JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 29, No. 1, January–February 2006

Engineering Notes

ENGINEERING NOTES are short manuscripts describing new developments or important results of a preliminary nature. These Notes should not exceed 2500 words (where a figure or table counts as 200 words). Following informal review by the Editors, they may be published within a few months of the date of receipt. Style requirements are the same as for regular contributions (see inside back cover).

Dynamic Modeling and Neural-Network Adaptive Control of a Deployable Manipulator System

Y. Cao* and C. W. de Silva†
University of British Columbia, Vancouver, British Columbia V6T 1Z4, Canada

I. Introduction

In this Note, the dynamic formulation is presented for a novel deployable manipulator system. Each module of this multimodule manipulator system consists of a deployable (prismatic) link and a slewing (revolute) joint. Previous research[1] on the dynamic modeling and adaptive control of manipulator system involves a revolute joint only. Here, a neural-network-based adaptive controller is developed for this deployable manipulator system. The model includes both joint flexibilities and link flexibilities. Neural networks are used to approximate the unknown terms in the dynamic equations of the manipulator. The controller is adapted on that basis, with the objective of reducing the tracking error of the robot. The applicability and effectiveness of the neural-network control scheme for this manipulator system are tested through computer simulations.

II. Dynamic Modeling

Consider the n-link deployable manipulator model shown in Fig. 1. Each manipulator module comprises two joints (degrees of freedom): one free to slew, through a revolute joint, and the other permitted to deploy (and retrieve), through a prismatic joint. The links and the joints are considered flexible. Detailed derivation of the dynamic model can be found in Ref. 2. For brevity, only the final form of equations of motion is shown in the following

$$\boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{V}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q}) = \boldsymbol{\tau} \tag{1}$$

where $\boldsymbol{M}(\boldsymbol{q})$, $\boldsymbol{V}(\dot{\boldsymbol{q}}, \boldsymbol{q})$, and $\boldsymbol{G}(\boldsymbol{q})$ are system matrices. Dynamic study of this class of manipulator system has been conducted by Modi et al.[3]

III. Neural-Network Adaptive Control

The neural networks are used to approximate the elements of system matrices $\boldsymbol{M}(\boldsymbol{q})$, $\boldsymbol{V}(\dot{\boldsymbol{q}}, \boldsymbol{q})$, and $\boldsymbol{G}(\boldsymbol{q})$, that is, $M_{kj}(\boldsymbol{q})$, $V_{kj}(\dot{\boldsymbol{q}}, \boldsymbol{q})$, and $G_k(\boldsymbol{q})$, respectively. Specifically,

$$M_{kj} = \boldsymbol{W}_{M_{kj}}^T \phi_{M_{kj}}(\boldsymbol{q}) + \varepsilon_{M_{kj}}(\boldsymbol{q}) \tag{2}$$

$$G_k = \boldsymbol{W}_{G_k}^T \phi_{G_k}(\boldsymbol{q}) + \varepsilon_{G_k}(\boldsymbol{q}) \tag{3}$$

$$V_{kj} = \boldsymbol{W}_{V_{kj}}^T \phi_{V_{kj}}(\dot{\boldsymbol{q}}, \boldsymbol{q}) + \varepsilon_{V_{kj}}(\dot{\boldsymbol{q}}, \boldsymbol{q}) \tag{4}$$

where $\boldsymbol{W}_{M_{kj}}$, $\boldsymbol{W}_{V_{kj}}$, and $\boldsymbol{W}_{G_k}$ are the vectors of neural-network weights; $\phi_{M_{kj}}(\boldsymbol{q})$, $\phi_{V_{kj}}(\dot{\boldsymbol{q}}, \boldsymbol{q})$, and $\phi_{G_k}(\boldsymbol{q})$ are the vectors of activation functions; and $\varepsilon_{M_{kj}}(\boldsymbol{q})$, $\varepsilon_{V_{kj}}(\dot{\boldsymbol{q}}, \boldsymbol{q})$, and $\varepsilon_{G_k}(\boldsymbol{q})$ are the approximation errors. Here, $\phi_{M_{kj}}(\boldsymbol{q})$, $\phi_{V_{kj}}(\dot{\boldsymbol{q}}, \boldsymbol{q})$, and $\phi_{G_k}(\boldsymbol{q})$ are the Gaussian radial basis functions.

In the adaptive neural-network control method the weights of the neural networks are updated online by a parameter adaptation algorithm requiring no matrix inversion. The estimates of $\boldsymbol{M}(\boldsymbol{q})$, $\boldsymbol{V}(\boldsymbol{q}, \dot{\boldsymbol{q}})$, and $\boldsymbol{G}(\boldsymbol{q})$ are obtained by replacing the true weighting matrices $\boldsymbol{W}_M$, $\boldsymbol{W}_V$, and $\boldsymbol{W}_G$ by their estimates $\hat{\boldsymbol{W}}_M$, $\hat{\boldsymbol{W}}_V$, and $\hat{\boldsymbol{W}}_G$, respectively, that is,

$$\hat{\boldsymbol{M}}(\boldsymbol{q}) = \hat{\boldsymbol{W}}_M^T \phi_M(\boldsymbol{q}), \qquad \hat{\boldsymbol{V}}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \hat{\boldsymbol{W}}_V^T \phi_M(z)$$

$$\hat{\boldsymbol{G}}(\boldsymbol{q}) = \hat{\boldsymbol{W}}_G^T \phi_M(\boldsymbol{q}) \tag{5}$$

We use the control law

$$\boldsymbol{\tau} = \hat{\boldsymbol{M}}(\boldsymbol{q})(\ddot{\boldsymbol{q}}_d + \boldsymbol{\Lambda}\dot{\boldsymbol{e}}) + \hat{\boldsymbol{V}}(\dot{\boldsymbol{q}}, \boldsymbol{q})(\dot{\boldsymbol{q}}_d + \boldsymbol{\Lambda}\boldsymbol{e}) + \hat{\boldsymbol{G}}(\boldsymbol{q}) + \boldsymbol{K}_v\dot{\boldsymbol{e}} + \boldsymbol{K}_v\boldsymbol{\Lambda}\boldsymbol{e} \tag{6}$$

where the tracking error $\boldsymbol{e}$ is defined as $\boldsymbol{e} = \boldsymbol{q}_d - \boldsymbol{q}$, with $\boldsymbol{q}_d$ denoting the desired trajectory and where $\boldsymbol{\Lambda}$ is a positive definite and diagonal matrix. With this control law, it is known[4] that if $K_v > 0$ then the closed-loop system is asymptotically stable, that is, $\boldsymbol{s} = \boldsymbol{\Lambda}\boldsymbol{e} + \dot{\boldsymbol{e}} \to 0$ as $t \to \infty$ under the following parameter adaptation laws:

$$\dot{\hat{\boldsymbol{W}}}_{M_{kj}} = \boldsymbol{\Gamma}_{M_{ij}} \phi_{M_{kj}} \ddot{\boldsymbol{q}}_{sj} \boldsymbol{s}_k, \qquad \dot{\hat{\boldsymbol{W}}}_{V_{kj}} = \boldsymbol{\Gamma}_{V_{kj}} \phi_{V_{kj}} \dot{\boldsymbol{q}}_{sj} \boldsymbol{s}_k$$

$$\dot{\hat{\boldsymbol{W}}}_{G_k} = \boldsymbol{\Gamma}_{G_k} \phi_{G_k} \boldsymbol{s}_k \tag{7}$$

where $\boldsymbol{\Gamma}_{M_{kj}}$, $\boldsymbol{\Gamma}_{V_{kj}}$, and $\boldsymbol{\Gamma}_{G_k}$ are symmetric positive-definite constant matrices; $\hat{\boldsymbol{W}}_{M_{kj}}$, $\hat{\boldsymbol{W}}_{V_{kj}}$, and $\hat{\boldsymbol{W}}_{G_k}$ are elements of $\hat{\boldsymbol{W}}_M$, $\hat{\boldsymbol{W}}_V$, and $\hat{\boldsymbol{W}}_G$, respectively; $\ddot{\boldsymbol{q}}_s = \ddot{\boldsymbol{q}}_d + \boldsymbol{\Lambda}\dot{\boldsymbol{e}}$; and $\dot{\boldsymbol{q}}_s = \dot{\boldsymbol{q}}_d + \boldsymbol{\Lambda}\boldsymbol{e}$. Figure 2 shows the neural-network adaptive control scheme.

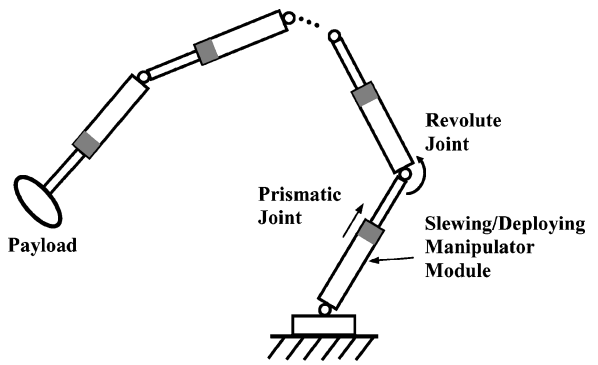

Fig. 1 Schematic diagram of multimodule deployable manipulator system.

Received 21 May 2004; revision received 17 November 2004; accepted for publication 22 November 2004.

*NSERC Postdoctoral Fellow, Department of Mechanical Engineering.
†Professor and NSERC Chair, Department of Mechanical Engineering.

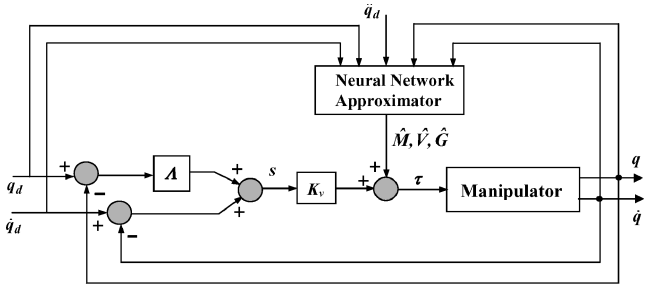

Fig. 2 Neural-network adaptive control of manipulator system.

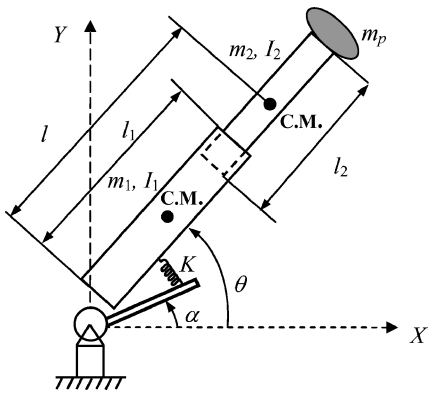

Fig. 3 Single-module deployable manipulator with joint flexibility.

IV. Numerical Simulations

First consider the manipulator system shown in Fig. 3. It consists of two joints: one free to slew and the other permitted to deploy. The vector of generalized coordinates, which describes the complete motion of the manipulator model, is given by $\boldsymbol{q} = [l \;\; \theta \;\; \alpha]^T$, where l represents the distance between the origin of the reference frame located at the revolute joint and the center of mass of the deployable link, θ denotes the slewing angle associated with the robot arm, and α denotes the motor rotor angular position. The equations of motion can be expressed in the vector matrix form as

$$\boldsymbol{M}(\boldsymbol{q}_r)\ddot{\boldsymbol{q}}_r + \boldsymbol{V}(\dot{\boldsymbol{q}}_r, \boldsymbol{q}_r)\dot{\boldsymbol{q}}_r + \boldsymbol{G}(\boldsymbol{q}_r) = \begin{bmatrix} \tau_l \\ K(\alpha - \theta) \end{bmatrix} \tag{8}$$

$$J\ddot{\alpha} + K(\alpha - \theta) = \tau_\alpha \tag{9}$$

where τ_l and τ_α are the force and torque inputs for the deployable link and the revolute joint, respectively; $\boldsymbol{q}_r = [l \;\; \theta]^T$; and $\boldsymbol{M}(\boldsymbol{q}_r)$, $\boldsymbol{V}(\dot{\boldsymbol{q}}_r, \boldsymbol{q}_r)$, and $\boldsymbol{G}(\boldsymbol{q}_r)$ are system matrices corresponding to the rigid model.

With the derived equations of motions, a singular perturbation method is used to transform the original flexible joint robot model into a two-timescale model, specifically, the well-known steady-state model and the boundary-layer model. The singular perturbation theory and its application to the dynamic model are considered now. Here, for brevity, only the final results of the approach are presented. A more comprehensive treatment of this theory is given by Khalil.[5] The steady-state model is

$$[\boldsymbol{M}(\boldsymbol{q}_r) + J]\ddot{\boldsymbol{q}}_r + \boldsymbol{V}(\dot{\boldsymbol{q}}_r, \boldsymbol{q}_r) + \boldsymbol{G}(\boldsymbol{q}_r) = \boldsymbol{\tau}_{\text{slow}} \tag{10}$$

where $\tau_{\text{slow}} = [\tau_l \;\; \tau_{\alpha,\text{slow}}]^T$. Note that the steady-state model obtained here has the form of equivalent rigid-body dynamics. The motor torque input $\tau_\alpha = \tau_{\alpha,\text{slow}} + \tau_{\alpha,\text{fast}}$. The slow control input $\tau_{\alpha,\text{slow}}$ is applied to the resulting steady-state model, and the fast control input $\tau_{\alpha,\text{fast}}$ is for the boundary-layer model, which provides damping and additional stiffness. We choose $\tau_{\alpha,\text{fast}} = K_d(\dot{\theta} - \dot{\alpha}) + K_p(\theta - \alpha)$, where K_d and K_p are control gains.

The steady-state model [Eq. (10)] has the form of an equivalent rigid-body model. Therefore, the neural-network-based adaptive controller is applied to control the steady-state model.

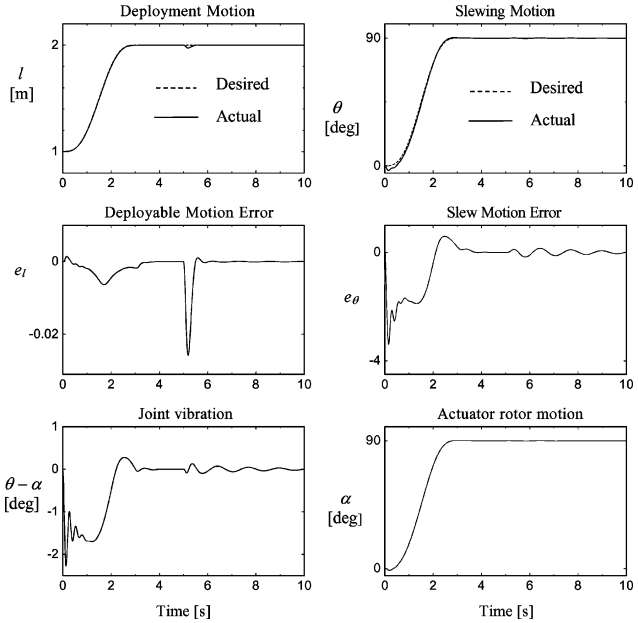

Fig. 4 Neural-network adaptive control of deployable manipulator.

Hence,

$$\boldsymbol{\tau}_{\alpha,\text{slow}} = \hat{\boldsymbol{M}}(\boldsymbol{q}_r)(\ddot{\boldsymbol{q}}_{rd} + \boldsymbol{\Lambda}\dot{\boldsymbol{e}}) + \hat{\boldsymbol{V}}(\dot{\boldsymbol{q}}_r, \boldsymbol{q}_r)(\dot{\boldsymbol{q}}_{rd} + \boldsymbol{\Lambda}\boldsymbol{e}) + \hat{\boldsymbol{G}}(\boldsymbol{q}_r) + \boldsymbol{K}_v\dot{\boldsymbol{e}} + \boldsymbol{K}_v\boldsymbol{\Lambda}\boldsymbol{e} \tag{11}$$

where $\boldsymbol{q}_{rd}$ represents the desired values for coordinates $\boldsymbol{q}_r = [l \;\; \theta]^T$.

In the present study, a sine-on-ramp acceleration profile is adopted for prescribed maneuvers. It assures zero velocity and zero acceleration both in the beginning and at the end of a maneuver, thereby smoothing the structural response of the system. The trajectory of a specified maneuver is a sine-on-ramp profile given by

$$\boldsymbol{q}_d(t) = \{[\boldsymbol{q}_d(t_d) - \boldsymbol{q}(0)]/t_d\}\{t - (t_d/2\pi)\sin[(2\pi/t_d)t]\} + \boldsymbol{q}(0) \tag{12}$$

where $\boldsymbol{q}_d$ is the desired trajectory, t is time, and t_d is the time required for the maneuver. The system generalized coordinates are initialized as $\boldsymbol{q}(0) = [r(0) \;\; \theta(0) \;\; \alpha(0)]^T = [1\,\text{m} \;\; 0\,\text{deg} \;\; 0\,\text{deg}]^T$, and the desired end values are $\boldsymbol{q}_d(t_d) = [2\,\text{m} \;\; 90\,\text{deg} \;\; 90\,\text{deg}]^T$, where $t_d = 3$ s. The adaptation algorithm is activated with $\Gamma_M = \text{diag}[2.0]$, $\Gamma_V = \text{diag}[2.0]$, and $\Gamma_G = \text{diag}[5.0]$. The controller gains are chosen as $\boldsymbol{K}_v = \text{diag}[10, 50]$, $K_d = 0.1$, and $K_p = 0.5$. Manipulator system parameters are $m_1 = m_2 = 2$ kg, $l_1 = l_2 = 1$ m, $J = 1\ \text{kg}\cdot\text{m}^2$, and $K = 500\ \text{N}\cdot\text{m/rad}$.

To test the capability of load disturbance rejection of the controller, a payload of mass $m_p = 0.8$ kg is picked up by the robot at time $t = 5.0$ s.

Figure 4 shows the simulation results, where the solid lines are used to represent the actual values and the dashed lines for the desired values. Note that e_l and e_θ denote the tracking errors of the prismatic and revolute joints, respectively, and $\theta - \alpha$ represents the joint elastic motion (vibration). The simulation results demonstrate that the adaptive neural-network controller can effectively control an unknown nonlinear robotic system, even in the presence of sudden load disturbances.

Now consider the manipulator system shown in Fig. 5. Both the robot arm and the revolute joint are considered flexible here. The vector of generalized coordinates, which describes the complete motion of the manipulator model, is given by $\boldsymbol{q} = [\alpha \;\; \theta \;\; \delta \;\; l]^T$, where α denotes the motor rotor position, θ denotes the slewing angle associated with the robot arm position, δ is the flexible generalized coordinate, and l represents the length of the manipulator.

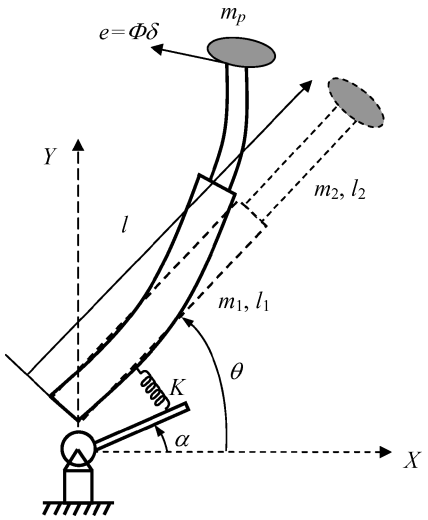

Fig. 5 Structure of deployable manipulator with flexible revolute joint and flexible arm.

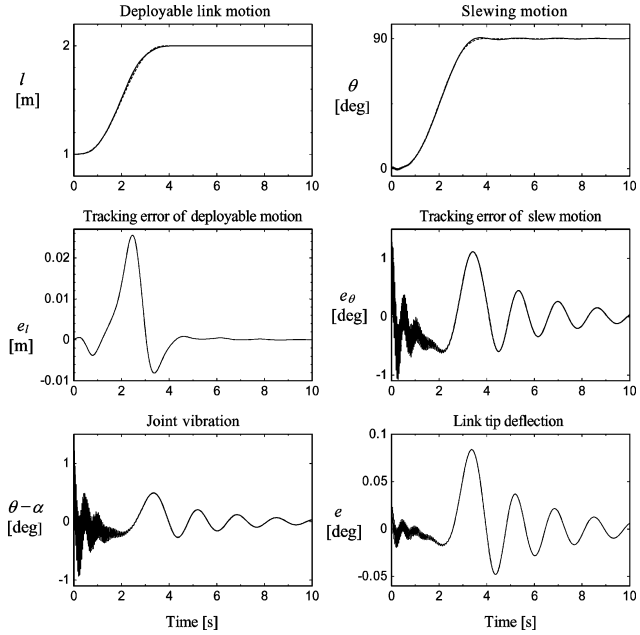

Fig. 6 System response and control input with $K_v = \text{diag}[10, 50]$, $K_d = 2$, and $K_p = 5$.

By the use of the Lagrangian procedure, the equations of motion of the system are described as

$$\begin{bmatrix} \boldsymbol{M}_{rr} & \boldsymbol{M}_{rf} \\ \boldsymbol{M}_{fr} & \boldsymbol{M}_{ff} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{q}}_r \\ \ddot{\boldsymbol{q}}_f \end{bmatrix} + \begin{bmatrix} \boldsymbol{V}_{rr} & \boldsymbol{V}_{rf} \\ \boldsymbol{V}_{fr} & \boldsymbol{V}_{ff} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{q}}_r \\ \dot{\boldsymbol{q}}_f \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & K_{ff} \end{bmatrix} \begin{bmatrix} \boldsymbol{q}_r \\ \boldsymbol{q}_f \end{bmatrix} = \begin{bmatrix} \tau_l \\ K(\alpha - \theta) \\ 0 \end{bmatrix} \tag{13}$$

$$j\ddot{\alpha} + K(\alpha - \theta) = \tau_\alpha \tag{14}$$

where τ_l and τ_α are the force and the torque inputs for the deployable link and the revolute joint, respectively; $\boldsymbol{q}_r = [l \;\; \theta]^T$; $\boldsymbol{q}_f = \delta$; and the system matrices $(\boldsymbol{M}, \boldsymbol{V})$ are partitioned according to the rigid and flexible generalized coordinates. Also, K_{ff} is the stiffness matrix. Because we only use one flexible mode, K_{ff} is a constant in the present case. Note that it is assumed that both deploying and slewing links are uniform and have the same flexural rigidity EI. Also, only one flexible mode shape is used in modeling.

The system generalized coordinates are initialized as $\boldsymbol{q}(0) = [\alpha(0) \;\; \theta(0) \;\; \delta(0) \;\; r(0)]^T = [0\,\text{deg} \;\; 0\,\text{deg} \;\; 0.01 \;\; 1\,\text{m}]^T$, and the desired values are $\boldsymbol{q}_d(t_d) = [90\,\text{deg} \;\; 90\,\text{deg} \;\; 0 \;\; 2\,\text{m}]^T$, where $t_d = 4$ s.

Again, a sine-on-ramp profile is used for the maneuvers. The adaptation algorithm is activated with $\Gamma_M = \text{diag}[2.0]$, $\Gamma_V = \text{diag}[2.0]$, and $\Gamma_G = \text{diag}[5.0]$. The gains for the slow controller are chosen as $\boldsymbol{K}_v = \text{diag}[10, 50]$. For the fast controller $K_d = 2$ and $K_p = 5$.

Figure 6 presents the response of the system variables. The tracking errors of the revolute joint e_θ, the deploying link e_l, and the joint flexibility angle $(\theta - \alpha)$ are also shown in Fig. 6. From the result, it is seen that the system is able to follow the trajectories accurately. With this control structure, the maximum tracking errors for the revolute joint and the deployable link are around 1 deg and 0.03 m, respectively. The level of control input is permissible. Furthermore, during the maneuver, the manipulator generates a maximum tip deflection of 0.06 m and a joint vibration $(\theta - \alpha)$ of amplitude less than 1 deg. Notice that the oscillations of the joint and the link converge to zero. Furthermore, the vibrations of the joint and the link are coupled, as expected. Initial vibrations are quickly damped out by the controller.

V. Conclusions

In this Note, a neural-network-based adaptive control was applied to a manipulator, including both joint flexibility and link flexibility. The singular perturbation approach was used for control in the presence of flexibilities. Simulation results showed the effectiveness of the approach. For future work, the neural-network adaptive control scheme is expected to be validated on a prototype deployable manipulator system in the Industrial Robotic Laboratory at the University of British Columbia.

References

[1]Sasiadek, J. Z., and Srinivasan, R., "Dynamic Modeling and Adaptive Control of a Single-Link Flexible Manipulator," *Journal of Guidance, Dynamics, and Control*, Vol. 12, No. 6, 1989, pp. 838–844.

[2]Cao, Y., "Dynamic Modeling and Advanced Control of Ground-Based and Space-Based Deployable Manipulator Systems," Ph.D. Dissertation, Dept. of Mechanical Engineering, Univ. of British Columbia, Vancouver, May 2004.

[3]Modi, V. J., Cao, Y., and de Silva, C. W., "A Class of Novel Space Platform-Based Manipulators With Slewing and Deployable Links: Analyses and Experiments," *Journal of Vibration and Control*, Vol. 7, No. 8, 2001, pp. 1111–1161.

[4]Lewis, F. L., and Jagannathan, S., *Neural Network Control of Robot Manipulators and Nonlinear Systems*, Taylor and Francis, Philadelphia, 1999, pp. 173–206.

[5]Khalil, H. K., *Nonlinear Systems*, 2nd ed., Prentice–Hall, Upper Saddle River, NJ, 1996, pp. 423–459.