Task-space regulation of rigid-link electrically-driven robots with uncertain kinematics using neural networks

Wenhui Si, Lingyan Zhao, Jianping Wei and Zhiguang Guan

Abstract
Extensive research efforts have been made to address the motion control of rigid-link electrically-driven (RLED) robots in literature. However, most existing results were designed in joint space and need to be converted to task space as more and more control tasks are defined in their operational space. In this work, the direct task-space regulation of RLED robots with uncertain kinematics is studied by using neural networks (NN) technique. Radial basis function (RBF) neural networks are used to estimate complicated and calibration heavy robot kinematics and dynamics. The NN weights are updated on-line through two adaptation laws without the necessity of off-line training. Compared with most existing NN-based robot control results, the novelty of the proposed method lies in that asymptotic stability of the overall system can be achieved instead of just uniformly ultimately bounded (UUB) stability. Moreover, the proposed control method can tolerate not only the actuator dynamics uncertainty but also the uncertainty in robot kinematics by adopting an adaptive Jacobian matrix. The asymptotic stability of the overall system is proven rigorously through Lyapunov analysis. Numerical studies have been carried out to verify efficiency of the proposed method.

Keywords
Task-space regulation, neural networks, RLED robot, uncertain kinematics, asymptotic stability

Date received: 12 September 2020; accepted: 25 November 2020

Introduction
A great many methods for motion control of robot manipulators have been proposed in literature in the past decades. It is noticed that for many of these control methods such as Slotine and Weiping1; Tomei2; Arimoto3; Kelly4; Lin5; Bascetta and Rocco6; Wai and Muthusamy7; Nojavanzadeh and Badamchizadeh8; Wu et al.,9 the controllers are formulated in the joint space. Nevertheless, for more and more modern applications of robot manipulators (e.g. service robot, surgical robot, and other human-robot interactions) the control task is defined in task/operational space. One major constraint of the joint-space control methods is that in order to fulfill the given task-space control objectives the desired or reference joint position or path must be calculated based on the robot’s inverse kinematics. Considering the complexity of inverse kinematics calculation, a first regulation control method that is designed directly in task-space was developed in Takegaki and Arimoto10 by using a transposed Jacobian matrix. Later on, many other results have been developed for task-space robot manipulator control such as Miyazaki and Masutani11; Lewis et al.12; Kelly13; Kelly et al.14 These task-space control methods are based on the common basis that the kinematic information of the robot manipulator is accurately known such that the exact Jacobian matrix mapping joint-space velocity to its task-space counterpart is available. But in real practice, the robot kinematics may not be perfectly known mainly due to two sources of uncertainty: robot calibration errors; end position change of the grasped object after/during pick-up manipulation and interaction with the working environment. In such cases, the robot kinematics cannot be accurately obtained and thus lead to degraded performance or even instability if the above
task-space control methods are directly applied. To handle the uncertain kinematics issue, several task-space control strategies have been proposed based on the concept of “approximate Jacobian matrix” in Yazarel and Cheah\(^\text{15}\); Liu and Cheah\(^\text{16}\); Liu et al.\(^\text{17}\).

It should be noted that, including the above-mentioned control methods, many robot manipulator motion control methods in literature only consider the manipulator dynamics assuming the required control torque can be faithfully generated by perfect joint actuators with no time-delay. However, if good control performance for the robot manipulator with high-speed and high-accuracy requirements is to be achieved, the actuator dynamics should be considered in the overall system control design as well as indicated by Good et al.\(^\text{18}\). To explicitly consider the actuator dynamics in control system design, several methods based on nonlinear feedback linearization, singular perturbation, adaptive, robust and fuzzy techniques have been developed to address the motion control problem for rigid-link electrically-driven (RLED) robots as in Good et al.\(^\text{19}\); Guenther and Hsu\(^\text{20}\); Su and Stepchenko\(^\text{21}\); Burg \textit{et al.}\(^\text{22}\); Ishii \textit{et al.}\(^\text{23}\); Oya \textit{et al.}\(^\text{24}\); Khorashadizadeh and Fateh\(^\text{25}\); Soltanpour \textit{et al.}\(^\text{26}\); Rani and Kumar.\(^\text{26}\) During the last few decades, neural network (NN) techniques have been employed for motion control of RLED robot manipulators, such as Lewis \textit{et al.}\(^\text{27}\); Ge and Hang\(^\text{28}\); Kim and Lewis.\(^\text{29}\) The advantage of neural network techniques when used for robot manipulator control is their capability to online approximate unknown nonlinear models. In Kwan \textit{et al.}\(^\text{30}\) a robust NN controller was proposed for robots with actuator dynamics. The proposed method only guarantees uniformly ultimately bounded (UUB) stability of the closed-loop system and hence the position errors are not guaranteed to converge. Later on, other types of NN-based methods have been proposed to address the same control problem with similar conclusions on motion tracking errors such as Huang \textit{et al.}\(^\text{31}\); Wu \textit{et al.}\(^\text{32}\); Zeng and Wang.\(^\text{33}\) The work in Shafiei and Soltanpour\(^\text{34}\) shows that asymptotic error convergence can be achieved using robust neural network control. But all these results are confined to joint-space control and only system uncertainties in dynamics are considered. Robot kinematics uncertainties are not taken into account in those control designs and therefore they suffer the same problem as the aforementioned joint-space controllers.\(^\text{1-14,18-26}\)

In this paper, we propose a new adaptive neural networks control method for task-space regulation of RLED robots with uncertainties existing in kinematics. The main contribution of the proposed method lies in that, different from most existing NN-based controllers for RLED robots, asymptotic stability of the overall system can be guaranteed with the presence of all uncertainties in the robot kinematics, dynamics and the actuator dynamics. Also, compared with the control methods using approximate Jacobian matrix to handle kinematics uncertainty in Yazarel and Cheah\(^\text{15}\); Liu and Cheah\(^\text{16}\); Liu \textit{et al.}\(^\text{17}\) no task-space velocity measurement is required. Two radial basis function (RBF) neural networks are employed to approximate unknown dynamics functions involving uncertain kinematics terms and the NN weights are updated on-line through two adaptation laws without the necessity of off-line training. The numerical studies confirm that the developed method is efficient for the RLED robot regulation task considering the uncertainties in kinematics and dynamics of the robot and actuator system.

**Preliminaries**

**Important properties of RLED robot**

**Kinematics.** When the control task is formulated in the operational/task space, reference position or trajectory for the robot manipulator to follow is usually defined directly in the corresponding operational/task space. The kinematic relationship between the joint space coordinates and the corresponding task space coordinates can be defined as:\(^\text{17}\):

\[
X = h(q),
\]

where \(X \in \mathbb{R}^m\) represents the manipulator end-point position vector in task space, \(q \in \mathbb{R}^n\) is the generalized joint coordinate vector, \(h(\cdot) : \mathbb{R}^n \to \mathbb{R}^m (m \leq n)\) is the nonlinear mapping function describing the relationship between the joint and task space. The task-space velocity vector \(\dot{X}\) can be linked with the joint-space velocity vector \(\dot{q}\) as:

\[
\dot{X} = J(q)\dot{q},
\]

where \(J(q) \in \mathbb{R}^{m \times n}\) is the Jacobian matrix mapping joint space to task space. It should be noted that the Jacobian matrix becomes uncertain if uncertainties exist in the robot kinematics.

**Dynamics.** When the actuator dynamics is considered in the overall robot manipulator system, the dynamics of RLED robot with \(n\) degrees of freedom can be described in the following form which consists two subsystems:

\[
M(q)\ddot{q} + C(q, \dot{q})\dot{q} + F(q, \dot{q}) + g(q) = K_N I,
\]

\[
LI + RI + K_F \dot{q} = u,
\]

where \(I \in \mathbb{R}^n\) represents the armature current vector and \(u \in \mathbb{R}^n\) the armature voltage vector. \(M(q) \in \mathbb{R}^{n \times n}\) represents the manipulator inertia matrix which is positive definite; \(C(q, \dot{q}) \in \mathbb{R}^{n \times n}\) denotes the centrifugal and coriolis force; \(g(q) \in \mathbb{R}^n\) is the gravity force vector. \(F(q, \dot{q})\) is the vector of joint friction which comprises several friction terms as shown in following:\(^\text{35}\):

\[
F(q, \dot{q}) = f_0 + f_\text{s} \text{sgn}(\dot{q}) + f_\text{c} \dot{q} + f_\text{a} \text{atan}(f_\text{s}\dot{q}),
\]

where \(f_0\) represents the zero drift error of friction torque, \(f_\text{c}\) the Coulomb friction coefficient, \(f_\text{a}\) the viscous
friction coefficient, and \( f_a \) and \( f_b \) denote the friction coefficients. \( \text{sgn}(\cdot) \) is a sign function defined as \( \text{sgn}(\cdot) = -1 \) for \( \cdot < 0 \), \( 1 \) for \( \cdot > 0 \) and \( 0 \) for \( \cdot = 0 \).

For the voltage-driven actuator dynamics which is expressed by (4), \( L \in \mathbb{R}^{n \times n} \) denotes the matrix of actuator inductance; \( R \in \mathbb{R}^{n \times n} \) represents the matrix of actuator resistance, \( K_E \in \mathbb{R}^{n \times n} \) is the actuator voltage constant matrix and \( K_N \in \mathbb{R}^{n \times n} \) represents the matrix of actuator torque constant characterizing the electro-mechanical conversion relationship between actuator armature current and corresponding generated joint torque. The above actuator dynamic matrices are positive definite and diagonal, and the element values in each matrix need careful calibration to know and would slowly change as the motor temperature changes and therefore may lead to uncertainties during the control process.

In this work, some important properties about the dynamics and kinematics of the robot manipulator will be used in the control design and stability analysis and hence are introduced here:

Property A: The inertia matrix \( M(q) \) is positive definite and symmetric.

Property B: The matrix \( C(q, \dot{q}) \) and the time derivative of the inertia matrix \( \dot{M}(q) \) satisfy the following relationship:

\[
\dot{q}^T \frac{1}{2} M(q) - C(q, \dot{q}) \dot{q} = 0, \quad \forall \ q, \dot{q} \in \mathbb{R}^n
\]  

Property C: The robot kinematics parameters can be linearized and extracted in form of a vector as

\[
J^T(q) \tau = Y_k(q, \tau) \theta_k, \quad \forall \ \tau \in \mathbb{R}^n
\]

where \( \theta_k \in \mathbb{R}^\ell \) is the robot kinematics parameter vector and \( Y_k(q, \tau) \in \mathbb{R}^{n \times \ell} \) is the kinematics regressor.

Property A, B will be shown important in the Lyapunov analysis of overall system stability, and Property C enables an adaptive Jacobian matrix to be handled to control the uncertainty in robot kinematics.

### Neural networks approximation

Neural networks possess many important properties. In control engineering, its capability to approximate a given nonlinear function \( f(x) \) up to a small error is the most useful one of concern. In this work, radial basis function (RBF) neural network is considered for the approximation purpose due to its merits of universal approximation and faster learning speed compared to other neural networks. As illustrated in Figure 1, the RBF network is a type of feed forward neural network composed of three layers, namely the input layer, the hidden layer, and the output layer. Gaussian function is chosen as the activation function in the computational units (nodes) of its hidden layer. The Gaussian function is given as

\[
\phi_i(x) = \exp\left( -\frac{(x - \mu_i)^T (x - \mu_i)}{\sigma^2} \right), \quad i = 1, \ldots, p
\]

where \( \mu_i \in \mathbb{R}^n \) is the center vector and \( \sigma \in \mathbb{R} \) is the variance. Define \( W \) as the collection of NN weights so that the output of the network is given by \( y \sim W \phi(x) \):

\[
y = W \phi(x),
\]

where \( \phi(x) = [\phi_1(x) \phi_2(x) \cdots \phi_p(x)]^T \) is the basis function vector. A nonlinear function \( f(x) \) can therefore be approximated by an RBF network with some “ideal” constant weights \( W \) as

\[
f(x) = W \phi(x) + \varepsilon,
\]

where \( \varepsilon \) is acceptable small NN functional reconstruction error. As will be seen in this paper, different from NN-based classification applications where the weights are fixed through off-line training, the weight matrix \( W \) used in proposed controller is on-line updated by a properly chosen adaptation law.

### Adaptive neural network regulation of RLED robot

In this section, a task-space adaptive neural network regulation method is proposed for RLED robots. The control development employs the backstepping technique. The general idea is to separate the actuator control voltage input design into two steps: the first step is to formulate an actuator armature current reference command \( I_d \) that should force the task-space position errors to converge based on the robot manipulator dynamics as defined in (3) despite the existence of uncertainties in the robot kinematics and actuator torque constants; then by using the actuator dynamics as defined in (4), the ultimate actuator control voltage input \( u \) is designed to make sure that the actual armature current \( I \) tracks the reference current \( I_d \). The asymptotic stability of the overall system is proved rigorously using Lyapunov theory.

![Figure 1. RBF neural network.](image-url)
Armature current reference design

The robot manipulator dynamics as defined in (3) could be re-formulated by considering the reference armature current \( I_d \) as

\[
M(q)\ddot{q} + C(q, \dot{q})\dot{q} + F(q, \dot{q}) + g(q) = K_N I_d + K_N \dot{I}
\]

(11)

where \( \dot{I} = I - I_d \) represents the deviation of actual armature current from the reference current. The new manipulator dynamics as in (11) can be considered as a system driven by input \( K_N I_d \) with disturbance \( K_N \dot{I} \).

Considering the existence of uncertainty in robot kinematics, the accurate Jacobian matrix is not available. Then an adaptive Jacobian matrix \( \hat{J}(\theta) \) is proposed according to Property C as

\[
\hat{J}^T(q, \hat{\theta}_k)s(e) = Y_1(q, e)\hat{\theta}_k,
\]

(12)

where \( e = X - X_d = (e_1, \ldots, e_m)^T \) represents the positioning error given a desired task-space position \( X_d \in \mathbb{R}^m \), \( \hat{\theta}_k \) is the estimated kinematics parameter vector updated by an adaptation law as define in (16). \( s_i(e)(i = 1, \ldots, m) \) represents a saturated function of \( e \). \( s_i(e) \) and its integration \( S_i(e) \) are illustrated in Figure 2 and possess the following properties:

1. \( S_i(e) > 0 \) if \( e \neq 0 \) and \( S_i(0) = 0 \).
2. \( S_i(e) \) is twice continuously differentiable, and its derivative \( s_i(e) = \frac{\partial S_i(e)}{\partial e} \) is strictly increasing for \( |e| < \gamma_i \) and saturated if \( |e| \geq \gamma_i \) for certain positive parameter \( \gamma_i \).
3. There are constants \( \hat{c}_i > 0 \) such that for \( e \neq 0 \),

\[
S_i(e) \geq \hat{c}_i s_i^2(e).
\]

An illustrative example of such kind of saturated function could be the hyperbolic tangent function \( \tanh(*) \).

Since the actuator dynamics parameters are assumed to be unknown, let \( \hat{K}_N \) be the approximate actuator torque constant matrix. Then the armature current reference \( I_d \) can be formulated as

\[
I_d = \hat{K}_N^{-1}(-K_p \hat{J}^T(q, \hat{\theta}_k)s(e) - K_\nu \dot{\hat{\theta}}_k + \hat{W}_1 \phi_1 - \nu_1),
\]

(14)

where \( K_p = K_p I_E \), \( K_\nu = k_1 I_E \in \mathbb{R}^{n \times n} \) are two positive definite control parameters and \( I_E \) is the identity matrix. \( \nu_1 \in \mathbb{R}^m \) is a robustifying term to be defined later in (39). \( \hat{W}_1 \phi_1 \) is the NN approximation of a complicated nonlinear function which is defined in following and its role will be clear as shown in equation (20). \( \phi_1 \) constitutes suitable basis functions of the neural network, the weight matrix \( \hat{W}_1 \) is updated by

\[
\hat{W}_1 = -L_1 y \phi_1^T,
\]

(15)

where \( y = \dot{\hat{\theta}} + \alpha \hat{J}^T(q, \hat{\theta}_k)s(e) \) and \( \alpha \in \mathbb{R} \) is a constant scalar. The adaptive kinematic parameter vector \( \hat{\theta}_k \) in the adaptive Jacobian matrix \( \hat{J}(q, \hat{\theta}_k) \) is updated by

\[
\dot{\hat{\theta}}_k = \Psi L_k \hat{Y}_1^T(q, e)\dot{\hat{\theta}}_k,
\]

(16)

where \( \Psi = \text{diag}\{ \Psi_1, \ldots, \Psi_f \} \in \mathbb{R}^{n \times f} \) is the projection operator defined as:

\[
\Psi_i = \begin{cases} 
0, & \text{if } \hat{\theta}_k \geq \theta_{\text{max}} \text{ and } \dot{\hat{\theta}}_k \geq 0 \\
0, & \text{if } \hat{\theta}_k \leq \theta_{\text{min}} \text{ and } \dot{\hat{\theta}}_k \leq 0 \\
1, & \text{otherwise}
\end{cases}
\]

(17)

Since the real kinematic parameters in \( \theta_k \) have physical meanings and are limited in value, \( \theta_{\text{max}} \) and \( \theta_{\text{min}} \) are used to denote the estimated maximum and minimum limits of \( \theta_k \). The projection operator \( \Psi \) is used to limit \( \theta_k \) in the feasible range of \( [\theta_{\text{min}}, \theta_{\text{max}}] \) and hence is bounded.37

Substituting the desired current (14) into (11), we have

Figure 2. (a) The integration \( S(e) \) of saturated function (b) The saturated function \( s(e) \).
The above expression can be rewritten with a modified estimation error and manipulator dynamics can be obtained as:
\[ f = \tilde{W}_1\phi_1 + K_N\tilde{K}_N^{-1}v_1 = K_N\tilde{I}. \]

It can be further written as
\[
\begin{align*}
M(q)\ddot{q} + C(q, \dot{q})\dot{q} + F(q, \dot{q}) + g(q) \\
+ K_N\tilde{K}_N^{-1}K_p\tilde{J}^T(q, \dot{\theta}_k)s(e) + K_N\tilde{K}_N^{-1}K_q\dot{q} \\
- K_N\tilde{K}_N^{-1}\tilde{W}_1\phi_1 + K_N\tilde{K}_N^{-1}v_1 = K_N\tilde{I}.
\end{align*}
\] (18)

The following nonlinear function can be approximated by a RBF neural network using basis function \( \phi_1 \) with some constant “ideal” weights \( W_1 \) as
\[
F(q, \dot{q}) + g(q) + (I - K_N\tilde{K}_N^{-1})[K_p\tilde{J}^T(q, \dot{\theta}_k)s(e) - K_q\dot{q}].
\] (19)

Substituting (24) into (23), we have
\[ \tilde{W}_1 = W_1 - \tilde{W}_1 \] denotes the weight matrix approximation error.

**Control voltage input design**

After a desired armature current has been formulated, in this step a control voltage input will be designed which should serve to make the actual current follow the desired signal. From the actuator dynamics subsystem (4), we have
\[
\dot{I} + LI + W_2\phi_2 + e_2 = u.
\] (25)

The control voltage input can be proposed as
\[ u = \tilde{W}_2\phi_2 - K_\tilde{I}\tilde{I} - v_2, \] (26)
where \( K_\tilde{I} \in \mathbb{R}^{m \times n} \) is a positive definite control parameter, \( v_2 \) is a robustifying term, the approximate weight matrix is updated through
\[ \tilde{W}_2 = -L_2\tilde{I}\tilde{W}_2. \] (27)

Substituting \( u \) into equation (25), the closed-loop actuator dynamics can be got as
\[ \dot{\tilde{W}}_2 = -W_2\tilde{W}_2 - e_2 + v_2 = 0 \] (28)
where \( \tilde{W}_2 = W_2 - \tilde{W}_2 \) denotes the weight matrix approximation error.

**Overall system stability analysis**

In the preceding two subsections, the closed-loop manipulator and actuator subsystems have been obtained with the proposed desired current and control voltage input. Now the stability of the overall closed-loop system will be evaluated using Lyapunov theory.

We propose a Lyapunov function for the overall RLED robot system in the following form
\[ V = \frac{1}{2}\dot{q}^TM(q)\dot{q} + \alpha\dot{q}^TM(q)\tilde{J}^T(q, \dot{\theta}_k)s(e) \]
\[ + \sum_{i=1}^{m-1}(k_p + ak_i)S_i(e) + \frac{1}{2}\dot{I}^TLI \]
\[ + \frac{1}{2}\sum_{i=1}^{n-1}\tilde{W}_iK_N^{-1}\tilde{W}_j^T + \frac{1}{2}\sum_{i=1}^{n-1}\tilde{W}_iL_{ki}^{-1}(k_p + ak_i)\dot{\theta}_k \] (29)

where \( \tilde{W}_i \) is the \( i \) th row of the matrix \( \tilde{W}_j \) (\( j = 1, 2 \)).

Note that
\[ 1 \leq 0 < \alpha, k_p, k_i \]
and the control parameters \( \alpha, k_p, \) and \( k_i \) can be chosen freely in our control design. If \( \alpha \) is chosen small enough and/or \( k_p \) and \( k_i \) are chosen large enough, then the following inequality can be achieved:
\[ (k_p + ak_i)\dot{\epsilon}_i - \alpha^2\lambda_m > 0, \] (30)

and the control parameters \( \alpha, k_p, \) and \( k_i \) can be chosen freely in our control design. If \( \alpha \) is chosen small enough and/or \( k_p \) and \( k_i \) are chosen large enough, then the following inequality can be achieved:
\[ (k_p + ak_i)\dot{\epsilon}_i - \alpha^2\lambda_m > 0, \] (31)
where \( \lambda_m = \lambda_{\text{max}}(\mathbf{J}(q, \dot{q}_h)M(q)J^T(q, \dot{q}_h)) \), \( \lambda_{\text{max}}[^*] \) is the maximum eigenvalue of matrix \( * \).

Hence by substituting inequality (31) into (29), it has

\[
V > \frac{1}{4} \hat{q}^T M(q) \hat{q} + \sum_{i=1}^{m} |(k_p + ak_k) \dot{c}_i - \alpha^2 \lambda_m | \hat{s}_i^2 (c_i) \\
+ \frac{1}{2} \hat{J}_L^T \hat{L} \hat{J} + \sum_{n=1}^{i-1} \frac{W_{ii}}{K_{N_iL_{ii}}} \hat{W}_i^T + \frac{1}{2} \hat{L}_k^T \hat{L} \hat{J}_k^T \hat{L}_k \\
+ \frac{1}{2} \sum_{n=1}^{i-1} W_{2i} \frac{1}{L_{2i}} \hat{W}_i^T + \frac{1}{2} \hat{L}_k^T \hat{L}_k^{-1} (k_p + ak_k) \dot{c}_k
\]

(32)

It’s clear that if inequality (31) is satisfied, the Lyapunov candidate \( V > 0 \).

Differentiating \( V \) with respect to time, it can be obtained that

\[
\dot{V} = \hat{q}^T M(q) \hat{q} + \frac{1}{2} \hat{q}^T \hat{J}_L \hat{J} + \alpha \hat{q}^T \mathbf{J}(q, \dot{q}_h) M(q) \hat{q} \\
+ \hat{q}^T M(q) \hat{J}_L \hat{J} (q, \dot{q}_h) s(e) + \hat{q}^T M(q) \hat{J}_L \hat{J} (q, \dot{q}_h) \hat{q}(q, \dot{q}_h) s(e) \\
+ \hat{q}^T M(q) \hat{J}_L \hat{J} (q, \dot{q}_h) \hat{s}(e) + \hat{q}^T \mathbf{C}(q, \dot{q}_h) \hat{J}_L \hat{J} (q, \dot{q}_h) \hat{s}(e) \\
+ \hat{q}^T \mathbf{C}(q, \dot{q}_h) \hat{J}_L \hat{J} (q, \dot{q}_h) \hat{s}(e) - \hat{q}^T \mathbf{C}(q, \dot{q}_h) \hat{J}_L \hat{J} (q, \dot{q}_h) \hat{s}(e) \\
\leq \alpha (c_0 \| \hat{q} \|^2 + c_1 \| s(e) \|^2).
\]

(34)

Substituting \( M(q) \hat{q} \) from equations (22) and \( \hat{L} \hat{J} \) from (28) into (33) and using inequality (34) and the definition \( y = \hat{q} + \alpha \hat{J}_L(q, \dot{q}_h) s(e) \) as in (15), we have

\[
\dot{V} \leq -q^T K_s \hat{q} - \alpha k_p s^T(e) \hat{J}(q, \dot{q}_h) \hat{J}_L^T (q, \dot{q}_h) s(e) \\
- (k_p + ak_k) q^T \hat{J}_L (q, \dot{q}_h) s(e) - q^T \hat{J}_L (q, \dot{q}_h) s(e) \\
- q^T K_s \hat{q} + \alpha c_0 \| \hat{q} \|^2 + \alpha c_1 \| s(e) \|^2 \\
- (q + \alpha \hat{J}_L (q, \dot{q}_h) s(e)) \hat{J}_L [\hat{W}_1 \phi(q) \\
+ K_N \hat{K}_N^{-1} \hat{W}_2 \phi_e + \epsilon_i + K_N \hat{K}_N^{-1} \nu_1 - K_N \hat{I}] \\
- \hat{T} \hat{W}_2 \phi_e - \hat{T} \epsilon_2 - \hat{T} \nu_2 - \sum_{n=1}^{i-1} \frac{W_{1i}}{K_{N_iL_{ii}}} \hat{W}_i^T + \frac{1}{2} \hat{L}_k^T \hat{L}_k^{-1} (k_p + ak_k) \dot{c}_k
\]

(35)

here Property B has been used.

Substituting the NN weights updating laws (15) and (27) and using equation (12), the above inequality can be written as

\[
\dot{V} \leq -q^T K_s \hat{q} - \alpha k_p s^T(e) \hat{J}(q, \dot{q}_h) \hat{J}_L^T (q, \dot{q}_h) s(e) \\
+ (k_p + ak_k) \hat{q}^T \hat{J}_L (q, \dot{q}_h) s(e) - q^T \hat{J}_L (q, \dot{q}_h) s(e) \\
+ \alpha c_0 \| \hat{q} \|^2 + \alpha c_1 \| s(e) \|^2 - y^T \hat{W}_1 \phi(q) \\
+ y^T K_N \hat{K}_N^{-1} \hat{W}_2 \phi_e - y^T \epsilon_1 - y^T K_N \hat{K}_N^{-1} \nu_1 + y^T K_N \hat{I} \\
- \hat{T} \epsilon_2 - \hat{T} \nu_2 - (k_p + ak_k) \hat{q}^T \hat{J}_L (q, \dot{q}_h) s(e) \\
- \sum_{n=1}^{i-1} \frac{W_{1i}}{K_{N_iL_{ii}}} \hat{W}_i^T - \hat{L}_k^T \hat{L}_k^{-1} (k_p + ak_k) \dot{c}_k
\]

(36)

According to adaptation law (16) and the definition of the projection operator, it’s easily verified that

\[
(k_p + ak_k) \hat{q}^T \hat{J}_L (q, \dot{q}_h) s(e) > 0.
\]

(37)

Hence we have

\[
\dot{V} \leq -q^T K_s \hat{q} - \alpha k_p s^T(e) \hat{J}(q, \dot{q}_h) \hat{J}_L^T (q, \dot{q}_h) s(e) \\
- (k_p + ak_k) q^T \hat{J}_L (q, \dot{q}_h) s(e) - q^T \hat{J}_L (q, \dot{q}_h) s(e) \\
- q^T K_s \hat{q} + \alpha c_0 \| \hat{q} \|^2 + \alpha c_1 \| s(e) \|^2 - y^T \hat{W}_1 \phi(q) \\
+ y^T K_N \hat{K}_N^{-1} \hat{W}_2 \phi_e - y^T \epsilon_1 - y^T K_N \hat{K}_N^{-1} \nu_1 + y^T K_N \hat{I} \\
- \hat{T} \epsilon_2 - \hat{T} \nu_2 - (k_p + ak_k) \hat{q}^T \hat{J}_L (q, \dot{q}_h) s(e) \\
- \sum_{n=1}^{i-1} \frac{W_{1i}}{K_{N_iL_{ii}}} \hat{W}_i^T - \hat{L}_k^T \hat{L}_k^{-1} (k_p + ak_k) \dot{c}_k
\]

(38)

Now define the two robustifying terms \( v_1 \) and \( v_2 \) as

\[
v_1 = \frac{\lambda_{\text{max}}[\hat{K}_N]}{\lambda_{\text{min}}[\hat{K}_N]} \epsilon_1 \cdot \text{sgn}(y),
\]

(39)

\[
v_2 = \epsilon_2 \cdot \text{sgn}(\tilde{J}),
\]

(40)

where \( \epsilon_1 \) and \( \epsilon_2 \) are given bounds of the two NN reconstruction errors \( \epsilon_1 \) and \( \epsilon_2 \), respectively.

Then inequality (38) can be simplified to

\[
\dot{V} \leq -q^T K_s \hat{q} - \alpha k_p s^T(e) \hat{J}(q, \dot{q}_h) \hat{J}_L^T (q, \dot{q}_h) s(e) \\
- (k_p + ak_k) q^T \hat{J}_L (q, \dot{q}_h) s(e) - q^T \hat{J}_L (q, \dot{q}_h) s(e) \\
- q^T K_s \hat{q} + \alpha c_0 \| \hat{q} \|^2 + \alpha c_1 \| s(e) \|^2 \\
- (k_p + ak_k) \hat{q}^T \hat{J}_L (q, \dot{q}_h) s(e) - \hat{T} \hat{K}_L \hat{I} \\
+ (k_p + ak_k) \hat{q}^T \hat{J}_L (q, \dot{q}_h) s(e) - \hat{T} \hat{K}_L \hat{I} \\
- \hat{T} \epsilon_2 + \hat{T} \nu_2 - (k_p + ak_k) \hat{q}^T \hat{J}_L (q, \dot{q}_h) s(e) \\
- \sum_{n=1}^{i-1} \frac{W_{1i}}{K_{N_iL_{ii}}} \hat{W}_i^T + \frac{1}{2} \hat{L}_k^T \hat{L}_k^{-1} (k_p + ak_k) \dot{c}_k
\]

(41)
where $\lambda_m = \lambda_{\text{max}}[K_1]$, $b_j = \|J(q, \theta_k)\|$, $b_{JJ} = \|\dot{J}(q, \theta_k)\|$, and

$$P = \begin{bmatrix}
    k_v & 0 & -\frac{1}{2}\lambda_m \\
    0 & \alpha(k_pb_{JJ} - c_1) & -\frac{1}{2}\alpha b_p \lambda_m \\
    -\frac{1}{2}\lambda_m & -\frac{1}{2}\alpha b_p \lambda_m & \lambda_{\text{min}}[K_1]
\end{bmatrix}$$

(42)

According to Sylvester’s Theorem, if the following conditions are satisfied

$$k_p b_{JJ} - c_1 > 0,$$

(43)

$$4 b_{JJ} b_{\text{min}}[K_1] - b_{JJ}^2 \lambda_m^2 - \frac{\alpha}{k_p} b_p^2 \lambda_m^2 > 0,$$

(44)

then matrix $P$ is positive definite and hence it has

$$\dot{V} < -\lambda_{\text{min}}[P](\|\dot{q}\|^2 + \|s(e)\|^2 + \|I\|^2).$$

(45)

Then we can reach the following Theorem:

Theorem: With the armature current reference design (14), the control voltage input $u$ (26) and the parameter updating laws given by (15), (16), (27), the overall RLED robot system achieves the convergence of $X \rightarrow X_d$ and $\dot{q} \rightarrow 0$ as $t \rightarrow \infty$ if the conditions (31), (43), and (44) are satisfied.

Proof: If condition (31) is satisfied, it is clear from inequality (32) that $V$ is positive definite in $\dot{q}$, $s(e)$, $I$ and adaptive parameter vectors $W_1$, $W_2$, and $\theta_k$. With conditions (43) and (44) satisfied, $\dot{V}$ is shown to be negative definite in $\dot{q}$, $s(e)$ and $I$ and hence $V$ is bounded as well as the variables $\dot{q}$, $s(e)$, $I$, $W_1$, $W_2$, and $\theta_k$. Then from equation (22), it can be concluded that $\dot{q}$ is bounded and consequently its integration $\dot{q}$ is uniformly continuous. Since $\dot{q}$ is bounded, then $\dot{e} = J(\ddot{q})\ddot{q}$ is bounded as well and hence its time integration $e$ is uniformly continuous as well as the saturated function $s(e)$. Due to the fact that $V$ is bounded and $\dot{V}$ is negative definite as seen from (45), it has $s(e), \dot{q} \in C_2(0, \infty)$. Then it follows from Arimoto\textsuperscript{3} and Desoer and Vidyasagar\textsuperscript{39} that $s(e) \rightarrow 0$ and $\dot{q} \rightarrow 0$ as $t \rightarrow \infty$. Since $s(e) = 0$ implies $e = X - X_d = 0$, we have $X \rightarrow X_d$ as $t \rightarrow \infty$. \hfill $\Box$

Remark 1: It should be noted that the stability conditions (31), (43), and (44) are not really restrictive and can be achieved by simply choosing large enough control gains $K_p$, $K_l$, and small enough scalar $\alpha$. Also these conditions are sufficient conditions to achieve system stability according to Lyapunov analysis and therefore conservative. In practical implementations, stability of the closed-loop system could still be retained even these conditions are relaxed to certain extent.

Remark 2: Although $\alpha$ should be set small enough, it cannot be set to 0. From inequality (41), it can be seen that if $\alpha = 0$ then $\dot{V}$ will be negative definite only in $\dot{q}$ and $I$ but no longer in $s(e)$ which is the system state variable directly linked to position regulation error $e$. As the consequence, we can only conclude the asymptotic convergence of $\dot{q}$ and $I$. And from the closed-loop manipulator dynamics equation (22), by setting $I = 0$ and $\dot{q} = 0$ (therefore $\ddot{q} = 0$ too), no conclusion about the convergence of $s(e)$ can be obtained. Therefore, the parameter $\alpha$ is introduced to enable the convergence of position regulation error $e$ by playing its role in the adaptation law for the first neural network weight parameters $W_1$ as shown in (15). Theoretically and intuitively, it makes sense to include the positioning error information in the NN weights adaptation law such that the neural network will keep evolving until the position error $e$ converges to 0 and thus the regulation task is achieved.

Numerical study

In order to evaluate the efficiency of the proposed control strategy, numerical studies have been carried out based on a 2-link RLED robot holding an object as illustrated in Figure 3. The regulation task is to move the end-point of the grasped object from a staring position to a desired position defined in the robot base frame which can be taken as the world frame or Cartesian frame in this study. The object position is supposed to be obtained through an external measure system (e.g. camera, optical tracker, etc.). The robot kinematics parameters together with the object length and its grasping angle are assumed to be unknown as well as the manipulator and actuator dynamics parameters, which presents a challenging task for the RLED robot control.

Three case studies were carried out taking into consideration of different initial and desired positions in the regulation control task in order to demonstrate the general effectiveness of the proposed controller within the robot workspace and its robustness against uncertainties. Case 1 sets the initial position of the object end-point to $X_0 = (0.8, 0.32)$ and the desired position to $X_d = (0.5, 0.55)$. Two different levels of kinematic and dynamic uncertainties are designed in this case study as well, that is, small (Case 1.a) and large (Case 1.b) deviations from the true values. Case 2 has
the same initial position $X_0 = (0.8, 0.32)$ but different desired position $X_d = (0.65, 0.5)$. In Case 3, both the initial and desired positions are set at different locations as $X_0 = (0.69, 0.74)$ and $X_d = (0.4, 0.9)$.

It should be noted that for all three case studies the same control parameters have been used as

$$K_p = 3500I_E, K_r = 1500I_E, \alpha = 4, K_I = 1500I_E,$$

$$L_1 = L_2 = 1500I_E, L_s = I_E.$$  

The two Gaussian RBF neural networks used in this simulation each contains 20 nodes, and the values of $\mu$ and $\sigma$ are randomly chosen from $[-25, 25]$ and $[1.5, 2]$ respectively in advance and fixed during the simulation. The uncertainty settings for Case 2 and Case 3 are the same as the one for Case 1b where larger uncertainties are introduced in the system set-up as defined later.

Ground truth settings

The true robot manipulator and object kinematic and dynamic parameters (ground-truth) are set to the following:

- Mass (kg): $m_1 = 17.4, m_2 = 4.8, m_o = 2$,
- Inertia (kg m$^2$): $I_1 = 6.3, I_2 = 1.136, I_o = 0.549$,
- Friction: $f_o = f_c = f_d = 0.1, f_s = 3$,
- Length (m): $l_1 = 0.4318, l_2 = 0.4318, l_o = 0.2$,
- Center of mass (m): $l_{c1} = 0.068, l_{c2} = 0.07, l_{c0} = 0.1$,

and the object grasping angle is set as $q_0 = 30^\circ$. The true actuator dynamics parameters are set to:

$$L = \text{diag} \{1 1\}, R = \text{diag} \{15 15\},$$

$$K_E = \text{diag} \{10 8\}, K_N = \text{diag} \{10 8\}.$$  

Computational method

The system diagram of the proposed control method is illustrated in Figure 4 and serves as the guidance of the computational method development in the numerical studies. With the ground truth parameters, the 2-link (2-DoF) RLED system dynamics as expressed in (3) and (4) can be explicitly written in form of (46) and (47):

$$M(q)\ddot{q} + \begin{bmatrix} -0.1451\sin(q_1)q_2 + q_1q_2^2 \\ 0.1452u(q_1)q_2^2 \end{bmatrix}$$

$$+ \begin{bmatrix} 0.1 + 0.1\sin(q_2) + 3q_1 + 0.1\tan(0.1q_2) \\ 0.1 + 0.1\sin(q_2) + 3q_1 + 0.1\tan(0.1q_2) \end{bmatrix}$$

$$+ \begin{bmatrix} 40.3705\cos(q_1) + 11.7561\cos(q_1 + q_2) + 1.96\cos(q_1 + q_2 + q_o) \\ 11.7561\cos(q_1 + q_2) + 1.96\cos(q_1 + q_2 + q_o) \end{bmatrix} x$$

$$\begin{bmatrix} 10 \\ 0 \end{bmatrix} [I_1]_k = \begin{bmatrix} I_1 \\ I_2 \end{bmatrix}$$

(46)

where $M(q) = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}$ and

$$M_{11} = 19.253 + 0.2902\cos(q_2) + 0.8995\cos(0.0962 + q_2),$$

$$M_{12} = M_{21} = 1.6851 + 0.1451\cos(q_2) + 0.4498\cos(0.0962 + q_2),$$

$$M_{22} = 11.6851.$$  

Using the defined control parameters and defining the saturation function as $s(e) = \tanh(e)$, the control voltage $u$ and armature current reference $I_d$ as defined in (26) and (14) can be written respectively as

$$u = W_2\phi_2 - 1500\dot{I} - v_2,$$  

(48)

and

$$I_d = \dot{K}_N^{-1}(- 3500\dot{J}(q, \theta_k) \tanh(e) - 1500\dot{\phi})$$

$$+ W_1\phi_1 - v_1,$$  

(49)

where the specific values of $\dot{K}_N, v_1,$ and $v_1$ are different for different uncertainty settings used in each case study.

The adaptive parameters $\dot{W}_1, \dot{W}_2,$ and $\dot{\theta}_k$ are updated according to the following specific adaptation laws:

$$\dot{W}_1 = - 1500(\dot{q} + 4J^T(q, \theta_k) \tanh(e))\phi_1^T,$$  

(50)

$$\dot{W}_2 = - 1500\dot{I}\phi_2^T,$$  

(51)

$$\dot{\theta}_k = \Psi Y^T(q, e)\phi,$$  

(52)

whose general forms are given in (15), (27), and (16) of the control design section.

![Figure 4. Control system diagram.](image-url)
Case studies

Case 1. In the first case study, control performances of the proposed method facing two different levels of system uncertainties are evaluated. First of all, small uncertainties are introduced into the control design and the following inaccurate kinematic and dynamic parameters

\[ \dot{\theta}_{1} = 0.5m, \dot{\theta}_{2} = 0.5m, \dot{\theta}_{0} = 0.25m, \dot{q}_{0} = 45^\circ, \]

and the approximate actuator matrix \( \hat{K}_{N} = \text{diag}\{15, 12\} \) are used in the controller. The position errors, joint velocities and the end-point trajectory in X-Y plane are illustrated in Figure 5. The history plots (overall and transient) of armature current tracking errors are also shown in Figure 5(d). The time evolution of online updating RBF NN weights \( W_{1}, W_{2} \in \mathbb{R}^{2 \times 20} \) is shown in Figure 6.

Next, larger uncertainties are introduced into the RLED robot system and the approximate kinematic and dynamic parameters used in the control design are

\[ \dot{\theta}_{1} = 0.6m, \dot{\theta}_{2} = 0.6m, \dot{\theta}_{0} = 0.3m, \dot{q}_{0} = 60^\circ, \]

and the approximate actuator matrix is set to \( \hat{K}_{N} = \text{diag}\{20, 16\} \). With the same control parameters as in the Case 1.a, the system performance is illustrated as in Figure 7. Similar trajectory and NN weights evolution are observed and hence not included due to space limit.

Case 2. In this case study, the position regulation task is to move from the same initial position \( X_{0} = (0.8, 0.32) \) as Case 1 to a different desired position \( X_{d} = (0.65, 0.5) \). The same uncertainty setting for both kinematics and dynamics as in Case 1.b is used, that is,

\[ \dot{\theta}_{1} = 0.6m, \dot{\theta}_{2} = 0.6m, \dot{\theta}_{0} = 0.3m, \dot{q}_{0} = 60^\circ, \]

\[ \hat{K}_{N} = \text{diag}\{20, 16\} \]

With the same control parameters as in Case 1, the regulation performance is illustrated in Figure 8.

Case 3. In this case study, the position regulation task is to move from a different initial position \( X_{0} = (0.69, 0.74) \) to a different desired position \( X_{d} = (0.4, 0.9) \). The same uncertainty setting for both kinematics and dynamics as in Case 1.b and Case 2 is used. With the same control parameters as in Case 1 and Case 2, the regulation performance is illustrated in Figure 9.

Discussions

From the results of the three numerical studies, it can be seen that despite the existence of relatively large uncertainties in the kinematics and dynamics of RLED robot system the proposed regulation method can always provide good control performance with fast and smooth convergence of position errors \( \epsilon \) and joint velocities \( \dot{q} \). The numerical studies also show that the actuator currents converge to the current references much faster (within 0.2s) than the robot end-point position errors (within 2s), which is logical because the current convergence presents the prerequisite for position error convergence and is the direct consequence of backstepping control strategy.

From the numerical study results, it is also noted that the use of neural network with online updating weight gains does not lead to slow response of the control system which again confirms its utility in real-time robot control as reported in literature. Indeed, the neural network weights settle down quickly as shown in Figure 6. More precisely, the NN weight \( W_{2} \) converges much faster than \( W_{1} \) which can be easily interpreted by its adaptation law

\[ \dot{W}_{2} = -L_{2} \tilde{\phi}_{1}^{T}, \]

where the current tracking error \( \tilde{\epsilon} \) converges very fast to 0. The fast current convergence also explains why almost all parameters in \( W_{2} \) settle down to very small values around 0 (from scale \( 10^{-13} \) to \( 10^{-49} \)) from their initial values 0 as the update stops almost immediately after it starts. Nevertheless, it’s observed that two parameters at random positions in \( W_{2} \) always show much higher steady state values in each test but not affecting the overall control performance, which is hard to interpret due to the black-box nature of neural network.

In comparison, the parameters in \( W_{1} \) show slower and bigger changes in value since their adaptations are driven by slower converging position error \( \epsilon \) and joint velocity \( \dot{q} \) as seen in (15):

\[ \dot{W}_{1} = -L_{1} \tilde{\phi}_{1}^{T} = -L_{1}(\dot{q} + \alpha J^{T}(q, \dot{q})v(e))\tilde{\phi}_{1}^{T}. \]

Another important remark to make is that there’s no guarantee for the adaptive parameters \( (\theta_{k}, W_{1}, W_{2}) \) to converge to their true values since the stability analysis only concludes the boundedness of these parameters.

It’s well known in adaptive parameter identification that to make the adaptive parameters to converge to their true values, the persistent excitation conditions should be satisfied which normally requires one individual frequency for every two parameters to be identified.

For regulation control task, it is difficult to introduce exciting frequencies to meet the persistent excitation conditions since the control reference (desired position) is constant. Comparatively, in trajectory tracking tasks, these conditions are easier to meet as complex enough time-variant trajectory can be designed to contain different frequencies, which is however out of the scope of discussion in this work.
Figure 5. Case 1.a: $X_0 = (0.8, 0.32) \rightarrow X_d = (0.5, 0.55)$ with smaller uncertainties: (a) position errors, (b) joint velocities, (c) regulation trajectory, and (d) current tracking errors.
Figure 6. Case 1.a: Time evolution of NN weights: (a) $\hat{W}_1(1)$, (b) $\hat{W}_1(2)$, (c) $\hat{W}_2(1)$, and (d) $\hat{W}_2(2)$.
Figure 7. Case 1.b: $X_0 = (0.8, 0.32) \rightarrow X_d = (0.5, 0.55)$ with larger uncertainties: (a) position errors and (b) joint velocities.

Figure 8. Case 2: $X_0 = (0.8, 0.32) \rightarrow X_d = (0.65, 0.5)$: (a) position errors and (b) joint velocities.
This paper presents a new task-space regulation method to solve the uncertain kinematics and dynamics problem for rigid-link electrically-driven (RLED) robots by resorting to adaptive Jacobian matrix and neural network technique. The proposed method can tolerate inaccuracy or changes in robot kinematics and dynamics caused by malcalibration or perturbations during the control process. The two Gaussian RBF neural networks employed in the controller can approximate complicated nonlinear dynamics functions and therefore eliminate the tedious and time-consuming regressors development work as required by traditional adaptive control methods. The NN weights are updated on-line without the necessity of off-line training. It is shown through rigorous Lyapunov analysis that asymptotic stability of the closed-loop overall system can be achieved instead of only uniformly ultimate boundedness as concluded by most existing neural network robot control methods. The numerical studies including three case studies with different regulation positions and different levels of system uncertainties have been carried out and the control performances verified the efficiency of the proposed control method. As direct extension of this work, more complex trajectory tracking control task using neural network techniques will be considered as part of the future works. Also, the possibility of handling all uncertainties by neural networks and without use of adaptive control will be explored in the future works.

Declaration of conflicting interests
The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding
The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This research is supported by the Doctoral Foundation Program of Shandong Jianzhu University (NO. X18016Z), Major Science and Technology Innovation Project of Shandong Province (2019JZZY020703), Science and Technology Support Plan for Youth Innovation in Universities of Shandong Province(2019KJB014).

ORCID iD
Zhiguang Guan https://orcid.org/0000-0001-7151-5819

References
1. Slotine JJE and Weiping L. Adaptive manipulator control: A case study. *IEEE Trans Autom Control* 1988; 33(11): 995–1003.
2. Tomei P. Adaptive PD controller for robot manipulators. *IEEE Trans Rob Autom* 1991; 7(4): 565–570.
3. Arimoto S. *Control theory of non-linear mechanical systems*. New York, NY: Oxford University Press, 1996.
4. Kelly R. Global positioning of robot manipulators via PD control plus a class of nonlinear integral actions. *IEEE Trans Autom Control* 1998; 43(7): 934–938.
5. Lin CK. Nonsingular terminal sliding mode control of robot manipulators using fuzzy wavelet networks. *IEEE Trans Fuzzy Syst* 2006; 14(6): 849–859.

6. Bascetta L and Rocco P. Revising the robust-control design for rigid robot manipulators. *IEEE Trans Rob* 2010; 26(1): 180–187.

7. Wai RJ and Muthusamy R. Fuzzy-neural-network inherited sliding-mode control for robot manipulator including actuator dynamics. *IEEE Trans Neural Netw Learn Syst* 2013; 24(2): 273–287.

8. Nojavanzadeh D and Badamchizadeh M. Adaptive fractional-order non-singular fast terminal sliding mode control for robot manipulators. *IET Control Theory Appl* 2016; 10(13): 1565–1572.

9. Wu L, Yan Q and Cai J. Neural network-based adaptive learning control for robot manipulators with arbitrary initial errors. *IEEE Access* 2019; 7: 180194–180204.

10. Takegaki M and Arimoto S. A new feedback method for dynamic control of manipulators. *J Dyn Syst Meas Control* 1981; 102: 119–125.

11. Miyazaki F and Masutani Y. Robustness of sensory feed-forward control based on imperfect Jacobian. In: *Robotics research: the fifth international symposium*, February 1991, pp.201–208. Cambridge, MA: MIT Press.

12. Lewis FL, Abdallah CT and Dawson DM. Control of robot manipulators. New York: Macmillan Publishing Co., 1993.

13. Kelly R. Regulation of manipulators in generic task space: an energy shaping plus damping injection approach. *IEEE Trans Rob Auton* 1999; 15: 381–386.

14. Kelly R, Carelli R, Nasisi O, et al. Stable visual servoing of camera-in-hand robotic systems. *IEEE ASME Trans Mechatron* 2000; 5(1): 39–48.

15. Yazarel H and Cheah CC. Task-space adaptive control of robotic manipulators with uncertainties in gravity regressor matrix and kinematics. *IEEE Trans Autom Control* 2002; 47(9): 1580–1585.

16. Liu C and Cheah CC. Task-space adaptive setpoint control for robots with uncertain kinematics and actuator model. *IEEE Trans Autom Control* 2005; 50(11): 1854–1860.

17. Liu C, Cheah CC and Slotine JJE. Adaptive Jacobian tracking control of rigid-link electrically driven robots based on visual task-space information. *Automatica* 2006; 42(9): 1491–1501.

18. Good MC, Sweet LM and Strobel KL. Dynamic models for control system design of integrated robot and drive systems. *J Dyn Syst Meas Control* 1985; 107(1): 53–59.

19. Guenther R and Hsu L. Variable structure adaptive cascade control of rigid-link electrically-driven robot manipulators. In: *Proceedings of 32nd IEEE conference on decision and control*, San Antonio, TX, 15–17 December 1993, pp.2137–2142. New York: IEEE.

20. Su CY and Stepanenko Y. Redesign of hybrid adaptive/robust motion control of rigid-link electrically-driven robot manipulators. *IEEE Trans Rob Auton* 1998; 14: 651–655.

21. Burg T, Dawson D, Hu J, et al. An adaptive partial state-feedback controller for RLED robot manipulators. *IEEE Trans Autom Control* 1996; 41: 1024–1030.