ORIGINAL ARTICLE

Adaptive Sliding Mode Control of a 6-DOF RLED Robot Manipulator with Uncertain Parameters and External Disturbances

Isro Hutama[1] and Chung-Hsien Kuo[1,*]

[1] *Department of Electrical Engineering, National Taiwan University of Science and Technology*

(**Received** 20 August 2015; **Accepted** 4 January 2016; **Published on line** 1 March 2016)

*Corresponding author: chkuo@mail.ntust.edu.tw

DOI: 10.5875/ausmt.v6i1.1026

Abstract: This paper addresses the end-effector position and orientation tracking problem of a 6-degrees of freedom (DOF) rigid link electrically driven (RLED) revolute joint serial robot manipulator with uncertain parameters and external disturbances. System uncertainties and external disturbances are bounded but their upper limits are unknown. The input matrix, which is always invertible and not necessarily a diagonal matrix, is also assumed to be uncertain with a certain bound. An adaptive sliding mode control (ASMC) is designed to solve the tracking problem given these uncertainties. Note that the designed control method can be applied to any RLED serial robot manipulator. No regression matrices are required and the uncertainty upper limits are not necessarily known. Furthermore, the controller computation time can be reduced by considering some parts of the system dynamics and the nth time-derivative reference signals of nth order system, in this case desired joint jerk signals, as unknown bounded uncertainties. The desired trajectory is defined in operational space while the designed ASMC works in joint space. The methods to convert the end-effector pose, velocity, acceleration, and jerk from operational space to joint space are also explained. The Lyapunov stability criterion is applied to prove the convergence of the adaptive gain of ASMC, the sliding surface, and system stability. Finally, simulation results verified ASMC performance.

Keywords: adaptive sliding mode control, inverse kinematics, trajectory tracking, robot manipulator

Introduction

Robot manipulators have been widely used in various areas, including medical [1], aerospace [2][3], and military applications [4], but they are mostly used in industrial automation to do tasks such as welding [5], painting [6], assembly process [7], and material handling [8]. They can replace human workers in performing dangerous tasks such as bomb disposal and radioactive material handling [9]. They can also perform very precise motions in tasks such as arc welding and surgery. In addition, robot manipulators have been widely deployed to replace human labor in manufacturing to achieve improved efficiency.

There are two well-known types of robot manipulators, parallel and serial. Parallel robot manipulators have advantages in terms of high load/weight ratio, velocity, stiffness, precision, and low inertia [10]. Serial robot manipulators have a larger range of motion than parallel robot manipulators, giving them more freedom to position and orient the end effector.

In most cases, the end effector must be able to assume arbitrary positions and orientations in 3-dimensional (3D) space. To achieve such freedom of movement, a serial robot manipulator needs at least 6 degrees of freedom (DOF), three of which are provided by three revolute joints for orienting, while the remaining three are used for positioning. Each joint is usually equipped with an actuator. Electric motors are used for most actuators, because they are compact and easy to control.

Various motion control methods have been developed for rigid robot manipulators. Most are designed at the torque input level, and actuator dynamics are ignored. However, Tarn et al. [11] showed that actuator dynamics play an important part in the overall robot manipulator dynamics, especially in the

Copyright © 2016 International Journal of Automation and Smart Technology

performance of high velocity motions, and disregarding actuator dynamics may reduce overall system performance or stability [12].

The proposed scheme uses electric DC motors as actuators in the first order system, while the rigid robot manipulator is the second order system. Combining both systems yields a third order system which is called a rigid link electrically driven (RLED) robot manipulator.

Complete knowledge of the dynamics of such a system is difficult to obtain, resulting in modeling inaccuracy. Terms such as friction and external disturbances are usually not modeled. The controllers are expected to follow the reference signals under these uncertainties. Problems raised by joint position tracking for a RLED robot manipulator with uncertain parameters are typically addressed using adaptive control [13][14][15][16][17] and robust control [18][19][20], including sliding mode control (SMC) [21]. Because the dynamics of an RLED robot manipulator have a strict-feedback form, some are designed using a backstepping strategy [21].

However, some of those control methods need to calculate regression matrices [13] - [16], [19] - [21], which are very complex for RLED robot manipulators with many joints, or to have knowledge of the uncertainty upper limits [18][19]. In addition, [17] proposed a function approximation technique (FAT) adaptive controller for RLED robot manipulators which do not require regression matrices but still need to define upper limits of function approximation errors.

In this paper, the adaptive sliding mode control (ASMC) is designed to solve the RLED serial robot manipulator joint position tracking problem with unknown but bounded system uncertainties and external disturbances. Because uncertain parameters are considered, the input matrix is assumed to be uncertain under some conditions. Using ASMC, no regression matrices and uncertainty upper limits are needed. Note that the designed control method is general, not specific for a 6-DOF RLED revolute joint serial robot manipulator, thus it can be applied to any RLED serial robot manipulator. This control method is inspired by Huang et al. [22] who addressed a general affine nonlinear system with unknown but bounded uncertainties and external disturbances with single inputs.

In this paper, the trajectory planning of the end-effector is designed in operational space but the motion control works in joint space. Inverse kinematics is used to convert the desired pose of the end-effector in operational space to the desired joint angles in joint space. The Jacobian matrix, along with the 1st and 2nd time-derivative of the Jacobian matrix are used to convert the desired translational and rotational velocity, acceleration, and jerk of the end-effector in operational space to the desired joint velocities, accelerations, and jerks in joint space. Using this method, the end-effector pose tracking problem becomes a joint angle tracking problem.

Kinematics and Dynamics Models

Inverse Kinematics of the Robot Manipulator

The Denavit-Hartenberg (DH) convention is used to define the Cartesian coordinate frames of a robot manipulator. The DH matrix of a coordinate frame can be expressed in the following form:

$$ {}^{j}\mathbf{Z}_i = \left[\begin{array}{c|c|c|c} {}^{j}\mathbf{x}_i & {}^{j}\mathbf{y}_i & {}^{j}\mathbf{z}_i & {}^{j}\mathbf{t}_i \\ \hline 0 & 0 & 0 & 1 \end{array}\right] \quad (1) $$

where ${}^{j}\mathbf{Z}_i$ represents the pose of the coordinate frame i with respect to the coordinate frame j, ${}^{j}\mathbf{t}_i$ represents the position of the coordinate frame i with respect to the coordinate frame j, ${}^{j}\mathbf{x}_i$, ${}^{j}\mathbf{y}_i$, and ${}^{j}\mathbf{z}_i$ are $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{z}$ axes of the coordinate frame i with respect to coordinate frame j.

The robot manipulator in this paper is a serial robot manipulator, as shown in Fig. 1. There are a total of six joints, all revolute. The DH parameters of the proposed robot manipulator are given in Table I where q_i is the joint angular position or the angle from $\mathbf{x}_{i-1}$ to $\mathbf{x}_i$ with rotation about $\mathbf{z}_{i-1}$, a_i is the offset distance from $\mathbf{z}_{i-1}$ to $\mathbf{z}_i$ along the common normal line on the direction of $\mathbf{x}_i$, d_i is the distance from the origin of the coordinate frame $i-1$ to the common normal line on the direction of $\mathbf{z}_{i-1}$, and α_i is the offset angle from $\mathbf{z}_{i-1}$ to $\mathbf{z}_i$ with rotation about $\mathbf{x}_i$.

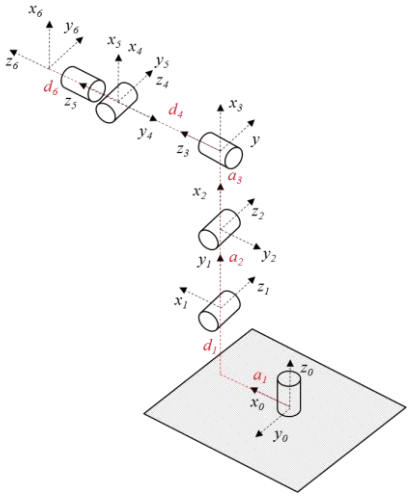


Figure 1. Configuration of the robot manipulator when q=[0 0.5π 0 0 0 0]T.

Inverse kinematics is used to convert the desired end-effector pose in operational space to the desired joint

Copyright © 2016 International Journal of Automation and Smart Technology

angles. Two methods to are commonly used to solve inverse kinematics: algebraic and geometrical methods. Computation using the geometrical method mostly depends on the geometric structure of the robot manipulator. The solution of the inverse kinematics is obtained from equations derived from the geometrical structure of the robot manipulator. This paper uses the geometrical method to solve the inverse kinematics of the robot manipulator.

Figure 2 shows the block diagram describing the relationship between solutions for the robot manipulator with DH parameters as listed in Table 1. The possible number of inverse kinematics solution are 0, infinite, 4 or 8. The number of solutions is zero if the pose of the end-effector cannot be reached. The number of solutions is infinite if q_5 is 0^0 or 180^0, or if the origin of coordinate frames *4* or *5* is on the **z** axis of coordinate frame 0. The number of solutions is 4 if the pose of the end-effector can only be reached by q_1(+) or q_1(-) solutions. The number of solutions is 8 if the pose of the end-effector can be reached by q_1(+) and q_1(-) solutions.

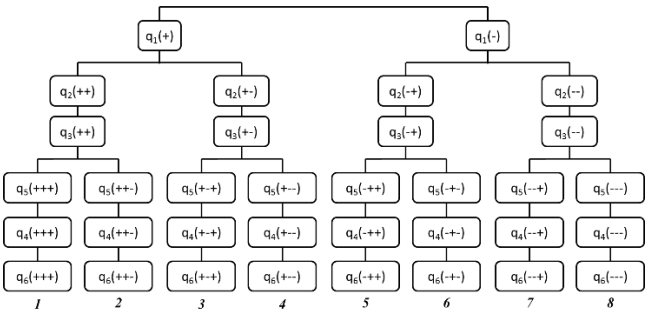


Figure 2. Block diagram of inverse kinematics solutions.

Joint or *i*	q_i	d_i	a_i	α_i
1	q_1	d_1	a_1	90^0
2	q_2	0	a_2	0^0
3	q_3	0	a_3	90^0
4	q_4	d_4	0	-90^0
5	q_5	0	0	90^0
6	q_6	d_6	0	0^0

Table 1. Denavit-hartenberg parameters of the robot manipulator.

First, the desired pose of the robot manipulator end-effector is defined as $\mathbf{E} = {}^0\mathbf{Z}_6$. The origin of coordinate frames *4* and *5* always shares the same position for all solutions. Based on the geometrical structure of the robot manipulator, we can obtain the position using ${}^0\mathbf{o}_4 = {}^0\mathbf{o}_5 = {}^0\mathbf{t}_6 - d_6{}^0\mathbf{z}_6$, where ${}^i\mathbf{t}_j = {}^i\mathbf{o}_j = [{}^io_{jx}\ {}^io_{jy}\ {}^io_{jz}]^T$ is the position of the origin of coordinate frame *j* with respect to coordinate frame *i*.

The DH parameters of the robot manipulator, a_1, d_1, a_2, a_3, and d_4 always share the same two-dimensional frame. Thus a two-dimensional coordinate frame is defined as coordinate frame *ij*, where its origin is the same as the origin of coordinate frame *0*, *j* axis is the same as $\mathbf{z}_0$, and *i* axis is defined by ${}^0\mathbf{i} = [{}^0o_{4x}/(({}^0o_{4x})^2 + ({}^0o_{4y})^2)^{0.5}\ {}^0o_{4y}/(({}^0o_{4x})^2 + ({}^0o_{4y})^2)^{0.5}]^T$. The origin of coordinate frames *4* or *5* can be rewritten with respect to coordinate frame *ij* as ${}^{ij}\mathbf{o}_4 = [(({}^0o_{4x})^2 + ({}^0o_{4y})^2)^{0.5}\ {}^0o_{4z}]^T$. The distance between the origins of coordinate frame *2* and *4* or *5* is always the same. We define it as $c = ((d_4)^2 + (a_3)^2)^{0.5}$.

There are two possible solutions for the angle of joint *1*, defined as q_1(+) and q_1(-). There is a 180^0 difference between them and they respectively result in coordinate frames *1*(+) and *1*(-), as shown in Fig. 3.

Before trying to determine solutions, it is necessary to find out whether the solutions of the end-effector desired pose exist or not. We define two vectors, ${}^{ij}\mathbf{e}(+) = {}^{ij}\mathbf{o}_4 - {}^{ij}\mathbf{o}_4(+)$ and ${}^{ij}\mathbf{e}(-) = {}^{ij}\mathbf{o}_4 - {}^{ij}\mathbf{o}_4(-)$, where ${}^{ij}\mathbf{o}_1(+) = [a_1\ d_1]^T$ and ${}^{ij}\mathbf{o}_1(-) = [-a_1\ d_1]^T$.

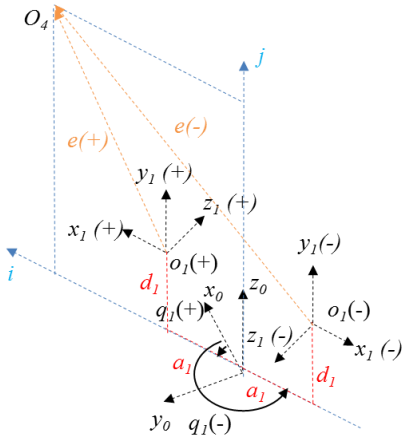


Figure 3. Coordinate frames 1(+) and 1(-).

Based on the geometrical structure of the robot manipulator, q_1(+) exists if and only if $|c - a_2| \le ||{}^{ij}\mathbf{e}(+)|| \le |c + a_2|$ and q_1(-) exists if and only if $|c - a_2| \le ||{}^{ij}\mathbf{e}(-)|| \le |c + a_2|$. If these conditions are met, the solutions of joint *1* are the following equations:

$$q_1(+) = \text{atan2}({}^0o_{4y}, {}^0o_{4x}) \quad (2)$$

$$q_1(-) = q_1(+) + \pi \quad (3)$$

Note that if the origin of coordinate frames *4* or *5* is on the **z** axis of the coordinate frame 0, there are infinite solutions.

There are two sets of solutions for joints *2* and *3* for each joint *1* solution. They are q_2(++) & q_3(++), q_2(+-) & q_3(+-), q_2(-+) & q_3(-+), and q_2(--) & q_3(--). q_1(+) has connections to the first two solution sets and q_1(-) has connections to the last two solution sets.

The values of the angles inside Fig. 4 are explained by the following equations: $b_1 = \text{atan}(d_4/a_3)$ where $0 \le b_1 \le \pi/2$, $b_4(+) = \text{atan2}({}^{ij}e_j(+), {}^{ij}e_i(+))$, $b_2(+) = \text{acos}((c^2 + (a_2)^2 - ||{}^{ij}\mathbf{e}(+)||^2)/(2ca_2))$ where $0 \le b_2(+) \le \pi$, and $b_3(+) = \text{acos}((||{}^{ij}\mathbf{e}(+)||^2 + (a_2)^2 - c^2)/(2||{}^{ij}\mathbf{e}(+)||a_2))$ where $0 \le b_3(+) \le \pi$. The solutions of q_2(++), q_3(++), q_2(+-), and q_3(+-) are expressed as follows,

Copyright © 2016 International Journal of Automation and Smart Technology

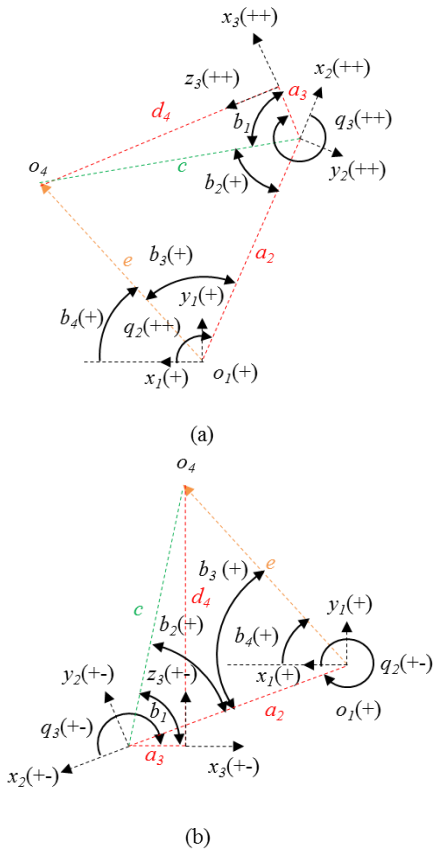


Figure 4. Joint 2 and 3 solutions for q1(+): (a) q2(++) and q3(++); (b) q2(+-) and q3(+-).

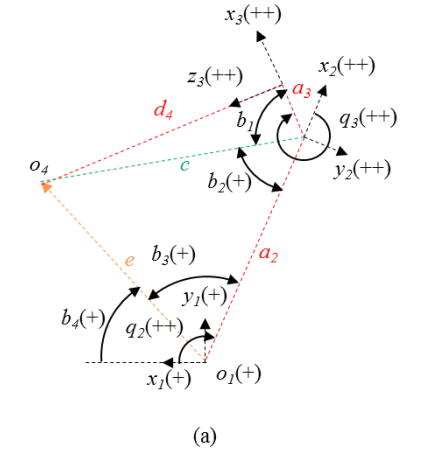
$$q_2(++) = b_3(+) + b_4(+) \quad (4)$$

$$q_3(++) = \pi + b_1 + b_2(+) \quad (5)$$

$$q_2(+-) = b_4(+) - b_3(+) \quad (6)$$

$$q_3(+-) = \pi + b_1 - b_2(+) \quad (7)$$

The values of the angles inside Fig. 5 are explained by the following equations: $b_4(-) = \text{atan2}({}^{ij}e_j(-), -{}^{ij}e_i(-))$, $b_2(-) = \text{acos}((c^2 + (a_2)^2 - ||{}^{ij}\mathbf{e}(-)||^2)/(2ca_2))$ where $0 \le b_2(-) \le \pi$, and $b_3(-) = \text{acos}((||{}^{ij}\mathbf{e}(-)||^2 + (a_2)^2 - c^2)/(2||{}^{ij}\mathbf{e}(-)||a_2))$ where $0 \le b_3(-) \le \pi$. The solutions of $q_2(-+)$, $q_3(-+)$, $q_2(--)$, and $q_3(--)$ are expressed as follows,



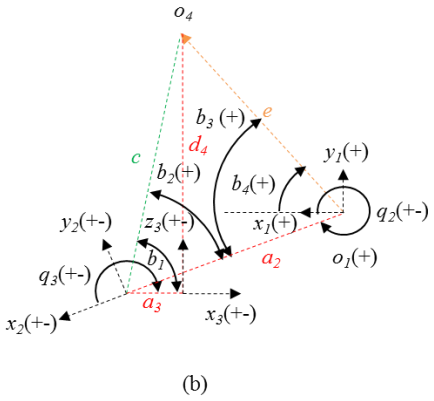


Figure 5. Joint 2 and joint 3 solutions for q1(-): (a) q2(-+) and q3(-+) ; (b) q2(--) and q3(--).

$$q2(-+) = b3(-) + b4(-) \quad (8)$$

$$q3(-+) = \pi + b1 + b2(-) \quad (9)$$

$$q2(--) = b4(-) - b3(-) \quad (10)$$

$$q3(--) = \pi + b1 - b2(-) \quad (11)$$

There are two solution sets for joints *4*, *5*, and *6* for each set of solutions of joints *2* and *3*. We can write that a set of solutions of $q_2(**)$ & $q_3(**)$ is connected to a set of $q_4(**+)$, $q_5(**+)$, & $q_6(**+)$ and a set of $q_4(**-)$, $q_5(**-)$, & $q_6(**-)$ where * can be + or -. Up to this point, four sets of solutions of joints *1*, *2*, and *3* have been obtained. Using forward kinematics, the pose of coordinate frame *3* can be computed.

Based on the geometrical structure of the robot manipulator as shown in Fig. 1, we can write the following equations: ${}^0\mathbf{z}_5 = {}^0\mathbf{z}_6$, ${}^0\mathbf{y}_4(**) = -{}^0\mathbf{z}_3(**)$, and ${}^0\mathbf{y}_4(**)\bullet(-{}^0\mathbf{z}_5) = {}^0\mathbf{x}_4(***)\bullet{}^0\mathbf{x}_5(***)$. Then, the solution of joint *5* can be written as, where $0 \le q5(**+) \le \pi$.

$$q5(**+) = \text{acos}(0y4(**)\bullet(-0z5)) \quad (12)$$

$$q5(**-) = -q5(**+) \quad (13)$$

Based on the geometrical structure of the robot manipulator as shown in Fig. 1, we can write the following equations:
${}^0\mathbf{z}_4(**+) = ({}^0\mathbf{y}_4(**) \times (-{}^0\mathbf{z}_5))/||{}^0\mathbf{y}_4(**) \times (-{}^0\mathbf{z}_5)||$,
${}^0\mathbf{z}_4(**-) = -{}^0\mathbf{z}_4(**+) = ((-{}^0\mathbf{z}_5) \times {}^0\mathbf{y}_4(**))/||(-{}^0\mathbf{z}_5) \times {}^0\mathbf{y}_4(**)||$, ${}^0\mathbf{x}_3(**)\bullet{}^0\mathbf{x}_4(***) = {}^0\mathbf{y}_3(**)\bullet{}^0\mathbf{z}_4(***)$, and ${}^0\mathbf{y}_3(**)\bullet{}^0\mathbf{x}_4(***) = -{}^0\mathbf{x}_3(**)\bullet{}^0\mathbf{z}_4(***)$. Then, the solution of joint *4* can be written as,

$$q_4(***) = \text{atan2}((-{}^0\mathbf{x}_3(**)\bullet{}^0\mathbf{z}_4(***)),$$
$$(0y3(**)\bullet0z4(***))) \quad (14)$$

Based on the geometrical structure of the robot manipulator as shown in Fig. 1, we can write the following equations: ${}^0\mathbf{x}_5(***)\bullet{}^0\mathbf{x}_6 = {}^0\mathbf{z}_4(***)\bullet{}^0\mathbf{y}_6$ and ${}^0\mathbf{y}_5(***)\bullet{}^0\mathbf{x}_6 = {}^0\mathbf{z}_4(***)\bullet{}^0\mathbf{x}_6$. Then, the solution of joint *6* can be written as,

$$q6(***) = \text{atan2}((0z4(***)\bullet0x6),$$
$$(0z4(***)\bullet0y6)) \quad (15)$$

Copyright © 2016 International Journal of Automation and Smart Technology

Note that this paper avoids joint positions in which a singularity occurs.

Jacobian Matrix and Its Time-Derivative Matrices

In addition to the desired joint angles, the controller in this paper also needs the desired joint velocities, accelerations, and jerks. To convert desired end-effector velocity, acceleration, and jerk from operational space to joint velocities, accelerations, and jerks, a Jacobian matrix is used along with its first and second time-derivative matrixes. The relationship of velocity, acceleration, and jerk in operational space and joint space can be described by following equations:

$$\begin{bmatrix} {}^0\dot{\mathbf{p}}_E \\ {}^0\dot{\boldsymbol{\theta}}_E \end{bmatrix} = \mathbf{J}_E(\mathbf{q})\dot{\mathbf{q}} \tag{16}$$

$$\begin{bmatrix} {}^0\ddot{\mathbf{p}}_E \\ {}^0\ddot{\boldsymbol{\theta}}_E \end{bmatrix} = \dot{\mathbf{J}}_E(\mathbf{q})\dot{\mathbf{q}} + \mathbf{J}_E(\mathbf{q})\ddot{\mathbf{q}} \tag{17}$$

$$\begin{bmatrix} {}^0\dddot{\mathbf{p}}_E \\ {}^0\dddot{\boldsymbol{\theta}}_E \end{bmatrix} = \ddot{\mathbf{J}}_E(\mathbf{q})\dot{\mathbf{q}} + 2\dot{\mathbf{J}}_E(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{J}_E(\mathbf{q})\dddot{\mathbf{q}} \tag{18}$$

where ${}^0\dot{\mathbf{p}}_E$ is translational velocity of the end-effector, ${}^0\dot{\boldsymbol{\theta}}_E$ is rotational velocity of the end-effector, $\mathbf{J}_E(\mathbf{q})$ is a Jacobian matrix of the end-effector, $\mathbf{q}$ is the angular position of the joints, $\dot{\mathbf{p}}_E, \dot{\boldsymbol{\theta}}_E, \ddot{\mathbf{p}}_E, \ddot{\boldsymbol{\theta}}_E, \dddot{\mathbf{p}}_E, \dddot{\boldsymbol{\theta}}_E \in \Re^3$, $\mathbf{J}_E(\mathbf{q}), \dot{\mathbf{J}}_E(\mathbf{q}), \ddot{\mathbf{J}}_E(\mathbf{q}) \in \Re^{6\times n}$, and $\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \dddot{\mathbf{q}} \in \Re^n$. In this paper, *n* equals six.

The Jacobian matrix of the end-effector can be expanded as follows,

$$\mathbf{J}_E(\mathbf{q}) = \begin{bmatrix} \mathbf{J}_{Ev}(\mathbf{q}) \\ \mathbf{J}_{E\omega}(\mathbf{q}) \end{bmatrix} \tag{19}$$

where $\mathbf{J}_{Ev}(\mathbf{q})$ is the translational velocity of the end-effector, $\mathbf{J}_{E\omega}(\mathbf{q})$ is the rotational velocity of the end-effector, and $\mathbf{J}_{Ev}(\mathbf{q}), \mathbf{J}_{E\omega}(\mathbf{q}) \in \Re^{3\times n}$. Both can be described as,

$$\mathbf{J}_{Ev}(\mathbf{q}) = \begin{bmatrix} {}^0\mathbf{z}_0 \times \left({}^0\mathbf{t}_n - {}^0\mathbf{t}_0\right) & {}^0\mathbf{z}_1 \times \left({}^0\mathbf{t}_n - {}^0\mathbf{t}_1\right) & \cdots & {}^0\mathbf{z}_{n-1} \times \left({}^0\mathbf{t}_n - {}^0\mathbf{t}_{n-1}\right) \end{bmatrix} \tag{20}$$

$$\mathbf{J}_{E\omega}(\mathbf{q}) = \begin{bmatrix} {}^0\mathbf{z}_0 & {}^0\mathbf{z}_1 & \cdots & {}^0\mathbf{z}_{n-1} \end{bmatrix} \tag{21}$$

$\dot{\mathbf{J}}_E(\mathbf{q})$ and $\ddot{\mathbf{J}}_E(\mathbf{q})$ can be described as,

$$\dot{\mathbf{J}}_E(\mathbf{q}) = \sum_{i=1}^{n}\left(\frac{\partial \mathbf{J}_E(\mathbf{q})}{\partial q_i}\dot{q}_i\right) \tag{22}$$

$$\ddot{\mathbf{J}}_E(\mathbf{q}) = \sum_{i=1}^{n}\left(\sum_{j=1}^{n}\left(\frac{\partial\left(\frac{\partial \mathbf{J}_E(\mathbf{q})}{\partial q_i}\right)}{\partial q_j}\dot{q}_j\right)\dot{q}_i + \frac{\partial \mathbf{J}_E(\mathbf{q})}{\partial q_i}\ddot{q}_i\right) \tag{23}$$

Note that this paper avoids joint positions in which a singularity occurs. Thus, $\mathbf{J}_E$ is always invertible.

Dynamics of the RLED Robot Manipulator

The dynamics of the robot manipulator and DC motor can be expressed as follows:

$$\mathbf{K}_t\mathbf{i} = \boldsymbol{\tau} = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) + \mathbf{D}\dot{\mathbf{q}} + \mathbf{w}_R \tag{24}$$

$$\mathbf{v} = \mathbf{R}_a\mathbf{i} + \mathbf{L}\dot{\mathbf{i}} + \mathbf{K}_v\dot{\mathbf{q}} + \mathbf{w}_A \tag{25}$$

where $\mathbf{K}_t$ is a torque constant matrix, $\mathbf{i}$ is the armature current vector, $\boldsymbol{\tau}$ is the torque vector, $\mathbf{M}(\mathbf{q})$ is the inertia matrix of the robot manipulator, $\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})$ is the centrifugal and Coriolis forces matrix, $\mathbf{g}(\mathbf{q})$ is the gravitational forces vector, $\mathbf{D}$ is the viscous friction coefficient matrix, $\mathbf{w}_R$ is the robot manipulator external disturbance vector, $\mathbf{v}$ is the applied armature voltage vector, $\mathbf{R}_a$ is the armature resistance matrix, $\mathbf{L}$ is the armature inductance matrix, $\mathbf{K}_v$ is the back electromotive force constant matrix, $\mathbf{w}_A$ is the DC motor external disturbance matrix, and $\mathbf{i},\dot{\mathbf{i}},\boldsymbol{\tau},\mathbf{v},\mathbf{w}_R,\mathbf{w}_A,\mathbf{g}(\mathbf{q}) \in \Re^n$, $\mathbf{K}_t,\mathbf{M}(\mathbf{q}),\mathbf{C}(\mathbf{q},\dot{\mathbf{q}}),\mathbf{D},\mathbf{R}_A,\mathbf{L},\mathbf{K}_v \in \Re^{n\times n}$.

$\mathbf{M}(\mathbf{q})$ is a positive definite matrix. $\mathbf{K}_t,\mathbf{D},\mathbf{R}_A,\mathbf{L},\mathbf{K}_v$ are diagonal matrices with positive elements.

The result of substituting (24) into (25) can be rearranged to yield

$$\dddot{\mathbf{q}} = \mathbf{f}_3 + \mathbf{w}_3 + \mathbf{B}_3\mathbf{v} \tag{26}$$

where,

$$\begin{aligned}\mathbf{f}_3 = -\mathbf{M}^{-1}\Big(&\dot{\mathbf{M}}\ddot{\mathbf{q}} + \dot{\mathbf{C}}\dot{\mathbf{q}} + \mathbf{C}\ddot{\mathbf{q}} + \dot{\mathbf{g}} + \mathbf{D}\ddot{\mathbf{q}} + \mathbf{K}_t\mathbf{K}_v\mathbf{L}^{-1}\dot{\mathbf{q}} \\ &+ \mathbf{R}_a\mathbf{L}^{-1}\left(\mathbf{M}\ddot{\mathbf{q}} + \mathbf{C}\dot{\mathbf{q}} + \mathbf{g} + \mathbf{D}\dot{\mathbf{q}}\right)\Big)\end{aligned} \tag{27}$$

$$\mathbf{w}_3 = -\mathbf{M}^{-1}\left(\dot{\mathbf{w}}_R + \mathbf{R}_a\mathbf{L}^{-1}\mathbf{w}_R + \mathbf{K}_t\mathbf{L}^{-1}\mathbf{w}_A\right) \tag{28}$$

$$\mathbf{B}_3 = \mathbf{M}^{-1}\mathbf{K}_t\mathbf{L}^{-1} \tag{29}$$

(26) is the dynamics model of the RLED robot manipulators. Define $\hat{\mathbf{f}}_3$ as the known model of $\mathbf{f}_3$, $\hat{\mathbf{B}}_3$ as the known model of $\mathbf{B}_3$, $\Delta\mathbf{f}_3$ as the uncertainty of $\mathbf{f}_3$, $\Delta\mathbf{B}_3$ as the uncertainty of $\mathbf{B}_3$, $\mathbf{f}_3 = \hat{\mathbf{f}}_3 + \Delta\mathbf{f}_3$, and $\mathbf{B}_3 = \hat{\mathbf{B}}_3 + \Delta\mathbf{B}_3$. Because $\mathbf{M},\mathbf{K}_t,\mathbf{L}$ are always invertible, $\mathbf{B}_3$ is also always invertible. By assuming that $\hat{\mathbf{M}},\hat{\mathbf{K}}_t,\hat{\mathbf{L}}$ are invertible, $\hat{\mathbf{B}}_3$ is also always invertible. Rewriting (26) results in

$$\dddot{\mathbf{q}} = \left(\hat{\mathbf{f}}_3 + \Delta\mathbf{f}_3\right) + \mathbf{w}_3 + \left(\hat{\mathbf{B}}_3 + \Delta\mathbf{B}_3\right)\mathbf{v} \tag{30}$$

Several assumptions regarding the uncertainties and external disturbances are made:

$$\left\|\Delta\mathbf{f}_3\right\|_\infty \le \xi$$

$$\left\|\mathbf{w}_3\right\|_\infty \le \varsigma$$

$$\left\|\Delta\mathbf{B}_3\hat{\mathbf{B}}_3^{-1}\right\|_\infty < \delta < 1$$

Copyright © 2016 International Journal of Automation and Smart Technology

Control System

Sliding Mode Control

The sliding surface function is defined as follows:

$$\mathbf{s} = \left(\frac{d}{dt} + \lambda\right)^{m-1} \mathbf{e} \tag{31}$$

$$\mathbf{e} = \mathbf{q} - \mathbf{q}_d \tag{32}$$

where m is the system order, which is 3 in this case, λ is a positive constant, $\mathbf{q}_d$ is a vector of the desired angle of joints, and $\mathbf{q}_d \in \Re^n$. Note that it is possible to have different values of λ_i for each s_i, where s_i is element i of **s**. Rewriting (25) with $m = 3$ yields

$$\mathbf{s} = \ddot{\mathbf{e}} + 2\lambda\dot{\mathbf{e}} + \lambda^2\mathbf{e} \tag{33}$$

Let the control input be

$$\mathbf{v} = \hat{\mathbf{B}}_3^{-1}\left(-\hat{\mathbf{f}}_3 + \ddot{\mathbf{q}}_d - 2\lambda\ddot{\mathbf{e}} - \lambda^2\dot{\mathbf{e}} - \beta\mathbf{sgn}(\mathbf{s})\right) \tag{34}$$

$$\mathrm{sgn}(s_i) = \begin{cases} 0 & if\ s_i = 0 \\ -1 & if\ s_i < 0 \\ 1 & if\ s_i > 0 \end{cases} \tag{35}$$

where β is a positive scalar, which is defined later, and $\mathbf{sgn}(\mathbf{s}) = [sgn(s_0)\ sgn(s_1)\ \cdots\ sgn(s_n)]^{\mathrm{T}}$.

Taking the time derivative of (33) results in

$$\begin{aligned} \dot{\mathbf{s}} &= \dddot{\mathbf{e}} + 2\lambda\ddot{\mathbf{e}} + \lambda^2\dot{\mathbf{e}} \\ &= \Delta\mathbf{f}_3 + \mathbf{w}_3 + \Delta\mathbf{B}_3\hat{\mathbf{B}}_3^{-1}\left(-\hat{\mathbf{f}}_3 + \dddot{\mathbf{q}}_d - 2\lambda\ddot{\mathbf{e}} - \lambda^2\dot{\mathbf{e}}\right) \\ &\quad - \left(\mathbf{I} + \Delta\mathbf{B}_3\hat{\mathbf{B}}_3^{-1}\right)\beta\mathbf{sgn}(\mathbf{s}) \end{aligned} \tag{36}$$

The Lyapunov stability criterion is used to prove stability. We define a well-known Lyapunov function for system with SMC as follows,

$$V = \frac{1}{2}\mathbf{s}^T\mathbf{s} \tag{37}$$

Taking the time derivative of (37) yields,

$$\begin{aligned} \dot{V} &= \mathbf{s}^T\dot{\mathbf{s}} \\ &\le \left|\mathbf{s}^T\right|\left|\Delta\mathbf{f}_3\right| + \left|\mathbf{s}^T\right|\left|\mathbf{w}_3\right| + \left|\mathbf{s}^T\right|\left|\Delta\mathbf{B}_3\hat{\mathbf{B}}_3^{-1}\right|\left(-\hat{\mathbf{f}}_3 + \dddot{\mathbf{q}}_d - 2\lambda\ddot{\mathbf{e}} - \lambda^2\dot{\mathbf{e}}\right) \\ &\quad - \mathbf{s}^T\left(\mathbf{I} - \left|\Delta\mathbf{B}_3\hat{\mathbf{B}}_3^{-1}\right|\right)\beta\mathbf{sgn}(\mathbf{s}) \end{aligned} \tag{38}$$

For simplicity, b_{ij} is defined as element i-j of matrix $\Delta\mathbf{B}_3\hat{\mathbf{B}}_3^{-1}$ and the last term of (38) is analyzed separately.

$$\begin{aligned} &-\mathbf{s}^T\left(\mathbf{I} - \left|\Delta\mathbf{B}_3\hat{\mathbf{B}}_3^{-1}\right|\right)\beta\mathbf{sgn}(\mathbf{s}) \\ &= -\mathbf{s}^T\begin{bmatrix} 1-|b_{11}| & -|b_{12}| & \cdots & -|b_{1n}| \\ -|b_{21}| & 1-|b_{22}| & \cdots & -|b_{2n}| \\ \vdots & \vdots & \ddots & \vdots \\ -|b_{n1}| & -|b_{n2}| & \cdots & 1-|b_{nn}| \end{bmatrix}\beta\mathbf{sgn}(\mathbf{s}) \\ &= -\mathbf{s}^T\begin{bmatrix} (1-|b_{11}|)\mathrm{sgn}(s_1) - |b_{12}|\mathrm{sgn}(s_2) - \cdots - |b_{1n}|\mathrm{sgn}(s_n) \\ -|b_{21}|\mathrm{sgn}(s_1) + (1-|b_{22}|)\mathrm{sgn}(s_2) - \cdots - |b_{2n}|\mathrm{sgn}(s_n) \\ \vdots \\ -|b_{n1}|\mathrm{sgn}(s_1) - |b_{n2}|\mathrm{sgn}(s_2) - \cdots + (1-|b_{nn}|)\mathrm{sgn}(s_n) \end{bmatrix}\beta \\ &\le -\Big[(1-|b_{11}|)|s_1| - |b_{12}||s_1| - \cdots - |b_{1n}||s_1| \\ &\quad - |b_{21}||s_2| + (1-|b_{22}|)|s_2| - \cdots - |b_{2n}||s_2| \\ &\quad + \cdots \\ &\quad - |b_{n1}||s_n| - |b_{n2}||s_n| - \cdots + (1-|b_{nn}|)|s_n|\Big]\beta \\ &\le -\left[(1-\delta)|s_1| + (1-\delta)|s_2| + \cdots + (1-\delta)|s_n|\right]\beta \\ &= -(1-\delta)\beta\|\mathbf{s}\|_1 \end{aligned} \tag{39}$$

Let define β as follows:

$$\beta = \frac{\varsigma + \xi + \delta\left\|\left(-\hat{\mathbf{f}}_3 + \dddot{\mathbf{q}}_d - 2\lambda\ddot{\mathbf{e}} - \lambda^2\dot{\mathbf{e}}\right)\right\|_\infty + \eta}{1-\delta} \tag{40}$$

where η is a positive constant. By substituting (39) and (40) into (38), we can obtain,

$$\begin{aligned} \dot{V} &= \mathbf{s}^T\dot{\mathbf{s}} \le -\eta\|\mathbf{s}\|_1 \le -\eta\|\mathbf{s}\|_2 \\ &= -\eta\sqrt{2}V^{0.5} < 0,\ for \mathbf{s} \ne 0 \end{aligned} \tag{41}$$

Using the Lyapunov stability criterion, (37) and (41) prove that sliding surface **s** converges to zero in finite time [23] despite the presence of system uncertainties including an uncertain input matrix and external disturbances. Thus, after condition **s = 0** is reached in finite time, **e** is guaranteed to converge to zero asymptotically.

The major drawback of SMC is the chattering in the control signals caused by the discontinuous term **sgn**(**s**) of the controller. To overcome this problem, a well-known method is to smooth out the controller discontinuity in a thin boundary layer neighboring the sliding surface [24]. Thus **sgn**(**s**) is replaced with **sat**(**s**/Φ), where Φ is the boundary layer width. This function can be expressed as follows,

$$\mathrm{sat}(s_i/\Phi) = \begin{cases} \mathrm{sgn}(s_i)\ if\ |s_i| \ge \Phi \\ s_i/\Phi \quad if\ |s_i| < \Phi \end{cases} \tag{42}$$

where **sat**(**s**/Φ) = [sat(s_1/Φ) sat(s_2/Φ) … sat(s_n/Φ)].

When s_i(t) ≥ Φ, the controller works using discontinuous function sgn(s_i). Thus, $|s_i(t)|$ reaches Φ in finite time $t_{r\Phi i}$. The smoothing function is used when s_i(t) < Φ and the stability is undetermined. The controller with smoothing only guarantees $|s_i(t)| \le \Phi$ for $t \ge t_{r\Phi i}$. Because all $|s_i(t)| \le \Phi$ for $t \ge t_{r\Phi max}$ and $\mathbf{s} = \ddot{\mathbf{e}} + 2\lambda\dot{\mathbf{e}} + \lambda^2\mathbf{e}$, the smoothing guarantees that the bounds $\|\mathbf{e}\|_\infty \le \Phi/\lambda^2$, $\|\dot{\mathbf{e}}\|_\infty \le 2\Phi/\lambda$, and $\|\ddot{\mathbf{e}}\|_\infty \le 4\Phi$ are obtained asymptotically [24], where $t_{r\Phi max}$ is the maximum value of all $t_{r\Phi i}$.

Copyright © 2016 International Journal of Automation and Smart Technology

Adaptive Sliding Mode Control

The SMC developed in the previous section can achieve stability and robustness for multiple-input/multiple-output nonlinear systems with bounded uncertainties and external disturbances. The disadvantage of the previously designed SMC is that the controller needs information about upper bounds of the uncertainties and the external disturbances, which are difficult to obtain. In this section, ASMC is designed to overcome this problem. ASMC is developed based on SMC which is presented in the previous section. It is basically the same as SMC with β tuned adaptively and without knowledge of the bounds of the uncertainties and the external disturbances.

By rewriting the time derivative of sliding surface, we can obtain,

$$\dot{\mathbf{s}} = \hat{\mathbf{f}}_3 + \hat{\mathbf{B}}_3\mathbf{v} - \ddot{\mathbf{q}}_d + 2\lambda\dot{\mathbf{e}} + \lambda^2\mathbf{e} + \mathbf{h} \tag{43}$$

$$\mathbf{h} = \Delta\mathbf{f}_3 + \mathbf{w}_3 + \Delta\mathbf{B}_3\mathbf{v} \tag{44}$$

The controller in this section is basically a modified version of (34). It is defined as follows,

$$\mathbf{v} = \hat{\mathbf{B}}_3^{-1}\left(-\hat{\mathbf{f}}_3 + \ddot{\mathbf{q}}_d - 2\lambda\dot{\mathbf{e}} - \lambda^2\mathbf{e} - \hat{\beta}\,\mathbf{sgn}(\mathbf{s})\right) \tag{45}$$

where $\hat{\beta}$ is adaptively an adjusted gain positive constant.

Assume there exists a positive constant β_d and a positive constant $\bar{h}$ that satisfy $\infty > \beta_d > \bar{h} \geq ||\mathbf{h}||_\infty$. Adaptation error is defined as $\tilde{\beta} = \beta_d - \hat{\beta}$.

The Lyapunov stability criterion is used to prove the stability, convergence of the adaptive gain, and reaching of the sliding surface.

$$\begin{aligned} V &= V_1 + V_2 \\ &= \frac{1}{2}\mathbf{s}^T\mathbf{s} + \frac{1}{2\gamma}\tilde{\beta}^2 \end{aligned} \tag{46}$$

where V_1 and V_2 are respectively the first and second terms of the right hand side of (46) and γ is a positive constant and also acts as adaptation rate, which is explained later. Taking the time derivative of (46) yields.

$$\begin{aligned} \dot{V} &= \dot{V}_1 + \dot{V}_2 \\ &= \mathbf{s}^T\dot{\mathbf{s}} + \frac{1}{\gamma}\tilde{\beta}\dot{\tilde{\beta}} = -\hat{\beta}\|\mathbf{s}\|_1 + \mathbf{s}^T\mathbf{h} + \frac{1}{\gamma}\tilde{\beta}\left(-\dot{\hat{\beta}}\right) \end{aligned} \tag{47}$$

We define the adaptive law as follows,

$$\dot{\hat{\beta}} = \gamma\|\mathbf{s}\|_1 \tag{48}$$

Substituting (48) to (47), we can obtain,

$$\begin{aligned} \dot{V} &= \mathbf{s}^T\mathbf{h} - \beta_d\|\mathbf{s}\|_1 \\ &\leq -\left(\beta_d - \bar{h}\right)\|\mathbf{s}\|_1 \\ &\leq -\left(\beta_d - \bar{h}\right)\|\mathbf{s}\|_2 \\ &= -\left(\beta_d - \bar{h}\right)\sqrt{2}V_1^{0.5} \leq 0 \end{aligned} \tag{49}$$

(46) and (49) prove that V converges to a limit, thus $\mathbf{s}$ and $\tilde{\beta}$ are bounded. $\mathbf{s}$ must converge to zero, because $\dot{\tilde{\beta}} = -\gamma\|\mathbf{s}\|_1$ and $\tilde{\beta}$ is bounded. $\tilde{\beta}$ must converge to some constant because $\tilde{\beta}$ is bounded and $\dot{\tilde{\beta}} \leq 0$.

Substituting the controller (45) into time-derivative of $\mathbf{s}$ (43) results in $\dot{\mathbf{s}} = -\hat{\beta}\,\mathbf{sgn}(\mathbf{s}) + \mathbf{h}$. The worst uncertainty condition for $\mathbf{s}$ is $\mathbf{h} = \bar{h}\,\mathbf{sgn}(\mathbf{s})$ when $\mathbf{s} \neq 0$ and $\mathbf{h} = -\bar{h}$ or $\mathbf{h} = +\bar{h}$ when $\mathbf{s} = 0$. Substituting the worst uncertainty condition $\mathbf{h}$ when $\mathbf{s} \neq 0$ into $\dot{\mathbf{s}}$ yields $\dot{\mathbf{s}} = \left(-\hat{\beta} + \bar{h}\right)\mathbf{sgn}(\mathbf{s})$. The resulting equation shows that the uncertainty causes the absolute value of $\mathbf{s}$ to increase when $\hat{\beta} < \bar{h}$.

If the worst uncertainty condition always occurs, $\mathbf{s}(t_1) = 0$, and $\hat{\beta}(t_1) < \bar{h}$ then the condition $|\mathbf{s}(t_2)| > 0$ will occur soon after that, where $\infty > t_2 > t_1 \geq 0$. Note that $|\mathbf{s}(t)|$ always increases as long as $\hat{\beta}(t) < \bar{h}$ because $\dot{\mathbf{s}} = \left(-\hat{\beta} + \bar{h}\right)\mathbf{sgn}(\mathbf{s})$ and $\hat{\beta}(t)$ always increases as long as $||\mathbf{s}(t)||_1 \neq 0$ because $\dot{\hat{\beta}}(t) = \gamma\|\mathbf{s}(t)\|_1$.

If the worst uncertainty condition always occurs, $|\mathbf{s}(t_1)| > 0$, and $\hat{\beta}(t_1) \leq \bar{h}$ then the condition $\hat{\beta}(t_2) > \bar{h}$ will be reached in finite time with the rate $\dot{\hat{\beta}}(t) \geq \gamma\|\mathbf{s}(t_1)\|_1$ for $t_2 > t \geq t_1$.

If the worst uncertainty condition always occurs, $\mathbf{s}(t_1) = 0$, and $\hat{\beta}(t_1) \geq \bar{h}$ then $\mathbf{s}(t)$ and $\dot{\hat{\beta}}(t)$ is always zero for $t \geq t_1$.

If the worst uncertainty condition always occurs, $|\mathbf{s}(t_1)| > 0$, and $\hat{\beta}(t_1) > \bar{h}$ then $\mathbf{s}(t)$ will converge to zero and $\hat{\beta}(t)$ will converge to some positive constant in finite time. Note that $\hat{\beta}(t) \geq \hat{\beta}(t_1)$ for $t \geq t_1$, then the finite time convergence of $\mathbf{s}(t)$ and $\hat{\beta}(t)$ when $t \geq t_1$ and $\hat{\beta}(t_1) > \bar{h}$ is proven by the following equation.

$$\begin{aligned} \dot{V}_1 &= \mathbf{s}^T\dot{\mathbf{s}} \\ &= \mathbf{s}^T\left(-\hat{\beta}\,\mathbf{sgn}(\mathbf{s}) + \mathbf{h}\right) \\ &\leq -\left(\hat{\beta} - \bar{h}\right)\|\mathbf{s}\|_1 \\ &\leq -\left(\hat{\beta} - \bar{h}\right)\|\mathbf{s}\|_2 \\ &\leq -\left(\hat{\beta} - \bar{h}\right)\sqrt{2}V_1^{0.5} \\ &\leq -cV_1^{0.5} \leq 0 \end{aligned} \tag{50}$$

After $\hat{\beta}$ converges to some constant and $\mathbf{s}$ converges to zero in finite time, $\mathbf{e}$ is guaranteed to converge to zero asymptotically.

Smoothing using $\mathbf{sat}(\mathbf{s}/\Phi)$ can also be applied to ASMC. However, the update law needs to be modified because the smoothing is only guaranteed to be bounded by $||\mathbf{s}||_\infty \leq \Phi$ in finite time, instead of $\mathbf{s} = 0$. The modification can prevent the value of $\hat{\beta}$ from increasing to infinity. The controller and the update law are then

Copyright © 2016 International Journal of Automation and Smart Technology

modified to

$$\mathbf{v} = \hat{\mathbf{B}}_3^{-1}\left(-\hat{\mathbf{f}}_3 + \ddot{\mathbf{q}}_d - 2\lambda\dot{\mathbf{e}} - \lambda^2\mathbf{e} - \hat{\beta}\mathbf{sat}(\mathbf{s}/\Phi)\right) \quad (51)$$

$$\dot{\hat{\beta}} = \begin{cases} \gamma\|\mathbf{s}\|_1 & if\ \|\mathbf{s}\|_\infty > \Phi \\ 0 & if\ \|\mathbf{s}\|_\infty \le \Phi \end{cases} \quad (52)$$

By assuming that $\dddot{\mathbf{q}}_d$ is bounded, **h** (44) and controller (45) can be redefined as,

$$\mathbf{h} = \Delta\mathbf{f}_3 + \mathbf{w}_3 + \Delta\mathbf{B}_3\mathbf{v} - \dddot{\mathbf{q}}_d \quad (53)$$

$$\mathbf{v} = \hat{\mathbf{B}}_3^{-1}\left(-\hat{\mathbf{f}}_3 - 2\lambda\dot{\mathbf{e}} - \lambda^2\mathbf{e} - \hat{\beta}\mathbf{sgn}(\mathbf{s})\right) \quad (54)$$

Note that unlike (45), (54) does not have $\dddot{\mathbf{q}}_d$. The update law is the same as for (48). By applying smoothing to the controller, the controller can be written as,

$$\mathbf{v} = \hat{\mathbf{B}}_3^{-1}\left(-\hat{\mathbf{f}}_3 - 2\lambda\dot{\mathbf{e}} - \lambda^2\mathbf{e} - \hat{\beta}\mathbf{sat}(\mathbf{s}/\Phi)\right) \quad (55)$$

with the same update law as for (52).

Simulations

Performance was evaluated by simulation models. Figure 6 shows the structure of the overall system. We can see that the plant consists of a robot manipulator dynamics and its actuators dynamics, which are electric DC motors. The inputs of the plant are voltage signals **v**(*t*) and the outputs are joint position signals $\mathbf{q}(t)$, joint velocity signals $\dot{\mathbf{q}}(t)$, and joint acceleration signals $\ddot{\mathbf{q}}(t)$. This paper seeks to design a controller so that the end-effector pose of the robot manipulator ${}^0\mathbf{Z}_E(t)$ follows the desired end-effector pose ${}^0\mathbf{Z}_{E\text{-}desired}(t)$.

Table 2 summarizes the parameters of the RLED robot manipulator for simulation. M_{Mi} is moment of inertia of DC motor at joint *i*.

The central of mass of the link is assumed to be located in the middle of each link. Their positions can be expressed as follows: ${}^1\mathbf{p}_{C1} = [-0.5a_1\ 0\ 0]^T$, ${}^2\mathbf{p}_{C2} = [-0.5a_2\ 0\ 0]^T$, ${}^3\mathbf{p}_{C3} = [-0.5a_3\ 0\ 0]^T$, ${}^4\mathbf{p}_{C4} = [0\ 0.5d_4\ 0]^T$, ${}^5\mathbf{p}_{C5} = [0\ 0\ 0]^T$, and ${}^6\mathbf{p}_{C6} = [0\ 0\ -0.5d_6]^T$.

The mass density of the links is assumed to be uniform and the form of the links is assumed to be a solid cylinder with a radius *r* and height *h*. It is assumed that the *r* values corresponding to links *1* to *6* are 0.07, 0.06, 0.04, 0.04, 0, and 0.04. Then the inertia tensor of each link can be written as

$${}^1\mathbf{I}_1 = \begin{bmatrix} m_1\left(3r_1^2 + d_1^2\right)/12 & 0 & 0 \\ 0 & m_1 r_1^2/2 & 0 \\ 0 & 0 & m_1\left(3r_1^2 + d_1^2\right)/12 \end{bmatrix} \quad (56)$$

$${}^2\mathbf{I}_2 = \begin{bmatrix} m_2 r_2^2/2 & 0 & 0 \\ 0 & m_2\left(3r_2^2 + a_2^2\right)/12 & 0 \\ 0 & 0 & m_2\left(3r_2^2 + a_2^2\right)/12 \end{bmatrix} \quad (57)$$

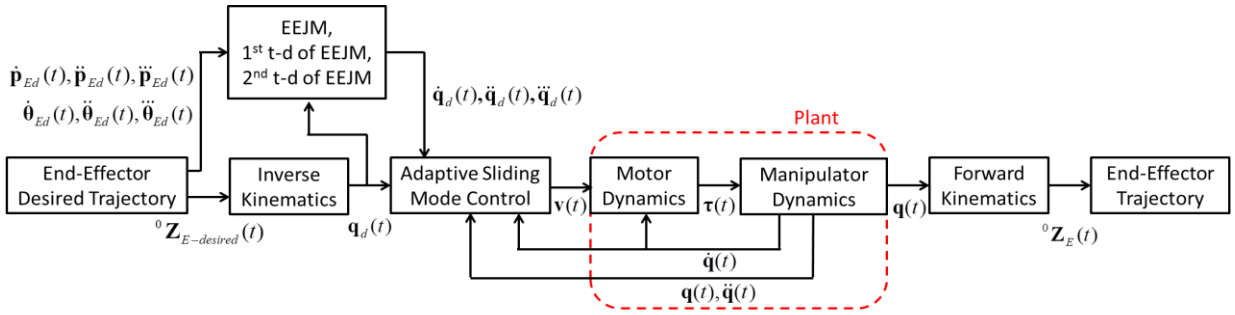
$${}^3\mathbf{I}_3 = \begin{bmatrix} m_3 r_3^2/2 & 0 & 0 \\ 0 & m_3\left(3r_3^2 + a_3^2\right)/12 & 0 \\ 0 & 0 & m_3\left(3r_3^2 + a_3^2\right)/12 \end{bmatrix} \quad (58)$$

$${}^4\mathbf{I}_4 = \begin{bmatrix} m_4\left(3r_4^2 + d_4^2\right)/12 & 0 & 0 \\ 0 & m_4 r_4^2/2 & 0 \\ 0 & 0 & m_4\left(3r_4^2 + d_4^2\right)/12 \end{bmatrix} \quad (59)$$

$${}^5\mathbf{I}_5 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (60)$$

$${}^6\mathbf{I}_6 = \begin{bmatrix} m_6\left(3r_6^2 + d_6^2\right)/12 & 0 & 0 \\ 0 & m_6\left(3r_6^2 + d_6^2\right)/12 & 0 \\ 0 & 0 & m_6 r_6^2/2 \end{bmatrix} \quad (61)$$

This simulation uses controller (55) with update law (52). The controller parameters are set as λ = 10 and Φ = 0.01. The adaptation rate is set as γ = 10000. Using controller (55), $\dddot{\mathbf{q}}_d$ need not be defined, and thus there is no need to define end-effector translational jerk $\dddot{\mathbf{p}}_{Ed}$



Note :
EEJM = end-effector Jacobian matrix
t-d = time-derivative

Figure 6. Block diagram of the overall RLED robot manipulator.

Copyright © 2016 International Journal of Automation and Smart Technology

i	d_i (m)	a_i (m)	α_i (degree)	m_i (kg)	M_{Mi} (kg/m^2)	R_i (Ω)	L_i (mH)	K_{ti} (N.m/A)	K_{vi} (v/rpm)	Max. Voltage (v)
1	0.201	0	90^0	4	1.7	2	1.5	33	3.8	100
2	0	0.293	0^0	4	1.7	2	1.5	33	3.8	100
3	0	0.085	90^0	3	0.2	1.4	1.8	15	1.7	100
4	0.463	0	-90^0	2	0.2	0.07	0.06	2.9	0.34	24
5	0	0	90^0	0	0.2	0.07	0.06	2.9	0.34	24
6	0.293	0	0^0	1.5	0.2	0.07	0.06	2.9	0.34	24

Table 2. Parameters of the rled robot manipulator.

and rotational jerk $\dddot{\boldsymbol{\theta}}_{Ed}$ or to compute the end-effector second time-derivative of the Jacobian matrix $\ddot{\mathbf{J}}_E$. This greatly reduces the computation time for the controller and for generating the controller reference signals. In addition, the controller's saturation function can prevent the chattering effect in the control signals.

In this simulation, the following conditions hold:

- The modeled dynamics are $\hat{\mathbf{f}}_3 = -\mathbf{M}^{-1}\mathbf{K}_t\mathbf{K}_v\mathbf{L}^{-1}\dot{\mathbf{q}}$ and $\hat{\mathbf{B}}_3 = \mathbf{B}_3\left(1+0.4\sin(10t)\right)$.
- The joint friction matrix is unknown and is set as,

$$\mathbf{D} = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \text{ in } \frac{N.m}{rad/s} \quad (62)$$

- The external disturbances are unknown and they are set as,

$$\mathbf{w}_R = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}^T \cdot \sin(10t) \text{ in } N \cdot m \quad (63)$$

$$\mathbf{w}_A = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}^T \cdot \sin(5t) \text{ in } volt \quad (64)$$

- Initial value of the robust gain $\hat{\beta}(0)$ is set at 0.
- The initial joint velocities are $\dot{\mathbf{q}}(0)$ = [0 0 0 0 0 0]^T, and the initial motor currents are $\mathbf{i}(0)$ = [0 0 0 0 0 0]^T. The initial joint positions are obtained from inverse kinematics solution *2* (see Fig. 2) of the following end-effector pose.

$$\mathbf{E}(0) = {}^0\mathbf{Z}_6(0) = \begin{bmatrix} 1 & 0 & 0 & 0.2 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (65)$$

- Only solution 2 of the inverse kinematics is used for converting the desired end-effector position and orientation from operational space to joint space.
- The motion law of the desired end-effector trajectory makes use of the fifth-order polynomial function. The initial and final values of the desired velocity and acceleration of the end-effector are zero.

Simulation results show that the RLED robot manipulator can follow the desired trajectory, despite large initial errors (see Figs. 7-9). This is because the value of the robust gain $\hat{\beta}$ is still not large enough to encounter the uncertainties. Figure 7 shows that the joint position errors move towards the guaranteed bound of $||\mathbf{e}||_\infty \le \Phi/\lambda^2 = 0.0001$ rad. Figure 8 shows that the end-effector position errors move towards some guaranteed bound that from the graph it is about 0.01 mm. To present the end-effector orientation error, the norm-2 of x, y, and z axis errors is used. For example, at time *t* the orientation error is presented by $||{}^0\mathbf{x}_E(t)-{}^0\mathbf{x}_{Ed}(t)||_2$, $||{}^0\mathbf{y}_E(t)-{}^0\mathbf{y}_{Ed}(t)||_2$, and $||{}^0\mathbf{z}_E(t)-{}^0\mathbf{z}_{Ed}(t)||_2$. Figure 9 shows that the end-effector orientation errors move towards some guaranteed bound that from the graph it is about 0.0004.

Figure 10 shows that around t = 1 the sliding function signals reached the guaranteed bound $||\mathbf{s}||_\infty \le \Phi = 0.01$ and the values are always inside the boundary thereafter. At the same time, the robust gain $\hat{\beta}$ also converges to some positive constant as shown in Fig. 11.

Figure 12 shows that there is no chattering in the control signal because the controller uses the saturation function instead of the sign function.

Figure 13 shows the motion of the RLED robot manipulator in operational space. First, the end-effector position moves in a straight line and simultaneously changes its orientation. Second, the end-effector position moves in a circle and simultaneously changes its orientation. Third, the end-effector position moves in a straight line and simultaneously changes its orientation with a different trajectory when compared to the first trajectory.

Conclusion and Future Works

This paper proposes ASMC for solving end-effector position and orientation tracking problems of a 6 DOF RLED revolute joint serial robot manipulator with uncertain parameters and external disturbances. The end-effector position and orientation tracking problem is converted to a joint positions tracking problem, because

Copyright © 2016 International Journal of Automation and Smart Technology

ASMC works in joint space. The desired end-effector position and orientation, which are the outputs of points in operational space, is converted to the desired joint positions using inverse kinematics. The desired end-effector translational and rotational velocities, accelerations, and jerks are converted to joint velocities, accelerations, and jerks using a Jacobian matrix and its 1st and 2nd time-derivatives because they are also required in ASMC. Theoretical analysis using the Lyapunov stability criterion shows that ASMC can solve the joint position tracking problem without regression matrices, knowledge of the uncertainty upper limits, or defining desired jerk signals. Simulation results verify the performance of the inverse kinematics and the ASMC. Note that the uncertain parameters in this paper are not kinematic parameters. If the kinematic parameters are uncertain, the end-effector pose tracking problem and joint position tracking problem conversion, which is used in this paper, cannot be applied. In the future, uncertainties from external disturbances will use random noises to reflect practical situations. Meanwhile, the proposed approaches will be used to control a real 6 degrees of freedom manipulator to verify its practical effectiveness and performance.

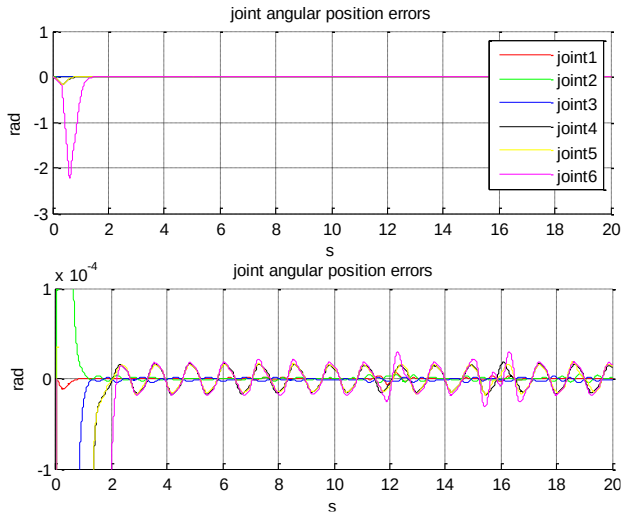


Figure 7. Joint angular position errors.

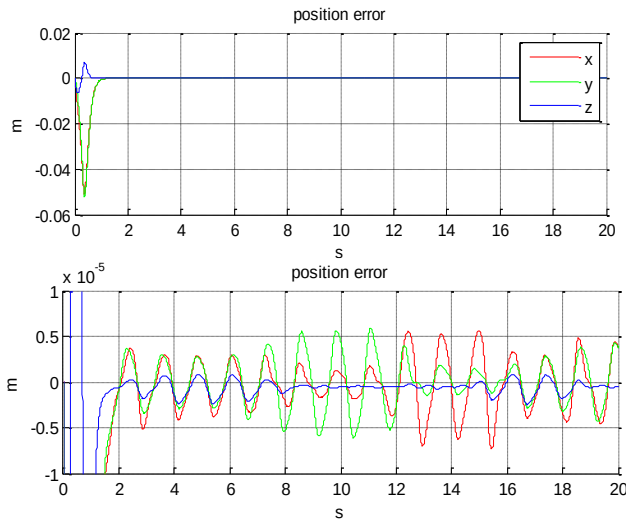


Figure 8. End-effector position errors.

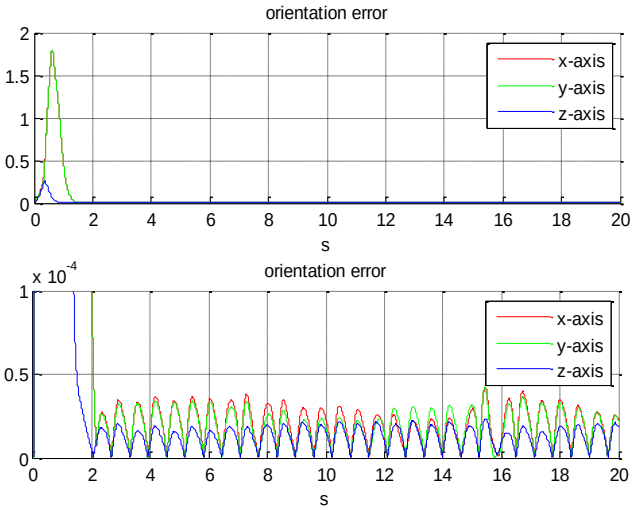


Figure 9. End-effector orientation errors.

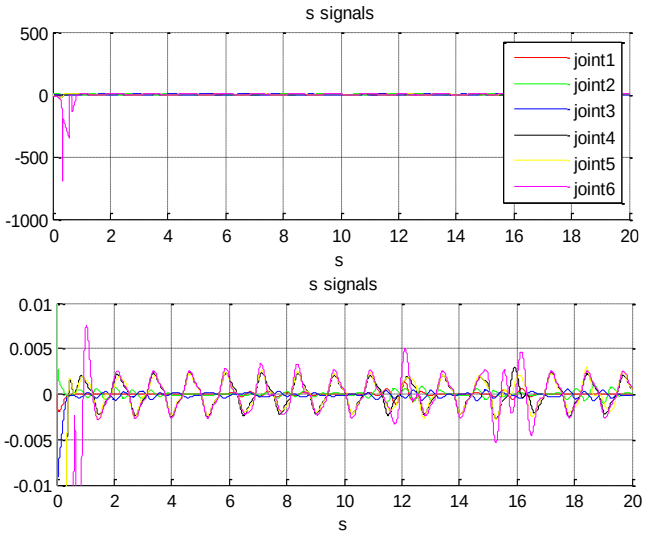


Figure 10. *s* signals.

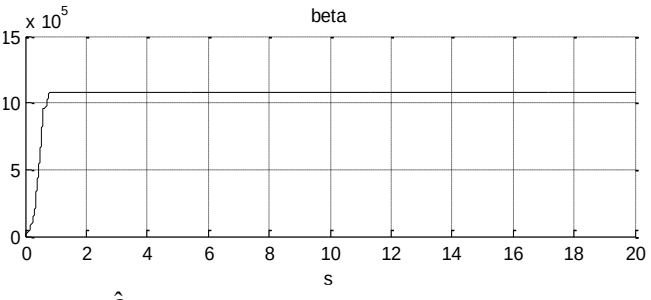


Figure 11. $\hat{\beta}$ signals.

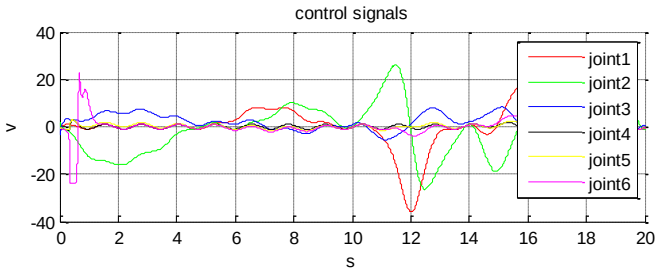


Figure 12. Control signals.

Copyright © 2016 International Journal of Automation and Smart Technology

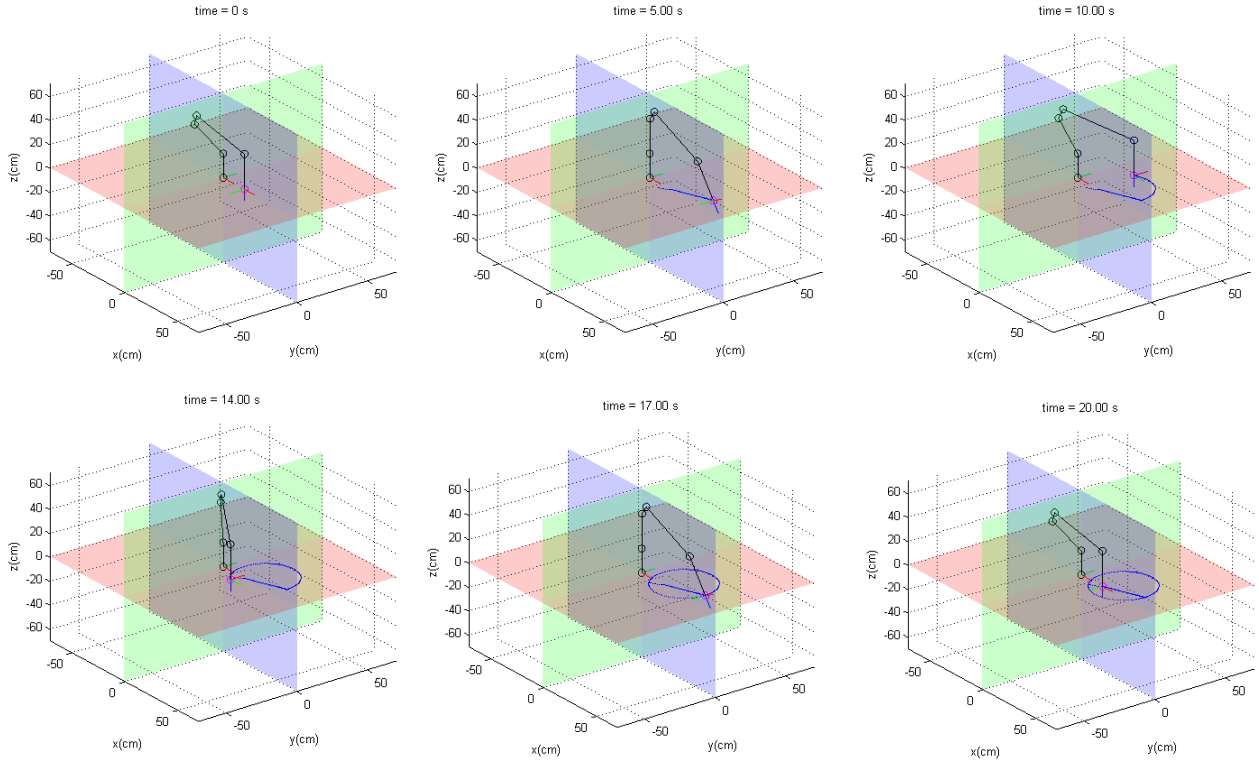


Figure 13. Motion of the RLED robot manipulator at t=0, t=5, t=10, t=14, t=17, and t=20.

References

[1] K.-Y. Kim, H.-S. Song, J.-W. Suh, and J.-J. Lee, "A Novel Surgical Manipulator with Workspace-Conversion Ability for Telesurgery," *IEEE/ASME Transactions on Mechatronics*, vol. 18, no. 1, pp. 200–211, 2013. doi: 10.1109/TMECH.2011.2164929

[2] W. Xu, B. Liang, D. Gao, and Y. Xu, "A Space Robotic System Used for On-Orbit Servicing in The Geostationary Orbit," in proceeding of *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipei, Taiwan, Oct. 18-22, 2010, pp. 4089–4094. doi: 10.1109/IROS.2010.5648894

[3] J. Harder, M. Wilde, and A. Fleischner, "Technology Development for Real-Time Teleoperated Spacecraft Mission Operations," in proceeding of *IEEE Aerospace Conference*, Big Sky, MT, 2013, pp. 1–10. doi: 10.1109/AERO.2013.6497343

[4] M. W. Carey, E. M. Kurz, J. D. Matte, T. D. Perrault, and T. Padir, "Novel EOD Robot Design with Dexterous Gripper and Intuitive Teleoperation," in proceeding of *World Automation Congress*, Puerto Vallarta, Mexico, June 24-28, 2012, pp. 1–6.

[5] S. Hagspiel, "Modern Expert Arc and Spot Welding: World's First Arc and Spot Welding Robots with 7 Controlled Axes," in proceeding of *41st International Symposium on Robotics and 6th German Conference on Robotics*, Munich, Germany, June 7-9, 2010, pp. 1 – 6.

[6] P. J. From, J. Gunnar, and J. T. Gravdahl, "Optimal Paint Gun Orientation in Spray Paint Applications-Experimental Results," *IEEE Transactions on Automation Science and Engineering*, vol. 8, pp. 438–442, 2011. doi: 10.1109/TASE.2010.2089450

[7] C. Park, K. Park, and D. Kim, "Design of Dual Arm Robot Manipulator for Precision Assembly of Mechanical Parts," in proceeding of *International Conference on Smart Manufacturing Application*, Goyang-si, Korea, April 9-11, 2008, pp. 424–427. doi: 10.1109/ICSMA.2008.4505576

[8] T. I. J. Tsay, Y.-F. Lai, and Y.-L. Hsiao, "Material Handling of A Mobile Manipulator using An Eye-in-Hand Vision System," in proceeding of *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipei, Taiwan, Oct. 18-22, 2010, pp. 4743–4748. doi: 10.1109/IROS.2010.5649578

[9] J.-K. Lee, K. Kim, B.-S. Park, and J.-S. Yoon, "Force-Reflecting Servo-Manipulators for Remote Handling Task in A Radioactive Environment," in proceeding of *International Conference on Control, Automation, and Systems*, Seoul, Korea, Oct. 17-20, 2007, pp. 1025–1028. doi: 10.1109/ICCAS.2007.4407048

[10] X.-J. Liu, J. Wang, F. Gao, and L.-P. Wang, "On the Analysis of A New Spatial Three-Degrees-of-Freedom

Copyright © 2016 International Journal of Automation and Smart Technology

Parallel Manipulator," *IEEE Transactions on Robotics and Automation*, vol. 17, pp. 959–968, 2001. doi: 10.1109/70.976031

[11] T. J. Tarn, A. K. Bejczy, X. Yun, and Z. Li, "Effect of Motor Dynamics on Nonlinear Feedback Robot Arm Control," *IEEE Transactions on Robotics and Automation*, vol. 7, pp. 114–122, 1991. doi: 10.1109/70.68075

[12] M. Ahmadipour, A. Khayatian, and M. Dehghani, "Adaptive Backstepping Control of Rigid-Link Electrically Driven Robots with Uncertain Kinematics and Dynamics," in proceeding of *The 2nd International Conference on Control, Instrumentation, and Automation*, Shiraz, Iran, Dec. 27-29, 2011, pp. 911–916. doi: 10.1109/ICCIAutom.2011.6356783

[13] M. M. Bridges, D. M. Dawson, and X. Gao, "Adaptive Control of Rigid-Link Electrically-Driven Robots," in proceeding of *32nd IEEE Conference on Decision and Control*, San Antonio, TX, Dec. 15-17, 1993, pp. 159–165. doi: 10.1109/CDC.1993.325171

[14] Y. Stepanenko, "Adaptive Motion Control of Rigid-Link Electrically-Driven Robot Manipulators," in proceeding of *IEEE International Conference on Robotics and Automation*, San Diego, CA, May 8-13, 1994, pp. 630–635. doi: 10.1109/ROBOT.1994.351415

[15] T. Burg, D. Dawson, J. Hu, and M. De Queiroz, "An Adaptive Partial State-Feedback Controller for RLED Robot Manipulators," *IEEE Transactions on Automatic Control*, vol. 41, pp. 1024–1030, 1996. doi: 10.1109/9.508910

[16] M. Trusca, G. Lazea, and E. Lupu, "Adaptive Control Approach in Case of A Rigid Link Robot System," in proceeding of *IEEE International Conference on Automation, Quality and Testing, Robotics*, Cluj-Napoca, Romania, May 22-25, 2008, pp. 211–214. doi: 10.1109/AQTR.2008.4588823

[17] A.-C. Huang and M.-C. Chien, *Adaptive Control of Robot Manipulators*. World Scientific Publishing Co. Pte. Ltd., 2010, pp. 105–127.

[18] G.-J. Liu and A. A. Goldenberg, "Robust Control of Robot Manipulators Incorporating Motor Dynamics," in proceeding of *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Yokohama, Japan, July 26-30, 1993, pp. 68–75. doi: 10.1109/IROS.1993.583081

Copyright © 2016 International Journal of Automation and Smart Technology