Adaptive Control of Robotic Manipulators with Unified Motion Constraints

Mingming Li, Yanan Li, Member, IEEE Shuzhi Sam Ge, Fellow, IEEE Tong Heng Lee, Senior Member, IEEE

Abstract—In this paper, we present an adaptive control of robotic manipulators with parametric uncertainties and motion constraints. Position and velocity constraints are considered and they are unified and converted into the constraint of the nominal input. An adaptive neural network control is developed to achieve trajectory tracking, while the problems of motion constraints are addressed by considering the saturation effect of the nominal input. The uniform boundedness of all closed-loop signals is verified through Lyapunov analysis. Simulation and experiment results on a 2 DOF robotic manipulator demonstrate the effectiveness of the proposed method.

Index Terms—robotic manipulator; adaptive control; unified motion constraints; input saturation; neural network approximation

I. INTRODUCTION

In robotic applications with unstructured environments, especially those related to human-robot interaction [1], [2], [3], [4], [5], robots must be subject to certain motion constraints, which typically include position and velocity constraints. In the example of teleoperated surgical robots, they must be maneuvered in constrained task spaces, often through a narrow entry portal into the patient’s body [6]. In another example of human-robot collaboration, a robot shares a common workspace with a human, so its velocity must be constrained within a predefined bound to guarantee the safety of both the robot and the human. An impact test is conducted in [7] to evaluate endangerment to human beings caused by collision with robots moving at different velocities. Results have shown that the head injury criterion (HIC), which is used to quantify the injury level of human beings, significantly increases as the impact velocity of robots becomes larger.

In the literature, research effort has been made on control of robotic manipulators with motion constraints. A bounded control for set-point regulation problem of robotic manipulators with joint velocity constraints is proposed in [8]. In [9], a trajectory tracking control of robotic manipulators is achieved on a surface with joint velocity constraints. A unified quadratic programming formulation based dynamical system approach is proposed in [10] to solve the joint torque optimization problem of physically constrained redundant manipulators, but it requires complete knowledge of the system dynamics. In [11], [12], adaptive control of robotic manipulators with position (output) constraints is developed by employing the barrier Lyapunov function (BLF), in the presence of parametric uncertainties and external disturbances. In [13], optimal control of robotic manipulators with joint position constraints is achieved using adaptive dynamic programming. Researches of task space region control are proposed in [14], [15], [16] and adaptive control of robot subject to position constraints is achieved in [17] by defining the objective functions in the form of a set of inequalities. However, only position constraints are considered in the designed region control of [17] as these objective functions (the inequalities) are constructed with respect to only the robot’s task space positions.

Alternatively, since a robotic manipulator can be considered as a nonlinear system subject to some physical properties [18], nonlinear control with consideration of state constraints may be applied [19], [20], [21]. Neural network based predictive control is proposed in [22], [23] with system identification implemented using historical data and off-line training while predictive control with online estimation proposed in [24], [25], [26] is only applicable to single-input-single-output (SISO) system. Recently, a robust adaptive neural tracking control with integral BLF [27] is proposed in [28] with integral BLFs for a class of SISO strict-feedback nonlinear systems under state constraints while in [29], adaptive control subject to full state constraints is achieved for a more general SISO pure-feedback nonlinear system, with simulation results on a single-link robot. To tackle the uncertainties in multi-input-multi-output (MIMO) systems like robots, neural networks [30], [31] or fuzzy logic systems (FLS) [32], [33], [34] are employed as online model-free approximators. In [35], an adaptive fault tolerant control is derived for a class of input and state constrained MIMO nonlinear systems, where the state constraints are formulated as a constraint of the state vector’s norm. Therefore, it implies that the state constraints have to be symmetric. On the other hand, in [36], [37], prescribed performance adaptive neural network control is developed to constrain the error signals within a prescribed region, which is shaped by a performance function. This method may be extended to handle the problem of motion constraints, but it is not straightforward to define the corresponding performance function which is related to the tracking error and is required to be exponentially decaying.

In this paper, an adaptive neural network control is proposed for robotic manipulators with unknown dynamics and motion constraints of position and velocity. This method is different from previous methods since it is able to directly handle various types of motion constraints that are independently and explicitly defined. Specifically, a unified framework is established to convert motion constraints into an input saturation...
problem with uncertainties incorporated in both the nominal control input and saturation law, which is different from the input saturation problem in [38]. By using the GL matrix and operator [39], an adaptive neural network approximator is designed to address the issue of unknown system dynamics. By Lyapunov analysis, the boundedness of all closed-loop signals is shown to be guaranteed while the motion constraints are not violated. Simulations and experiments are conducted to illustrate the efficacy of the proposed method. The remainder of this paper is organized as follows. In Section II, the model of a robotic manipulator with motion constraints is presented, a general method is introduced to convert motion constraints into the saturation of the control input, and some useful preliminaries are given. In Section III, the adaptive control design is detailed with rigorous Lyapunov stability analysis. Simulation and experiment results are presented in Sections IV and V to demonstrate the effectiveness of the proposed method, followed by some concluding remarks in Section VI.

II. PROBLEM FORMULATION

A. Kinematics and Dynamics

The dynamics of an \( n \) degree-of-freedom (DOF) robotic manipulator can be described as a MIMO nonlinear system as follows [18]:

\[
M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau
\]

where \( q \in \mathbb{R}^n \) is the joint position, \( M(q) \in \mathbb{R}^{n \times n} \) is the inertia matrix, \( C(q, \dot{q}) \dot{q} \in \mathbb{R}^n \) is the Coriolis and centrifugal force, \( G(q) \in \mathbb{R}^n \) is the gravitational force, and \( \tau \in \mathbb{R}^n \) is the input joint torque. The joint space dynamics (1) are transformed into the task space dynamics

\[
M_\eta(\eta)\ddot{\eta} + C_\eta(\eta, \dot{\eta})\dot{\eta} + G_\eta(\eta) = \tau
\]
via the forward kinematics

\[
\eta = \Omega(q), \quad \dot{\eta} = \frac{\partial \Omega}{\partial q} \dot{q} =: J(q)\dot{q}
\]

where \( \eta = [\eta_1, \eta_2, \cdots, \eta_n]^T \) is a vector of task variables, \( M_\eta \) is the inertia matrix, \( C_\eta \) is the Coriolis and centrifugal force, \( G_\eta \) is the gravitational force, and \( \tau \) is the control input in the task space given as

\[
M_\eta = J^T M J^{-1}, \quad C_\eta = J^T (C - M J^{-1} \dot{J}) J^{-1}, \quad G_\eta = J^T \tau
\]

For simplicity, only the non-redundant \((m = n)\) non-singular manipulators are considered in this paper. Let \( x_1(t) = \eta, x_2(t) = \dot{\eta} \), we have the dynamics of the robotic manipulator in the state-space form, as below

\[
\begin{cases}
\dot{x}_1 = x_2, \\
\dot{x}_2 = M_\eta^{-1}(x_1)(u - C_\eta(x_1, x_2)x_2 - G_\eta(x_1)), \\
y = x_1
\end{cases}
\]

Note that \( M_\eta^{-1}(x_1), C_\eta(x_1, x_2) \) and \( G_\eta(x_1) \) are all unknown matrices. The control objective is to make the position of the end-effector \( y(t) \) track the desired trajectory \( y_d(t) \). In addition, it is expected to keep all closed-loop signals bounded and prevent the violation of motion constraints, which will be introduced in the following.

Assumption 1: [40] For arbitrary \( t > 0 \), there exist \( b_1 > 0 \) and \( b_2 > 0 \) such that \( \|\dot{y}_d(t)\| \leq b_1 \) and \( \|\ddot{y}_d(t)\| \leq b_2 \).

B. Motion Constraints

A motion constraint of the end-effector is a kinematic/dynamic constraint related to the variable that describes the end-effector’s motion status. In this paper, we consider two typical motion constraints, i.e., position and velocity constraints.

For the robotic manipulator (5), let \( y_1, y_2, x_1, x_2 \) denote the \( i \)-th element of the vectors \( y, y_d, x_1, x_2 \) for \( i = 1, \cdots, n \), respectively. Motion constraints are defined as

\[
p_i^- \leq y_i - p_i^+ \leq p_i^+ \quad \text{and} \quad v_i^- \leq \dot{y}_i \leq v_i^+
\]
for \( i = 1, 2, \cdots, n \), where \( p_i^-, v_i^-, p_i^+, v_i^+ \) are known constant or time-varying limits.

Remark 1: Note that the constraints investigated in this paper are given in the form of two-sided inequalities. Compared with those considered in [28] and [35], which are described as bounds of the absolute value/norm of a state variable, respectively, the constraints formulated in (6) are asymmetric and they can be assigned independently as the upper and lower bounds of the state variable. Therefore, it can account for more general task space constraints in practice and offer greater flexibility during control design.

Assumption 2: The initial position and velocity of the end-effector satisfy \( p_i^- \leq y_i(0) \leq p_i^+ \) and \( v_i^- \leq \dot{y}_i(0) \leq v_i^+ \) for \( i = 1, 2, \cdots, n \).

Based on the above assumption, the two-sided inequalities of the motion constraints are converted following the procedures below. Choose positive scalars \( k_{1i}, k_{2i}, k_{3i} \), and consider the scalar \( \dot{y}_i \) regulated as

\[
-k_{1i}(y_i - p_i^-) \leq \dot{y}_i \leq -k_{1i}(y_i - p_i^+)
\]

Considering the following inequalities

\[
\dot{y}_i \geq -k_{1i}(y_i - p_i^-) \geq 0 \quad \text{if} \quad y_i \leq p_i^-, \quad \text{and}
\]

\[
\dot{y}_i \leq -k_{1i}(y_i - p_i^+) \leq 0 \quad \text{if} \quad y_i \geq p_i^+
\]
we find that \( \dot{y}_i \) cannot transgress the constraints when \( \dot{y}_{i,\text{min}} = -k_{1i}(y_i - p_i^-) \) and \( \dot{y}_{i,\text{max}} = -k_{1i}(y_i - p_i^+) \). Similarly, we can guarantee \( y_i \in [y_{i,\text{min}}, y_{i,\text{max}}] \) if \( \dot{y}_i \) satisfies

\[
-k_{2i}(y_i - y_{i,\text{min}}) \leq \dot{y}_i \leq -k_{2i}(y_i - y_{i,\text{max}})
\]

Thus, according to (5), we have

\[
h_{pi}^- \leq \dot{x}_{2i} \leq h_{pi}^+
\]

where \( h_{pi}^- = -k_{2i}(x_{2i} + k_{1i}(x_{1i} - p_i^-)) \) and \( h_{pi}^+ = -k_{2i}(x_{2i} + k_{1i}(x_{1i} - p_i^+)) \).

Consider the velocity constraint of the end-effector defined as \( v_i^- \leq \dot{x}_{2i} \leq v_i^+ \). Similarly to the process of converting the position constraint, we have

\[
-k_{3i}(x_{2i} - v_i^-) \leq \dot{x}_{2i} \leq -k_{3i}(x_{2i} - v_i^+)
\]
Thus, $x_{2i}$ is constrained as
\[ h_{v_i}^- \leq x_{2i} \leq h_{v_i}^+ \]  
(12)
where $h_{v_i}^- = -k_3(x_{2i} - v_i^-)$ and $h_{v_i}^+ = -k_3(x_{2i} - v_i^+)$. Thus, $x_{2i}$ will not violate the velocity constraint $v_i^- \leq x_{2i} \leq v_i^+$ if $y_i(0) = x_{2i}(0) \in [v_i^-, v_i^+]$.

As both (10) and (12) are with respect to $\dot{x}_2$, they can be combined as a unified constraint as below
\[ h_i^- \leq \dot{x}_2 \leq h_i^+ \]  
(13)
where $h_i^- = \max\{h_{v_i}^-, h_{P_i}^\tau\}$ and $h_i^+ = \min\{h_{P_i}^+, h_{v_i}^\tau\}$.

Remark 2: In this way, the position and velocity constraints have been converted into a unified one as (13). In fact, it is clear that one can even incorporate the robot’s acceleration constraints $a_i \leq \dot{x}_{2i} \leq a_i^+$ into (13) if necessary, which requires no change in control design as all motion constraints are compactly represented as (13) and our control is derived solely based on it. In addition, such a conversion process can also be applied to constraints in the joint space when the robotic manipulator is subject to different types of joint limits and a joint space controller is to be designed. Comparatively, one may extend region control with the unified objective bound in [17] to handle velocity or even acceleration constraints by adding the corresponding inequalities of the task space velocities and accelerations into the objective functions. However, considering these additional constraints with the method in [17] will require the signals $\ddot{q}$ and $\dddot{q}$ in the resulted controller as it is derived by taking the time-derivatives of the objective functions. Considering the fact that the usage of $\dddot{q}$ or even $\dddot{q}$ in control design is generally not desired due to the difficulties in obtaining their accurate measurements [18], the practicability of this region control method for motion constraints investigated in this paper may be limited.

C. Preliminaries: Neural Network

It has been shown that the radial basis function neural network (RBFNN) can approximate an arbitrary continuous function $a(Z)$ over a compact set $\Omega \subset \mathbb{R}^{n_Z}$ to any accuracy [41] as below
\[ a(Z) = w^T \phi(Z) + e_Z, \forall Z \in \Omega \]  
(14)
where $Z \in \mathbb{R}^{n_Z}$ is the input vector, $w^* \in \mathbb{R}^{n_w}$ are the optimal constant weights with $n_w$ being the number of nodes, $e_Z \in \mathbb{R}$ is the functional approximation error, and $\phi(Z) = [\phi^{(1)}(Z), \phi^{(2)}(Z), \ldots, \phi^{(n)}(Z)] \in \mathbb{R}^{n_w}$ are vectors of Gaussian functions as below
\[ \phi^{(i)}(Z) = \exp\left(-\frac{(Z - \mu^{(i)})^T (Z - \mu^{(i)})}{\sigma^{(i)}_2}\right) \]  
(15)
with $\mu^{(i)}$ being the center of the Gaussian function and $\sigma^{(i)}$ being the variance.

According to [42], there exist optimal weights $w^*$ such that $|e_Z| \leq e^*_Z$ with $e^*_Z \geq 0$, which can be made arbitrary small provided that $n_w$ is sufficiently large. Thus, the optimal weights $w^*$ are defined such that $e_Z$ is minimized for all $Z \in \Omega_Z$, i.e.,
\[ w^* := \arg \min_{w \in \mathbb{R}^{n_w}} \left\{ \sup_{Z \in \Omega_Z} |a(Z) - w^T \phi(Z)| \right\}. \]  
(16)

Then, an approximation of $a(Z)$ can be constructed as
\[ \hat{a}(Z) = \hat{w}^T \phi(Z) \]  
(17)
where $\hat{a}(Z)$ is the approximation of $a(Z)$ and $\hat{w} \in \mathbb{R}^{n_w}$ are the estimates of the corresponding optimal weights $w^*$ defined in (16).

By employing RBFNNs to approximate each of its element, a matrix function $A(Z) \in \mathbb{R}^{n_1 \times n_2}$ is approximated as $\hat{A}(Z)$. Following the convention of GL matrices and GL operator for vectors and matrices [39], the expression of $\hat{A}(Z)$ is given as
\[ \hat{A}(Z) = \left[ \begin{array}{cccc} \hat{a}_{11}(Z) & \hat{a}_{12}(Z) & \cdots & \hat{a}_{1n_2}(Z) \\ \hat{a}_{21}(Z) & \hat{a}_{22}(Z) & \cdots & \hat{a}_{2n_2}(Z) \\ \vdots & \vdots & \ddots & \vdots \\ \hat{a}_{n_11}(Z) & \hat{a}_{n_12}(Z) & \cdots & \hat{a}_{n_1n_2}(Z) \end{array} \right] = [W]^T \{\Phi\} = \left[ \begin{array}{cccc} \hat{w}_1^T \phi_{11} & \hat{w}_1^T \phi_{12} & \cdots & \hat{w}_1^T \phi_{1n_2} \\ \hat{w}_2^T \phi_{21} & \hat{w}_2^T \phi_{22} & \cdots & \hat{w}_2^T \phi_{2n_2} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{w}_n^T \phi_{n_11} & \hat{w}_n^T \phi_{n_12} & \cdots & \hat{w}_n^T \phi_{n_1n_2} \end{array} \right] \]
where $\{W\}, \{\Phi\}$ are the GL matrices and $\bullet$ is the GL operator.

Remark 3: In this paper, RBF neural networks are employed due to its capabilities of approximating any unstructured smooth nonlinear functions to arbitrary accuracy over a compact set. In fact, one can use other online approximation method instead, such as fuzzy logic systems [43], [44], [45], [32]. The online approximator can be further reduced to a regressor function by assuming that the uncertainties are structured and are linear in parameters [46], which requires a set of model-specific basis functions.

III. CONTROL DESIGN

To incorporate the constraint (13) into the actual control input $u$, the state equation in (5) is rewritten as
\[ \dot{x}_2 = U + P(x_1, x_2) \]  
(18)
where $U = M_1^{-1}(x_1)u \in \mathbb{R}^n$ is the nominal control input and $P(x_1, x_2) = -M_1^{-1}(x_1)(C_1(x_1, x_2) + G_n(x_1)) \in \mathbb{R}^n$. Let $U_i$ and $P_i(x_1, x_2)$ denote the $i$-th element of the vector $U$ and $P(x_1, x_2)$, respectively. In order to guarantee (13), $U_i$ has to be saturated as
\[ h_i^- - P_i(x_1, x_2) \leq U_i \leq h_i^+ - P_i(x_1, x_2) \]  
(19)
Note that both $U$ and $P(x_1, x_2)$ incorporate unknown components, i.e., $M_1^{-1}(x_1)$, $C_1(x_1, x_2)$, and $G_n(x_1)$. Thus, we construct RBFNNs to express $M_1^{-1}$ and $P$ as below
\[ M_1^{-1} = \left[ \begin{array}{c} W_{M_1}^T \bullet \{\Phi_M(x_1)\} + E_M \\ P \end{array} \right] = \left[ \begin{array}{c} W_{M_1}^T \bullet \{\Phi_M(x_1)\} + E_M \\ P \end{array} \right] \]
where $\{W_{M_1}^T\}, \{E_{M_1}\}, \{\Phi_M(x_1)\}$, and $\{\Phi_M(x_1, x_2)\}$ are GL functions formed by optimal neural network weight vectors $W_{M_1} = \mathbb{R}^{n_{M_1} \times n}$ and $W_P = \mathbb{R}^{n_{P} \times n}$, and basis function vectors $\Phi_{M_1} = \mathbb{R}^{n_{M_1} \times n}$ and $\Phi_P = \mathbb{R}^{n_{P} \times n}$, respectively. $E_M \in \mathbb{R}^{n \times n}$ and $E_P \in \mathbb{R}^{n \times n}$ are formed by approximation errors $e_{M_1}$ and $e_P$, respectively. Let $E_M$ and $E_P$ denote the matrices formed by $e_{M_1}$ and $e_P$, which are the corresponding upper bounds.
of $e_{M_1}$ and $e_{P_1}$ used to deal with the approximation errors later. Then, unknown matrices $M^{-1}_n$ and $P$ are estimated as

$$M^{-1}_n = \{W_M\}^T \Phi_M(x_1)$$

$$P = \{W_P\}^T \Phi_P(x_1, x_2)$$

(22)
(23)

The nominal control input in (18) is estimated as

$$\hat{U} = (\tilde{M}^{-1} - \delta)u = ((\tilde{W}_M)^T \{\Phi_M(x_1)\} - \delta)u$$

(24)

where $\delta$ will be used later to deal with the presence of functional approximation errors $E_M$. Similarly to (19), with the estimated terms (23) and (24), we can obtain the constraints of the estimated nominal control input as follows:

$$h_i^- - \hat{P}_i(x_1, x_2) \leq \hat{U}_i \leq h_i^+ - \hat{P}_i(x_1, x_2)$$

(25)

Hereinafter, we will show that the estimated weights in (23) and (24) will exponentially converge at a rate that can be controlled by properly choosing the design parameters. In addition, the functional approximation errors can be made arbitrarily small by using sufficiently large numbers of neurons [41]. Therefore, (25) is valid to prevent the violation of motion constraints.

Let $U_0 \in \mathbb{R}^n$ and $U_{0i}$ denote the unsaturated nominal control and its $i$-th element, respectively. From (25), the relationship between $\hat{U}_i$ and $U_{0i}$ is given as

$$\hat{U}_i = \begin{cases} h_i^- - \hat{P}_i(x_1, x_2), & \text{if } U_{0i} < h_i^- - \hat{P}_i(x_1, x_2) \\ h_i^+ - \hat{P}_i(x_1, x_2), & \text{if } U_{0i} > h_i^+ - \hat{P}_i(x_1, x_2) \\ U_{0i}, & \text{otherwise} \end{cases}$$

(26)

To achieve the control objective, one needs to properly design $U_0$, which will be introduced later. Then, according to (24) and (26) and taking the inverse of $(\tilde{W}_M)^T \{\Phi_M(x_1)\} - \delta$, we have

$$u' = (\{\tilde{W}_M\}^T \{\Phi_M(x_1)\} - \delta)^{-1} \hat{U}$$

(27)

Note that, in (27), the estimated term $\{\tilde{W}_M\}^T \{\Phi_M(x_1)\} - \delta$ may be ill-conditioned or even singular. Thus, we perform singular value decomposition [47] with this term and have

$$\{\tilde{W}_M\}^T \{\Phi_M(x_1)\} - \delta = U_s \Sigma_s V_s^T$$

(28)

where $U_s$ and $V_s \in \mathbb{R}^{n \times n}$ are two orthogonal matrices, and $\Sigma_s \in \mathbb{R}^{n \times n}$ is a diagonal matrix formed by the singular values $\sigma_{si}$ of the matrix $\{\tilde{W}_M\}^T \{\Phi_M(x_1)\} - \delta$. Then, instead of forming the pseudoinverse of $\{\tilde{W}_M\}^T \{\Phi_M(x_1)\} - \delta$ by directly replacing every nonzero diagonal entry with their reciprocals, we define $\Sigma_s^+ = \text{diag}(\sigma_{s1}^+, \ldots, \sigma_{sn}^+)$ where

$$\sigma_{si}^+ = \begin{cases} 0, & \text{if } 0 \leq \sigma_{si} < \tilde{\sigma} \\ \sigma_{si}^{-1}, & \text{otherwise} \end{cases}$$

(29)

for $i = 1, \ldots, n$, and $\tilde{\sigma}$ is a small positive design parameter. According to the above definition, the actual control is obtained as below

$$u = V_s \Sigma_s^+ U_s^T \hat{U}$$

(30)

Now, we proceed to develop $U_0$ in (26) following the backstepping techniques. Firstly, we define the error variables $z_1 = y - y_d = x_1 - x_d$ and $z_2 = x_2 - \alpha_1$, where $\alpha_1 \in \mathbb{R}^n$ is a virtual control variable. Considering system dynamics (5), the time derivative of $z_1$ is

$$\dot{z}_1 = z_2 + \alpha_1 - \dot{y}_d$$

(31)

The virtual control variable is designed as

$$\alpha_1 = \dot{y}_d - L_1 z_1$$

(32)

where $L_1 = L_1^T > 0$. Substituting (32) into (31), we have

$$\dot{z}_1 = z_2 - L_1 z_1$$

(33)

According to (5), the time derivative of $z_2$ is

$$\dot{z}_2 = \dot{x}_2 - \dot{\alpha}_1 = U + P(x_1, x_2) - \dot{\alpha}_1$$

(34)

where $\dot{\alpha}_1 = \dot{y}_d - L_1 \dot{z}_1$. To analyze the saturation effect of the estimated nominal control (26), the following auxiliary design system is given

$$\dot{\xi} = \left\{ \begin{array}{ll} -L_{21} \xi + \frac{\sqrt{\|\Delta \dot{U}\|}}{\|\alpha\|} \xi + \Delta \dot{U}, & \|\xi\| \geq \chi \\ 0, & \|\xi\| < \chi \end{array} \right.$$ 

(35)

where $\Delta \dot{U} = \dot{U} - U_0, L_{21} = \text{diag}(l_{21}, \ldots, l_{2n}) = L_{11}^T > 0, L_{21} = L_{21}^T > 0, \chi$ is a small positive design parameter, and $\xi \in \mathbb{R}^n$ is the state of the auxiliary design system. Let $L_{20} = U_0, L_{20}^T > 0$, the unsaturated nominal control $U_0$ is given as

$$U_0 = -L_{21}^{-1} z_1 - L_{21}^{-1} L_{20}(z_2 - \xi) - \{W_P\}^T \{\Phi_P(x_1, x_2)\} + \dot{\alpha}_1$$

(36)

The term $\delta$ and the update laws for vectors $\tilde{W}_P, \tilde{W}_{Mij}$ in GL matrices $\{\tilde{W}_P\}, \{\tilde{W}_{Mij}\}$ are designed as:

$$\delta_{ij} = -\text{sgn}(z_{2i}l_{2ij}) s_{ij}$$

(37)

$$\tilde{W}_P = A_i(\phi_P(x_1, x_2) z_{2i}l_{2i} - \bar{\beta}_i W_{P_i})$$

(38)

$$\tilde{W}_{Mij} = \Gamma_{ij}(\phi_{Mij}(x_1, x_2) u_{ij} z_{2i}l_{2ji} - \gamma_{ij} \tilde{W}_{M_{ij}})$$

(39)

where $\text{sgn}(\cdot)$ is the sign function, $s_{ij}$ is a constant gain that satisfies $s_{ij} \geq e_{M_{ij}}, A_i \in \mathbb{R}^{n \times n}$, $L_{21} = L_{21}^T > 0, \Gamma_{ij} \in \mathbb{R}^{n \times n}, \Gamma_{ij} = \Gamma_{ij}^T > 0, \beta_i, \gamma_{ij} > 0$. \textbf{Remark 4:} [48] If the use of the discontinuous sign function $\text{sgn}(z_{2i}l_{2ij})$ in (37) is undesired, an alternative $\delta$ can be chosen as $\delta_{ij} = -s_{0ij} f(z_{2i}l_{2ij}), \text{where } s_{0ij}$ is a positive constant and $f(z_{2i}l_{2ij})$ is a continuous odd function of $z_{2i}l_{2ij}$. In such a case, it is clear that $\int_{t_0}^{t} L_2(E_M u + \delta u) \leq \sum_{i=1}^{n} \sum_{j=1}^{n} |z_{2i}l_{2ij}|(e_{Mij} - s_{0ij} f(z_{2i}l_{2ij})) \leq 0$ for all $z_{2i}l_{2ij}$ outside the region

$$\Omega_0 = \{z_{2i}l_{2ij} \text{sgn}(z_{2i}l_{2ij}) f(z_{2i}l_{2ij}) < \frac{e_{M_{ij}}}{s_{0ij}} \}$$

which can be made arbitrarily small by increasing the value of the design constant $s_{0ij}$.

\textbf{Theorem 1:} Consider the robotic manipulator (5) satisfying Assumptions 1-2, with the actual control input (30) and neural network weight update laws (38) and (39). For bounded initial conditions, the closed-loop signals $z_1, z_2, \xi, \{\tilde{W}_M\}$ and $\{\tilde{W}_P\}$ are uniformly bounded. In addition, the tracking error $z_1$ remains within the compact set $\Omega_z$ and it will eventually and exponentially converge to the steady state compact set $\Omega_z$ to be defined in the following proof. Finally, motion constraints (6) are not violated.
Proof 1: Consider a Lyapunov function candidate
\[ V^* = \frac{1}{2} z^T L_1 z_1 + \frac{1}{2} \xi^T \xi + \frac{1}{2} z^T L_2 z_2 \] (40)

In the following derivations, let us first consider the case when \( \|\xi\| \geq \chi \) in (35) while the other condition \( \|\xi\| < \chi \) will be discussed later. Invoking (33) to (35), we have
\[
\dot{V} = -z_1^T L_1 z_1 + z_1^T z_2 - \xi^T L_2 \xi - \frac{1}{2} \Delta W^T \Delta U + \xi^T \Delta U
- z_2^T \Delta U |U (P(x_1, x_2) - \alpha_1)
\] (41)

As \( -\frac{1}{2} \Delta W^T \Delta U + \xi^T \Delta U \leq \frac{1}{2} \xi^T \xi \), we have
\[
\dot{V} \leq -z_1^T L_1 z_1 + z_1^T z_2 - \xi^T ((L_2 - \frac{1}{2} I)) \xi
- |z_2^T \Delta U |U (P(x_1, x_2) - \alpha_1)
\] (42)

where \( I \) denotes the identity matrix with a proper dimension. As \( U = \dot{U} + U - \dot{U} \), we obtain
\[
\dot{V}^* \leq -z_1^T L_1 z_1 + z_1^T z_2 - \xi^T ((L_2 - \frac{1}{2} I)) \xi
+ z_2^T L_2 (U + U - \dot{U}) \] (43)

Substituting (36) into (43), we have
\[
\dot{V}^* \leq -z_1^T L_1 z_1 + z_1^T z_2 - \xi^T ((L_2 - \frac{1}{2} I)) \xi
+ z_2^T L_2 (P (x_1, x_2) - \alpha_1)
\] (44)

where \( \{\hat{W}_P\} = \{\hat{W}_P\} - \{W_P\} \). Consider
\[
U - \hat{U} = -(W_M)^T \cdot \{\Phi_M (x_1)\} u + (E_M u + \delta u) \] (45)

where \( \{W_M\} = \{\hat{W}_M\} - \{W_M\} \). Substituting (37) into the term \( z_1^T L_2 (E_M u + \delta u) \), we have \( z_1^T L_2 (E_M u + \delta u) \leq \sum_{i=1}^{n} \sum_{j=1}^{n} \|z_2^T L_2 u_j\| (\varepsilon M_{ij} - \varepsilon (z_1^T L_2 u_j) \varepsilon) \leq 0 \).

Through the above mathematical manipulations, we have
\[
\dot{V}^* \leq -z_1^T L_1 z_1 - \xi^T ((L_2 - \frac{1}{2} I)) \xi - z_2^T L_2 (z_2 - \xi)
+ z_2^T L_2 (P (x_1, x_2) - \alpha_1)
\] (46)

To investigate the convergence of the error signals \( \{\hat{W}_P\} \) and \( \{W_M\} \), the augmented Lyapunov function candidate is given as
\[
V = V^* + \frac{1}{2} \sum_{i=1}^{n} \hat{W}_P^T A_i^{-1} \hat{W}_P + \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \hat{W}_P^T G_i^{-1} \hat{W}_M
\] (47)

It is easy to find that
\[
\frac{1}{2} z_1^T L_2 z_1 \leq \frac{1}{2} z_1^T L_2 z_2 + \frac{1}{2} \sigma z_1^T L_2 z_2 \] (48)
\[
\frac{1}{2} z_2^T L_2 z_2 \leq \frac{1}{2} z_2^T L_2 z_2 + \frac{1}{2} ||L_2 E_P||^2 \] (49)

where \( \sigma > 0 \). Considering (48), (49) and the update laws (38), (39), we have
\[
\dot{V} \leq -z_1^T L_1 z_1 - \xi^T ((L_2 - \frac{1}{2} I)) \xi
- z_2^T L_2 (P (x_1, x_2) - \alpha_1)
- \sum_{i=1}^{n} \sum_{j=1}^{n} \Gamma_{ij} \hat{W}_M^T \hat{W}_M + \frac{1}{2} ||L_2 E_P||^2
\] (50)

Based on the facts
\[
\hat{W}_P^T \hat{W}_P \geq \frac{1}{2} ||\hat{W}_P||^2 - \frac{1}{2} ||W_P||^2 \] (51)
\[
\hat{W}_M^T \hat{W}_M \geq \frac{1}{2} ||\hat{W}_M||^2 - \frac{1}{2} ||W_M||^2 \] (52)

we obtain
\[
\dot{V} \leq -z_1^T L_1 z_1 - \xi^T (L_2 - 0.5I - 0.5\sigma^{-1} L_2 W_{20} L_20) \xi
- z_2^T (L_2 - (0.5\sigma + 0.5)I) z_2 + \frac{1}{2} ||L_2 E_P||^2
\]
\[
\leq -z_1^T L_1 z_1 - \xi^T (L_2 - 0.5I - 0.5\sigma^{-1} L_2 W_{20} L_20) \xi
- z_2^T (L_2 - (0.5\sigma + 0.5)I) z_2 + \frac{1}{2} ||L_2 E_P||^2
\]
\[
- \sum_{i=1}^{n} \sum_{j=1}^{n} \beta_{ij} ||\hat{W}_P||^2 - \sum_{i=1}^{n} \sum_{j=1}^{n} \gamma_{ij} ||\hat{W}_M||^2
\]
\[
\leq -\rho V + \zeta \rho \] (53)

\[
\rho = \min (2\lambda_{\text{min}}(L_1), 2\lambda_{\text{min}}(L_21), 2\lambda_{\text{min}}(L_20 L_2^{-1})),
\min (\frac{\beta_{ij}}{\lambda_{\text{max}}(L_2^{-1})}, \min(\frac{\gamma_{ij}}{\lambda_{\text{max}}(L_1^{-1})})) \] (54)

\[
\zeta = \sum_{i=1}^{n} \frac{\beta_{ij}}{2} ||\hat{W}_P||^2 + \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{\gamma_{ij}}{2} ||\hat{W}_M||^2 + \frac{1}{2} ||L_2 E_P||^2 \] (55)

with \( L_2 = L_2 - 0.5 \sigma + 0.5 I \) and \( L_21 = L_21 - 0.5 \sigma L_20 L_20^{-1} \). To ensure \( \rho > 0 \), the design parameters \( \beta_{ij} > 0, \gamma_{ij} > 0, \sigma > 0, L_2 = L_20 > 0, L_20 = L_20 > 0, \) and \( L_2 = L_21 > 0 \) are to satisfy the following conditions
\[
\lambda_{\text{min}}(L_1) > 0, \lambda_{\text{min}}(L_21) > 0, \lambda_{\text{min}}(L_20) > 0 \] (56)

According to Lemma 1.2 i) in [49], (53) indicates that all the closed-loop signals \( z_1, z_2, \xi, \{W_M\} \) and \( \{W_P\} \) are uniformly bounded. Particularly, the tracking error \( z_1 \) is uniformly bounded by a compact set \( \Omega_{z_1} \) derived as follows:

Multiplying (53) by \( e^{\rho t} \) yields
\[
\frac{d}{dt} (V(t)e^{\rho t}) \leq \zeta e^{\rho t}. \] (57)

By integrating (57) over \([0, t] \), we obtain
\[
0 \leq V(t) \leq (V(0) - \frac{\zeta}{\rho}) e^{-\rho t} + \frac{\zeta}{\rho} \] (58)

From (58), it is easy to find that
\[
0 \leq V(t) \leq (V(0) - \frac{\zeta}{\rho}) e^{-\rho t} + \frac{\zeta}{\rho} \leq V(0) + \frac{\zeta}{\rho} \] (59)

According to (47), we have \( \frac{1}{2} z_1^T z_1 \leq V(t) \). Thus,
\[
||z_1||^2 \leq 2(V(0) + \frac{\zeta}{\rho}). \] (60)

Hence, the tracking error \( z_1 \) is uniformly bounded by the compact set
\[
\Omega_{z_1} = \{z_1(t)||z_1(t)|| \leq \sqrt{2(V(0) + \frac{\zeta}{\rho})}\} \] (61)

In addition, according to (59) and Lemma 1.2 ii) in [49], we can see that \( z_1 \) is uniformly ultimately bounded and it
From (54) and (55), the size of $\Omega_s$ can be reduced to be sufficiently small by designing $\rho$ such that it is large enough. In addition, from (58), the convergence speed of $z_1$ can also be boosted with a large $\rho$. In other words, the robot’s end-effector can be kept inside the constrained region at steady state by simply designing $\rho$ that is large enough.

Remark 5: The state $\xi$ in the auxiliary design system (35) indicates whether there exists saturation of the nominal input. In particular, $\|\xi\| \geq \chi$ means that there exists saturation of $U_0$, which has been considered in the above stability analysis. If there is no saturation, we have $\|\xi\| < \chi$ and $\Delta U = 0$, i.e., $\hat{U} = U_0$, and thus the nominal input is bounded. Therefore, following (40) to (55), it is easy to show that Theorem 1 is still valid [38].

Remark 6: To guarantee that the approximation property of RBFNNs (20) and (21) holds, their inputs have to be in the corresponding compact sets, i.e., $x_1 \in \Omega_{x_1}$ and $[x_1^T, x_2^T]^T \in \Omega_{x_1, x_2}$, respectively. From Theorem 1 and Proof 1, it is clear that the ultimate bound of $z_1$ can be made arbitrarily small by properly choosing the parameters such as $L_1$ and $L_20$. Therefore, $x_1$ and $[x_1^T, x_2^T]^T$ can be guaranteed to be in the corresponding compact sets.

Remark 7: In Theorem 1, the designed unsaturated nominal control $U_0$ and the corresponding weight update laws (38) and (39) lead to the Uniformly Ultimate Boundedness (UUB) of the tracking error $z_1$. It is shown in Proof 1 that this ultimate bounds can be made arbitrarily small by increasing $\rho$. It is worth mentioning that the asymptotic convergence result of $z_1$ can also be obtained within our control design framework. To do this, we can simply add a term $-\text{sgn}(z_2)z_2 L_3 \hat{E}_2$, to $U_0$ in (36) to cancel the effects of the NN approximation error $E_P$ in Lyapunov analysis (46) and choose $\gamma_{ij} = \beta_i = 0$. In this way, it is clear that $\dot{V} \leq 0$ and $\dot{V} < 0$ for all $z_1 \neq 0$. However, this result comes at a cost of possible chattering in the system due to usage of the discontinuous sign function.

Remark 8: The proposed method is motivated by controlling a robotic manipulator with motion constraints, but it is also applicable to other systems which are affine in control, e.g., ocean vessels [50].

### Table I

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m_1$</td>
<td>Mass of link 1</td>
<td>1.151kg</td>
</tr>
<tr>
<td>$m_2$</td>
<td>Mass of link 2</td>
<td>0.575kg</td>
</tr>
<tr>
<td>$l_1$</td>
<td>Length of link 1</td>
<td>0.31m</td>
</tr>
<tr>
<td>$l_2$</td>
<td>Length of link 2</td>
<td>0.24m</td>
</tr>
<tr>
<td>$I_1$</td>
<td>Moment of inertia of link 1</td>
<td>0.3kgm²</td>
</tr>
<tr>
<td>$I_2$</td>
<td>Moment of inertia of link 2</td>
<td>0.3kgm²</td>
</tr>
</tbody>
</table>

IV. SIMULATION

In this section, we simulate a 2-DOFs robotic manipulator moving in a horizontal plane under different types of motion constraints. The following general simulation scenarios can reflect a variety of new robotic applications such as robotic tool use [51] and human-robot physical collaboration [52], especially in unstructured environments where asymmetric, time-varying and task-dependent motions constraints are usually required to guarantee the safety of operations. The dynamic model of the robotic manipulator is given by Eq. (63) and values of the parameters are given in Table I.

The initial positions of the end-effector are given as $q(0) = [-\frac{1}{2}\pi, \frac{1}{2}\pi]^T$. The desired trajectory is a circular path in the task space, which is described by $y_d(t) = 0.5 \cos(0.5t)$ and $y_{d2}(t) = 0.5 \sin(0.5t)$. To quantify and evaluate the tracking performance, the tracking error is defined as $e_i = |y_i - y_{d_i}|$ for $i = 1, 2$. Motion constraints of the robotic manipulator are defined as $-0.6 \leq y_1 \leq 0.6, -0.6 \leq y_2 \leq 0.6, -0.4 \leq \dot{y}_1 \leq 0.4$ and $-0.4 \leq \dot{y}_2 \leq 0.4$.

The design parameters are chosen as $k_1 = [1000, 1000]^T$, $k_2 = [100, 100]^T$, $k_3 = [10^4, 10^4]^T$, $L_1 = \text{diag}[5, 7]$, $L_20 = \text{diag}[500, 500]$, $L_21 = \text{diag}[100, 1000]$, $L_2 = \text{diag}[1, 2]$, $\sigma = 200$, $\chi = 0.001$, $s_{ij} = 0.001$ for $i, j = 1, 2$, $\xi(0) = [0.001, 0.001]^T$ and $\bar{\sigma} = 10^{-10}$. For the RBFNNs (22) and (23), the number of nodes $n_{wM} = n_{wP} = 2^7 = 128$. $\Gamma_{ij} = \Lambda_i = 0.01 \times I_{128 \times 128}$ and $\bar{\beta}_i = \gamma_{ij} = 0.1$ for $i, j = 1, 2$. The centers of the radial basis functions are evenly distributed in $[-1, 1]$ and their variance is set to be 1. The initial weights are $w_{M11} = 0.3 \times [1, 1, \cdots, 1]^T \in \mathbb{R}^{128}$ and $w_{M12} = w_{M21} = w_{M22} = w_{P1} = w_{P2} = 0.1 \times [1, 1, \cdots, 1]^T \in \mathbb{R}^{128}$.

A guideline for choosing the above parameters are given as follows: as the parameters $k_1, k_2, k_3$ are used when converting a motion constraint into the constraint of its derivative in (9), (11) and (12), they should be sufficiently large so that the converted motion constraints are hard enough and the resulted constrained region will not be over-conservative. The diagonal matrices $L_1, L_20, L_2$ are gains for the signals $z_1$ and $z_2$ used in the standard back-stepping control design procedure. Generally, $L_1$ and $L_20$ should be tuned according to the position and velocity errors, respectively and $L_2$ is a global scaling factor for these two gains. $L_21$ is a gain for the auxiliary signal designed to handle the saturation effects of the nominal input signals, which generally gives satisfactory results with the default values. As mentioned in Proof 1, all these gains should be properly chosen such that (56) is satisfied. On the other hand, parameters $\sigma, \chi, \bar{\sigma}$ are used for stability analysis and there is generally no need for tuning them. As for the learning rates $\Lambda, \Gamma, \lambda, \gamma$, they should be chosen to be large enough to achieve fast and stable online learning. More instructions on designing adaptive RBFNNs can be found in [39] and other related papers.

To investigate the efficacy of the proposed method, comparative simulation studies are conducted on its transient responses with and without motion constraints and the results are shown in Figs. 1 and 2. Note that the adaptive control without motion constraints in Fig. 2 is also implemented with the proposed method, where position and velocity constraints are defined with infinite values.

Figure 1 shows that, under the proposed control, the end-effector of the robot successfully tracks the desired trajectory.
and the simulated situation is reduced to a standard adaptive
with those in Fig. 2, where no motion constraint is imposed
proposed method can be obtained by comparing above results
Fig. 1. Robot’s transient response with motion constraints
quickly. However, since no motion constraint is considered ,
end-effector also successfully tracks the desired trajectory
while not violating the motion constraints. During the transient
response, the position and velocity of the end-effector are per-
fectly limited within the constrained region marked by the pink
dash-dotted lines. More specifically, in Figs. 1(c) and 1(d), we
can observe that the end-effector’s velocity stops increasing
when it reaches the boundary of the constrained region. This
demonstrates that the proposed control can effectively prevent
the violation of motion constraints.

\[
M(q) = \begin{bmatrix}
m_1 l_2^2 + m_2 l_2^2 + l_1 l_2 \cos q_2 + I_1 + I_2 & m_2 l_2^2 + l_1 l_2 \cos q_2 + I_2 \\
m_2 l_2^2 + l_1 l_2 \cos q_2 + I_2 & m_2 l_2^2 + I_2
\end{bmatrix},
\]

\[
C(q, \dot{q}) = \begin{bmatrix}
-m_2 l_1 c_2 \dot{q}_2 \sin q_2 & -m_2 l_1 c_2 (\dot{q}_1 + \dot{q}_2) \sin q_2 \\
m_2 l_1 c_2 \dot{q}_1 \sin q_2 & 0
\end{bmatrix},
G(q) = 0
\]

(63)

From this comparison, we show that motion constraints
the velocities of the end-effector rise to a much higher level
can be incorporated into our control design without causing
compared to those in Figs. 1(c) and 1(d) as the system driven
any degradation to the tracking performance. In addition, for
by the designed control tries to track the desired trajectory as
applications where the overshoot of position and/or velocity
soon as possible.
in the transient response is undesired, the proposed control
provides an explicit and direct way to address this problem by
adding some reasonable motion constraints into consideration.

A more illustrative demonstration of the efficacy of the
proposed method can be obtained by comparing above results
with those in Fig. 2, where no motion constraint is imposed
and the simulated situation is reduced to a standard adaptive
tracking control problem. As can be seen from Fig. 2, the
end-effector also successfully tracks the desired trajectory
quickly. However, since no motion constraint is considered,
0.2 \cos(t) \leq y_2(t) \leq 0.6 - 0.2 \sin(t), -0.5 - 0.1 \cos(t) \leq \dot{y}_1(t) \leq 0.5 - 0.1 \sin(t) \text{ and } -0.5 - 0.1 \cos(t) \leq \dot{y}_2(t) \leq 0.5 - 0.1 \sin(t). 

The simulation results are shown in Fig. 3.

\begin{figure}[h]
\centering
\subfigure[$y_1$ and $\dot{y}_1$]{{
\includegraphics[width=0.5\textwidth]{fig3a.png}
}}
\subfigure[$y_2$ and $\dot{y}_2$]{{
\includegraphics[width=0.5\textwidth]{fig3b.png}
}}
\subfigure[$\dot{y}_1$ and $\dot{y}_2$]{{
\includegraphics[width=0.5\textwidth]{fig3c.png}
}}
\subfigure[$\dot{y}_2$ and $\dot{y}_2$]{{
\includegraphics[width=0.5\textwidth]{fig3d.png}
}}
\subfigure[Control input $u$]{{
\includegraphics[width=0.5\textwidth]{fig3e.png}
}}
\caption{Robot’s transient response with time-varying asymmetric motion constraints}
\end{figure}

From Fig. 3, we can observe that excellent tracking performance can still be achieved with the same configuration of control parameters when the motion constraints have varied significantly. Both of the time-varying asymmetric position and velocity constraints (marked by the pink dash-dotted curves) are not violated even when the constrained regions are shrinking at the beginning of the transient response. These results show the efficacy of the proposed method in handling time-varying motion constraints, which corresponds to the cases when motion constraints are a function of time, the robot’s state or its operation environment.

It is worth mentioning that larger design parameters $k_1$, $k_2$ and $k_3$ result in harder motion constraints, i.e., the position/velocity will get nearer to or even stay at the boundaries of the constrained region when it is to escape this region. However, a larger control input is required to keep it at the boundary, as shown in Fig. 3(e), and mild chattering in the control input may appear. Therefore, a trade-off between tracking performance at the boundaries of the constrained region and control effort needs to be considered.

V. EXPERIMENT

In this section, the proposed method is further examined through an experiment, which is conducted on a 2-DOF planar robotic manipulator as shown in Fig. 4. The robot is composed of 2 MAXON motors with two MAXON EPOS2 70/10 dual loop controllers. The links of the robot are of lengths $l_1 = 0.14m$ and $l_2 = 0.15m$. A desktop PC is used to process the collected data and implement the proposed method.

The initial joint position of the robot is $q(0) = [0.5, 0.5]^T$ and the desired trajectory of the end-effector is $y_d(t) = [0.24 \cos(0.05 t + \frac{\pi}{2}), 0.24 \sin(0.05 t + \frac{\pi}{2})]^T$ for $t \in [0, 4]s$. The motion constraints are defined as $-0.245 \leq y_1 \leq 0.245, -0.245 \leq y_2 \leq 0.245, -0.1 \leq \dot{y}_1 \leq 0.1$ and $-0.05 \leq \dot{y}_2 \leq 0.05$.

The design parameters are chosen as $k_1 = [1, 30]^T$, $k_2 = [0.05, 1]^T$, $k_3 = [1, 1]^T$, $L_1 = \text{diag}(30, 30)$, $L_2 = \text{diag}(130, 12)$, $L_{21} = \text{diag}(500, 500)$, $L_2 = \text{diag}(2.5, 0.4)$, $\sigma = 20$, $\chi = 0.001$, $s_{i,j} = 0.001$ for $i,j = 1, 2$, $\xi(0) = [0.001, 0.001]^T$ and $\bar{\sigma} = 10^{-10}$. For the RBFNNs (22) and (23), the number of nodes $n_{W_M} = n_{W_P} = 2^7 = 128$. The centers of the radial basis functions are evenly distributed in $[-1, 1]$ and their variance is set to be 1. $\Lambda_i = 0.1 I_{128 \times 128}$, $\beta_i = 0.1$, $\Gamma_{1,1} = \Gamma_{2,1} = \Gamma_{2,2} = 0.05 I_{128 \times 128}$, $\gamma_{i,j} = 0.01$ for $i,j = 1, 2$. The initial weights are $W_{M11} = 0.6 \times [1, 1, \ldots, 1]^T \in \mathbb{R}^{128}$, $W_{M12} = W_{M21} = W_{M22} = -0.1 \times [1, 1, \ldots, 1]^T \in \mathbb{R}^{128}$, and $W_{P1} = W_{P2} = 0.0 \times [1, 1, \ldots, 1]^T \in \mathbb{R}^{128}$.

The transient response of the robot is shown in Fig. 5. As shown in Figs. 5(a) and 5(b), the end-effector is able to track the desired trajectory in about 3.5 seconds and position constraints (marked by pink dash-dotted lines) are constantly satisfied. From Figs. 5(c) and 5(d), we can see that the velocity...
of the end-effector is successfully limited within the prescribed constrained region. For the comparison purpose, we consider another experiment without velocity constraints. As a result, a larger velocity appears in the beginning of the transient response, as also shown in Figs. 5(c) and 5(d), due to the fact that the control tries to drive the end-effector to the desired position as fast as possible. These results are in accordance to the simulation results in the previous section, and illustrate the efficacy of the proposed method. Different from the simulation results, the velocities in the experiment are not so smooth, which is because they are obtained by differentiating the positions collected from the encoders. A filter may be designed to obtain a cleaner velocity signal if it is required in some applications.

VI. Conclusion

In this paper, we have investigated the control of a robotic manipulator with parametric uncertainties and two types of motion constraints. A framework has been proposed to convert motion constraints of different types into a unified constraint of the nominal input, based on which an adaptive neural network control has been developed. The efficacy of the proposed method has been demonstrated through simulation and experiment studies.

REFERENCES

Y. J. Liu, S. Tong, and Y. Gao, “Fuzzy adaptive control with
Y. J. Liu, Y. Gao, S. Tong, and Y. Li, “Fuzzy approximation-based
et al.

Mingming Li received the B.Eng. degree in communication engineering from Sun Yat-sen University, Guangzhou, China, in 2013. He is currently pursuing his Ph.D. degree with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore. His current research interests include adaptive control, associative memories and artificial intelligence.

Yanan Li (M’14) received the B.Eng degree in control science and engineering and the M.Eng degree in control and mechatronics engineering, from the Harbin Institute of Technology, China, in 2006 and 2009, respectively, and the Ph.D. degree from the NUS Graduate School for Integrative Sciences and Engineering, National University of Singapore, Singapore, in 2013. He has been a Research Scientist at the Institute for Infocomm Research, Agency for Science, Technology and Research (A*STAR), Singapore from 2013 to 2015. Currently, he is a Research Associate at the Department of Bioengineering, Imperial College London, UK. His general research interests include physical human-robot interaction and human-robot collaboration.

Shuzhi Sam Ge (S’90-M’92-SM’99-F’06) received the B.Sc. degree from Beijing University of Aeronautics and Astronautics, Beijing, China, in 1986, and the Ph.D. degree from the Imperial College of Science, Technology and Medicine, University of London, London, U.K., in 1993. He is the Founding Director of the Social Robotics Laboratory, Interactive Digital Media Institute, National University of Singapore and on leave with University of Electronic Science and Technology of China, Chengdu 610054, China. He has authored or coauthored six books and more than 400 international journal and conference papers. His current research interests include social robotics, multimedia fusion, medical robots, and intelligent systems.

Dr. Ge is the Editor-in-Chief of the International Journal of Social Robotics. He has served/been serving as an Associate Editor for a number of flagship journals. He also serves as an Editor of the Taylor & Francis Automation and Control Engineering Series. He also served as the Vice President of Technical Activities, 2009-2010, and the Vice President for Membership Activities, 2011-2012, IEEE Control Systems Society.
**Tong Heng Lee** (M’90) received the B.A. degree (with first class honors) in engineering tripos from Cambridge University, Cambridge, U.K., in 1980 and the Ph.D. degree in electrical engineering from Yale University, New Haven, CT, in 1987.

He is a Professor with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore, and is currently the Head of the Drives, Power, and Control Systems Group in this department. He is the Deputy Editor-in-Chief of the International Federation of Automatic Control Mechatronics International Journal and serves as the Associate Editor of many other flagship journals. He has also coauthored three research monographs and holds four patents (two of which are in the technology area of adaptive systems, and the other two are in the area of intelligent mechatronics). His research interests are in the areas of adaptive systems, knowledge-based control, intelligent mechatronics, and computational intelligence.

Dr. Lee received the Cambridge University Charles Baker Prize in Engineering.