tilde{e}||^2 + \frac{1}{2} ||\Lambda \tilde{q}||^2 \leq \frac{1}{2} ||\tilde{e}||^2 + \frac{1}{2} ||\Lambda \tilde{q}||^2 \leq \frac{1}{2} ||\tilde{e}||^2 + \frac{1}{2} ||\Lambda \tilde{q}||^2 \leq \frac{1}{2} ||\tilde{e}||^2 + \frac{1}{2} ||\Lambda \tilde{q}||^2 \leq \frac{1}{2} ||\tilde{e}||^2 + \frac{1}{2} ||\Lambda \tilde{q}||^2 \leq \frac{1}{2} ||\tilde{e}||^2 + \frac{1}{2} ||\Lambda \tilde{q}||^2 \leq \frac{1}{2} ||\tilde{e}||^2 + \frac{1}{2} ||\Lambda \tilde{q}||^2 \leq \frac{1}{2} ||\tilde{e}||^2 + \frac{1}{2} ||\Lambda \tilde{q}||^2 \leq \frac{1}{2} ||\tilde{e}||^2 + \frac{1}{2} ||\Lambda \tilde{q}||^2 \leq \frac{1}{2} ||\tilde{e}||^2 + \frac{1}
hold. Besides,
\[
\frac{1}{2} P_M \| \mathbf{x} \|^2 \leq V_0(\mathbf{x}) \frac{1}{2} P_M \| \mathbf{x} \|^2
\]  
(A.15)

From (A.12), (A.14) and (A.15) we obtain that if
\[
\| \mathbf{x}(0) \| < \frac{1}{2} \sqrt{\frac{P_M}{P_M}} \left( \frac{K_d m + F_{v,M} - A_1 M M M - \frac{3}{2} F_{v,M}}{C_M} - \frac{5}{2} V_m \right)
\]  
(A.16)

then
\[
V_0(\mathbf{x}) \leq V_0(\mathbf{x}(0)) \quad \forall t \geq 0
\]
\[
\dot{V}_0(\mathbf{x}) \leq -\chi \| \mathbf{x} \|^2 \quad \forall t \geq 0
\]

with $\chi$ a positive constant. By applying Definition 2 Theorem 1 is proved.

**Proof of Theorem 2**

Consider the same Lyapunov function as in Theorem 1. With control law (21) instead of (17), we have the following additional terms in $V_0$
\[
(S_1 + S_2)^T \{ M(q_a) - M(q) \} \dot{q}_a + (C(q, \dot{q}_a) \dot{q}_a - C(q, \dot{q}_a)) \dot{q}_d + G(q) - G(q) + F_s(q_a) - F_s(q) \} 
\]  
(A.17)

Refer to [30], $\mathbf{M}$, $\mathbf{C}$, $\mathbf{G}$ only contain trigonometric functions of $q$, hence the derivative of each element with respect to $\dot{q}$ is bounded. The additional terms in $\dot{V}_0$ can be overbounded by
\[
\begin{align*}
(\| S_1 \|^2 + \| S_2 \|^2) \left( (a_1 A_1 + \frac{3}{2} a_2 \sqrt{\lambda_2} + a_3) \| \dot{e} \| + C_M V_M (\| \ddot{q}_d - \dot{q}_d \|) \\
\leq (\| e \| + \| A_1 e \| + |\dot{e}| + |\lambda_2 \dot{e}|) \cdot (a_1 |\dot{A}_1 e| + C_M V_M (|\dot{e}| + |\dot{\dot{e}}| + |\lambda_2 \dot{e}|)) \\
\leq \frac{1}{2} (a_1 + 3 C_M V_M) (\| e \|^2 + |\dot{e}|^2 + |\lambda_2 \dot{e}|^2) \\
+ \frac{1}{2} (5 a_4 + 3 C_M V_M) |\dot{A}_1 e|^2
\end{align*}
\]  
(A.18)

With (A.18), we can write down $\dot{V}_0$ as
\[
\dot{V}_0 \leq -a_2 |\dot{e}|^2 - a_2 |\dot{A}_1 e|^2 - a_3 |\ddot{e}|^2 - a_4 |\lambda_2 \dot{e}|^2 \\
- (l_d M_m - 2 K_d m) \| S_2 \|^2
\]  
(A.19)

where $a_1 = a_1 - \frac{1}{2} C_M V_M (a_2 - \frac{1}{2} C_M V_M - a_3)$, $a_2 = a_2 - (3/2) C_M V_M (a_4 - 3/2 C_M V_M - (1/2) a_4)$, $a_3 = a_3 - (7/2) C_M V_M - (1/2) a_4$. Then following the same lines as that of the Theorem 1, Theorem 2 is proved under the conditions of (22).

**The Proof of Theorem 3**

Consider the Lyapunov function candidate
\[
V = V_0 + \frac{1}{2} \dot{\theta}^T \tau_k^{-1} \dot{\theta}
\]  
(A.20)

where $\dot{\theta} = \theta - \dot{\theta}$. With control law (27) instead of (21), we obtain the following additional terms in $V_0$
\[
S^T \left\{ A_2 \dot{q}_d, \dot{q}_d, \dot{q}_d, \dot{q}_d \right\} \dot{\theta} - F(q_d, \dot{q}_d) - K_a \text{sgn}(S') \right\}
\]  
(A.21)

and with adaptive law (28), we have
\[
\frac{1}{2} \dot{\theta}^T \tau_k^{-1} \dot{\theta} = -\frac{1}{2} \dot{\theta}^T \tau_k^{-1} \dot{\theta} \\
= S^T \left\{ A_2 \dot{q}_d, \dot{q}_d, \dot{q}_d, \dot{q}_d \right\} \dot{\theta} - \frac{1}{2} \dot{\theta} \left\| \dot{\theta} \right\|^2 \\
- \frac{1}{2} \dot{\theta} \left\| \dot{\theta} \right\|^2 + 2 \| K_a \|^2
\]  
(A.22)

With Theorem 2, (A.21) and (A.22), it is easy to obtain
\[
\dot{V} \leq -a_1 |\dot{e}|^2 - a_1 |A_1 e|^2 - a_3 |\ddot{e}|^2 - a_4 |\lambda_2 \dot{e}|^2 \\
- S^T (F(q_d, \dot{q}_d) + K_a \text{sgn}(S')) \\
- \frac{1}{2} \dot{\theta} \left\| \dot{\theta} \right\|^2 + 2 \| \dot{\theta} \|^2 + 2 \| K_a \|^2
\]  
(A.23)

where $s_i'$ is the $i$th component of $S'$, and $\overline{P}_M = \max \{ 6 A_1 \eta_m K_d M, 6 A_2 \eta m K_d M, \eta_{\text{min}} \}$ with $\eta_{\text{min}}$ being the minimum eigenvalue of $\eta_k$, and $\lambda_a = 2 \min \{ a_1, a_2, a_3 - 2, a_4 - (1/2) \sigma_a \}$
\[
\gamma_a = -\frac{1}{2} \sigma_a \left\| \dot{\theta} - \dot{\theta} \right\|^2 + \frac{1}{2} \sigma_a \left\| \dot{\theta} - \dot{\theta} \right\|^2 + 2 \| K_a \|^2.
\]  
It is concluded that $|\dot{e}|$, $|\ddot{e}|$ and $|\lambda_2 \dot{e}|$ will eventually fall into a residual set with the size $O(\gamma_a)$, and so will $|\dot{e}|$, $|\ddot{e}|$, $|\lambda_2 \dot{e}|$. By applying Definition 1 Theorem 3 is proved.

**The Proof of Theorem 4**

Consider the Lyapunov function candidate
\[
V = V_0 + \frac{1}{2} \sum_{i=1}^m \operatorname{tr} \left( W_i \tau_k^{-1} \tilde{W}_i^T \right)
\]  
(A.24)
where $\hat{W}_i = W_i - \hat{W}_i$. With control law (31) instead of (21), we obtain the following additional terms in $\dot{V}_0$

$$S^T \left\{ \hat{W}_m^T \hat{\phi}_m - W_m^T \phi_m - \varepsilon(\varphi) - \varepsilon_m \text{sgn}(S') \right\} = S^T \left\{ -W_m^T \hat{\phi}_m - W_m^T \phi_m - \varepsilon(\varphi) - \varepsilon_m \text{sgn}(S') \right\}$$

(A.25)

where $\hat{\phi}_m = \phi_m - \hat{\phi}_m$. With weight tuning laws (32) and (33), we have

$$\text{tr} \left( \dot{W}_m n_m^{-1} \dot{W}_m^T \right) = -\text{tr} \left( \hat{W}_m n_m^{-1} \hat{W}_m^T \right)$$

$$\leq - \| \hat{W}_m^T \hat{\phi}_i - \frac{1}{2} (W_i - W_i^0)^T \hat{\phi}_i - B_i S \| ^2$$

$$+ \frac{1}{2} \| W_i - W_i^0 \| ^2 \phi_i^2_{\max} + \frac{1}{2} \gamma \| S \| ^2 - \frac{1}{2} \gamma \| \dot{W}_i \| ^2$$

$$- \frac{1}{2} \gamma \| W_i - W_i^0 \| ^2 + \frac{1}{2} \gamma \| \dot{W}_i - W_i^0 \| ^2$$

(A.26)

$$\text{tr} \left( \hat{W}_m n_m^{-1} \hat{W}_m^T \right) = -\text{tr} \left( \hat{W}_m n_m^{-1} \hat{W}_m^T \right)$$

$$\leq \| \hat{W}_m^T \hat{\phi}_m - \frac{1}{2} \sigma_m \| W_m \| ^2 - \frac{1}{2} \sigma_m \| W_m - W_0 \| ^2$$

$$+ \frac{1}{2} \sigma_m \| W_m - W_0 \| ^2$$

(A.27)

With Theorem 2, and (A.25)–(A.27), it is easy to obtain

$$\dot{V} \leq -\alpha_1 \| \hat{\varphi} \|^2 - \alpha_2 \| \Delta_1 \varphi \|^2 - \alpha_3 \| \Delta_2 \varphi \|^2 + S^T \left\{ \hat{W}_m^T \hat{\phi}_m - W_m^T \phi_m - \varepsilon(\varphi) - \varepsilon_m \text{sgn}(S') \right\}$$

$$+ S^T \left\{ -W_m^T \hat{\phi}_m - W_m^T \phi_m - \varepsilon(\varphi) - \varepsilon_m \text{sgn}(S') \right\}$$

$$+ S^T \left\{ -W_m^T \hat{\phi}_m - W_m^T \phi_m - \varepsilon(\varphi) - \varepsilon_m \text{sgn}(S') \right\}$$

$$+ S^T \left\{ -W_m^T \hat{\phi}_m - W_m^T \phi_m - \varepsilon(\varphi) - \varepsilon_m \text{sgn}(S') \right\}$$

$$+ \frac{1}{2} \| \dot{W}_i - W_i^0 \| ^2 \phi_i^2_{\max} + \frac{1}{2} \sum_{i=1}^{m-1} \sigma_i \| \dot{W}_i \| ^2$$

(A.28)

Since

$$\| S \| ^2 = \| S_1 + S_2 \| ^2 \leq 4 \left( \| \hat{\varphi} \| ^2 + \| \Delta_2 \varphi \| ^2 + \| \dot{\varphi} \| ^2 + \| \Delta_1 \varphi \| ^2 \right)$$

(A.29)

Then (A.28) can be written as

$$\dot{V} \leq -\alpha_1 \| \hat{\varphi} \|^2 - \alpha_2 \| \Delta_1 \varphi \|^2 - \alpha_3 \| \Delta_2 \varphi \|^2 + S^T \left\{ \hat{W}_m^T \hat{\phi}_m - W_m^T \phi_m - \varepsilon(\varphi) - \varepsilon_m \text{sgn}(S') \right\}$$

$$+ S^T \left\{ -W_m^T \hat{\phi}_m - W_m^T \phi_m - \varepsilon(\varphi) - \varepsilon_m \text{sgn}(S') \right\}$$

$$+ S^T \left\{ -W_m^T \hat{\phi}_m - W_m^T \phi_m - \varepsilon(\varphi) - \varepsilon_m \text{sgn}(S') \right\}$$

$$+ \frac{1}{2} \| \dot{W}_i - W_i^0 \| ^2 \phi_i^2_{\max} + \frac{1}{2} \sum_{i=1}^{m-1} \sigma_i \| \dot{W}_i \| ^2$$

(A.30)

Theorem 2, and (A.25)–(A.27), it is easy to obtain

$$\dot{V} \leq -\alpha_1 \| \hat{\varphi} \|^2 - \alpha_2 \| \Delta_1 \varphi \|^2 - \alpha_3 \| \Delta_2 \varphi \|^2 + S^T \left\{ \hat{W}_m^T \hat{\phi}_m - W_m^T \phi_m - \varepsilon(\varphi) - \varepsilon_m \text{sgn}(S') \right\}$$

$$+ S^T \left\{ -W_m^T \hat{\phi}_m - W_m^T \phi_m - \varepsilon(\varphi) - \varepsilon_m \text{sgn}(S') \right\}$$

$$+ S^T \left\{ -W_m^T \hat{\phi}_m - W_m^T \phi_m - \varepsilon(\varphi) - \varepsilon_m \text{sgn}(S') \right\}$$

$$+ \frac{1}{2} \| \dot{W}_i - W_i^0 \| ^2 \phi_i^2_{\max} + \frac{1}{2} \sum_{i=1}^{m-1} \sigma_i \| \dot{W}_i \| ^2$$

(A.31)

where

$$\gamma = \frac{1}{2} \sum_{i=1}^{m-1} \sigma_i \| W_i - W_i^0 \| ^2$$

$$+ \frac{1}{2} \sum_{i=1}^{m-1} \| W_i - W_i^0 \| ^2 \phi_{\max}^2 + \sqrt{m} \varepsilon_m^2$$

It is concluded that $\| \hat{\varphi} \|$, $\| \dot{\varphi} \|$, and $\| \dot{W}_i \|$ will eventually fall into a residual set with the size $O(\gamma)$, and so will $\| \varphi \|$ and $\| \dot{\varphi} \|$. By applying Definition 1 Theorem 4 is proved.

**REFERENCES**


