designed controller was implemented using a real-time control environment and experimental results were presented. The robustness of the control system was validated for fixed as well as variable joint angles during manipulator motion.

References

[1] J. Angeles, A. Morozov, and O. Navarro, "A novel manipulator architecture for the production of SCARA motions," in *Proc. IEEE Int. Conf. Robot. Autom.*, San Francisco, CA, 2000, pp. 2370–2375.

[2] M. Benosman, F. Boyer, G. L. Vey, and D. Primault, "Flexible-link manipulators: From modelling to control," *J. Intell. Robot. Syst.*, vol. 34, no. 4, pp. 381–414, 2002.

[3] G. Bilodeau and S. Aziz, "International space station assembly flight: Simulation of SSRMS dynamics," in *Proc. Int. Symp. Robot.*, Montréal, Canada, 2000, pp. 447–452.

[4] W. J. Book, "Controlled motion in an elastic world," *ASME J. Dyn. Syst. Meas. Control*, vol. 115, pp. 252–260, 1993.

[5] W. J. Book and S. H. Lee, "Vibration control of a large flexible manipulator by a small robotic arm," in *Proc. Am. Control Conf.*, Pittsburgh, PA, 1989, vol. 2, pp. 1377–1380.

[6] S. B. Choi, Y. K. Park, and T. Fukuda, "A proof-of-concept investigation on active vibration control of hybrid smart structures," *Mechatronics*, vol. 8, no. 6, pp. 673–689, Aug. 1998.

[7] A. De Luca, S. Iannitti, R. Mattone, and G. Oriolo, "Control problems in underactuated manipulators," in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatronics*, Como, Italy, Jul. 2001, pp. 855–861.

[8] A. De Luca and G. Di Giovanni, "Rest to rest motion of a two-link flexible robot with a flexible forearm," in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatronics*, Como, Italy, Jul. 2001, pp. 929–935.

[9] J. J. Dosch, D. J. Inman, and E. Garcia, "A self-sensing piezoelectric actuator for collocated control," *J. Intell. Mater. Syst. Struct.*, vol. 3, pp. 166–185, 1992.

[10] J. Doyle, B. Francis, and A. Tannenbaum, *Feedback Control Theory*. Toronto, Canada: Macmillan, 1990.

[11] S. Hanagud, M. B. de Noyer, H. Luo, D. Henderson, and K. S. Nagaraja, "Tail buffet alleviation of high performance twin tail aircraft using piezostack actuators," *Am. Inst. Aeronaut. Astronaut.*, vol. 40, no. 4, pp. 619–627, 2002.

[12] J. M. Hervé and F. Sparacino, "STAR, a new concept in robotics," presented at the Int. Conf. 3K-ARK. Ferrara, Italy, Sep. 1992.

[13] M. R. Kermani, "Vibration suppression in large-scale robotic manipulators using piezoelectric materials," Ph.D. dissertation, The Univ. Western Ontario, Aug. 2005.

[14] M. R. Kermani, R. V. Patel, and M. Moallem, "Flexture control using piezostack actuators: Design and implementation," *IEEE/ASME Trans. Mechatronics*, vol. 10, no. 2, pp. 181–188, Apr. 2005.

[15] B. A. L. de la Barra, "On undershoot in SISO system," *IEEE Trans. Autom. Control*, vol. 39, no. 3, pp. 578–581, Mar. 1994.

[16] D. Niederberger, S. Behrens, A. J. Fleming, S. O. R. Moheimani, and M. Morari, "Adaptive electromagnetic shunt damping," *IEEE/ASME Trans. Mechatronics*, vol. 11, no. 1, pp. 103–108, Feb. 2006.

[17] A. Preumount, *Vibration Control of Active Structures, An Introduction*. Boston, MA: Kluwer Academic, 1997.

[18] S. Skogestad and I. PostlethWaite, *Multivariable Feedback Control, Analysis, and Design*. Chichester, U.K.: Wiley, 1996.

[19] W. Sunada and S. Dubowsky, "On the dynamic analysis and behavior of industrial robotic manipulators with elastic members," *ASME J. Mech. Des.*, vol. 105, no. 1, pp. 42–51, 1983.

[20] A. J. Young and C. H. Hansen, "Control of flexural vibration in stiffened structures using multiple piezoceramic actuators," *Appl. Acoust.*, vol. 49, no. 1, pp. 17–48, 1996.

A Neural Network Controller for Continuum Robots

David Braganza, Darren M. Dawson, Ian D. Walker,
and Nitendra Nath

***Abstract*—Continuum or hyper-redundant robot manipulators can exhibit behavior similar to biological trunks, tentacles, or snakes. Unlike traditional rigid-link robot manipulators, continuum robot manipulators do not have rigid joints, hence these manipulators are extremely dexterous, compliant, and are capable of dynamic adaptive manipulation in unstructured environments. However, the development of high-performance control algorithms for these manipulators is quite a challenge, due to their unique design and the high degree of uncertainty in their dynamic models. In this paper, a controller for continuum robots, which utilizes a neural network feedforward component to compensate for dynamic uncertainties is presented. Experimental results using the OCTARM, which is a soft extensible continuum manipulator, are provided to illustrate that the addition of the neural network feedforward component to the controller provides improved performance.**

***Index Terms*—Continuum robot, feedforward control, neural network, robot control.**

I. Introduction

Continuum or hyper-redundant manipulators [1], [2] belong to a special class of robotic manipulators, which are designed to exhibit behavior similar to biological trunks [3]–[6], tentacles [7], or snakes [8]. Unlike traditional rigid-link robot manipulators, continuum robot manipulators do not have rigid joints and they have a large number of degrees of freedom, this enables continuum manipulators to have some very useful properties. The continuum manipulators can be compliant, extremely dexterous, flexible, and capable of dynamic adaptive manipulation in highly unstructured environments. These properties of soft continuum robot manipulators make them uniquely suited for a large number of applications, including search and rescue, underwater, and space exploration.

The development of high-performance model-based control algorithms for continuum manipulators is a challenging problem for several reasons; since the manipulators must be modeled as continuous curves, the kinematic and dynamic models are difficult to derive, also the manipulators body is soft and flexible that makes accurate control difficult to achieve. There have been several different approaches, which researchers have studied for the control of continuum robot manipulators. For example, Matsuno *et al.* [9] proposed kinematic con-

Manuscript received January 8, 2007; revised May 24, 2007. This paper was recommended for publication by Associate Editor G. Antonelli and Editor K. Lynch upon evaluation of the reviewers' comments. This work was supported in part by the Department of Commerce (DOC), in part by the ARO Automotive Center, in part by the Department of Energy (DOE), in part by the Honda Corporation, and in part by the Defence Advanced Research Projects Energy (DARPA).

D. Braganza is with OFS Fitel, Sturbridge, MA 01566 USA (e-mail: dbraganza@ofsoptics.com).

D. Dawson, I. Walker and N. Nath are with the Department of Electrical and Computer Engineering, Clemson University, Clemson, SC 29634-0915 USA (e-mail: ddawson@ces.clemson.edu; ianw@ces.clemson.edu; nnath@ces.clemson.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TRO.2007.906248

trol techniques for continuum manipulators and [10]–[13] described set-point controllers for continuum manipulators. In [10], a fuzzy controller was presented and [11] presented an artificial potential function method for obstacle avoidance for a variable length continuum manipulator. In [12], an exponentially stable controller for inextensible continuum manipulators was presented and [13] described sliding mode and impedance control techniques for hyper-flexible manipulators. There are very few techniques, which target the more general tracking control problem for continuum manipulators with one of the exceptions being shape tracking control [14], where the manipulator follows a desired shape prescribed by a time-varying spatial curve. A trajectory tracking shape controller for a wheeled snake robot, based on the dynamic model and the nonholonomic constraint conditions of the robot was presented in [15]. All of the aforementioned control techniques, for the tracking control of continuum manipulators, require the exact dynamic model to be known.

The concept of using a neural network-based control strategy for joint tracking control of a conventional robot manipulator is quite well understood. In [16], [17], and the references therein, neural network controllers were developed for a large number of robot manipulator models, including rigid-link manipulators and flexible-joint manipulators. These two results survey in great detail the research, which has been conducted on closed-loop neural network control of robotic manipulators. A selection of some recent results on neural network control of robotic manipulators include [18]–[21]. Kim *et al.* [18] developed an output feedback controller for robot manipulators with online weight adaptation, which provided uniformly ultimately bounded (UUB) tracking. Sun *et al.* [19] presented a discrete neural network controller for robots with uncertain dynamics, which did not require offline training, however this controller also achieves only UUB tracking. Patino *et al.* [20] developed a neural network-based robust adaptive tracking controller based on static weights, which must be trained offline. The controller provided global asymptotic stability of the tracking error by including a signum function in the controller. However, the inclusion of the signum function in the controller leads to high-frequency chattering in the control signals, as this is undesirable the authors propose substituting the signum function with a saturation function, which leads to a degradation of the tracking performance from a global asymptotic stability tracking result to a UUB result. In [21], a parametric adaptive controller that adapts for robot dynamic parameters was coupled with a neural network to compensate for the unmodeled friction effects. The controller provided asymptotic stability, but required the structure of the robot's dynamic model to be known *a priori*.

From this brief survey of results and in general, from control theory, we note that the more information that a controller has about the plant, the better the tracking result. As has been mentioned previously, the dynamic modeling of extensible continuum manipulators is difficult and remains an active research topic. All of the previously mentioned controllers for continuum robot manipulators are either set-point controllers or tracking controllers, which require an accurate dynamic model of the manipulator. Hence, these controllers are not suitable for tracking control of continuum manipulators as they will exhibit diminished performance. This reduced performance can be a drawback as it does not allow all the capabilities of the continuum manipulator to be utilized. The problem of continuum robot control, thus represents a significant barrier to progress in this emerging field.

As a complete accurate dynamic model of the continuum manipulator does not exist, the main focus of the current work is to develop an efficient tracking controller for extensible continuum manipulators, which can deal with a high level of uncertainty in the structure of the manipulators dynamic model. The proposed controller, which is based on our preliminary work [22] consists of a neural network feedforward component along with a nonlinear feedback component. Specifically, the design of the neural network component is based on the augmented back propagation algorithm [16], and it is used to compensate for the nonlinear uncertain dynamics of the continuum robot manipulator by leveraging the universal approximation properties of the neural network as a feedforward compensator. The feedback component utilized is a continuous nonlinear controller [23], which does not require any model information. The advantages of the proposed control scheme compared to the previously mentioned works is that the controller is continuous and asymptotic tracking can be proved without any prior knowledge of the robot dynamic model. Furthermore, the back propagation technique enables the neural network weight matrices to be estimated online very quickly based on the tracking error signal and without utilizing any prior training period. This technique is particularly useful for real-time closed-loop control [24].

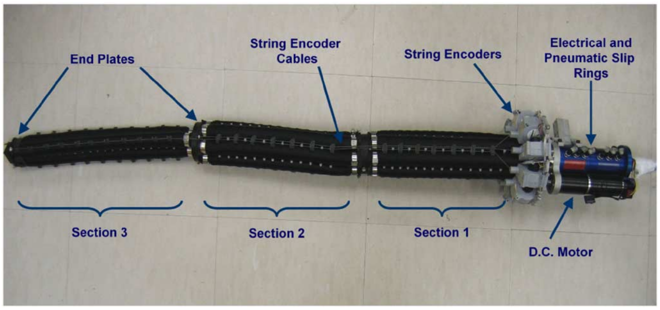

Fig. 1. OCTARM VI robotic manipulator.

The remainder of the paper is organized as follows. In Section II, the sensing and actuation of the OCTARM VI, which is a soft extensible continuum robot manipulator (see Fig. 1) are briefly discussed. In Section III, we discuss some of the prior efforts to develop dynamic models for continuum robot manipulators and present a model for the manipulator. In Section IV, the control objective is explicitly defined and the design of a controller with the neural network feedforward component is presented. To demonstrate the performance of the proposed controller with the neural network feedforward component, the controller was tested on the OCTARM. In Section V, the performance of the controller without the neural network feedforward component is compared with that of the controller with the neural network feedforward component to illustrate the effectiveness of the proposed strategy.

II. Robot Sensing and Actuation

The OCTARM VI manipulator [2], [7], [25] is a biologically inspired soft continuum manipulator resembling a trunk or tentacle. The OCTARM is significantly more versatile and adaptable than conventional robotic manipulators, capable of adaptive and dynamic manipulation in unstructured environments. To provide the desired dexterity, the OCTARM VI is constructed with high-strain extensor air muscles called McKibben actuators. These actuators are constructed by covering latex tubing with a double-helical weave, plastic mesh sheath [26]. These actuators provide a large strength to weight ratio and strain, which are required for soft continuum manipulators.

The OCTARM VI (refer to Fig. 1) is divided into three sections, with each section consisting of three McKibben actuators. Each section is capable of two axis bending and extension, hence allowing nine total degrees of freedom for the manipulator. The manipulator is pneumatically actuated through nine pressure regulators, which maintain the pressure in the actuators at a desired value set using an input voltage. The pressure regulators provide a linear relationship between the control voltage and the air pressure. By varying the air pressure to the actuators on a section, the length and shape of the section can be controlled.

The shape of the manipulator can be inferred in terms of curvatures and extensions by measuring the length of each of the nine actuators and using the forward kinematics described in [27]. In this work, we are only concerned with developing an efficient low level controller, which regulates the length of each of the actuators on the manipulator to follow a desired trajectory.[1]

III. Robot Dynamic Model

From a review of current literature, it is evident that the theory related to the dynamic modeling of continuum robot arms is still in its nascence with few published works available. Some of the previous research includes [28]–[30], where planar models of the continuum structure were considered and [31], [32], where the authors develop a three-dimensional dynamic model for an inextensible (constant length) continuum manipulator. As such, the complete dynamic modeling of variable length continuum robot arms remains an open research area. In [31], the developed dynamic model was shown to satisfy the familiar property that the continuum manipulators inertia matrix is symmetric and positive definite. With this in mind, in the development being considered, we will assume that the dynamic model of a 9 degree of freedom (DOF) continuum robot manipulator can be described by the following expression

$$M(q)\ddot{q} + N(q,\dot{q}) = u \tag{1}$$

where $M(q) \in \mathbb{R}^{9\times 9}$ represents the inertia matrix, $N(q,\dot{q}) \in \mathbb{R}^9$ represents the remaining dynamic terms such as centripetal, coriolis, and frictional forces, $u(t) \in \mathbb{R}^9$ represents the control input vector, and $q(t),\ \dot{q}(t),\ \ddot{q}(t) \in \mathbb{R}^9$ represent the actuator length, velocity, and acceleration, respectively.

The subsequent development is based on the following assumptions

Assumption 1: The manipulators position $q(t)$ and velocity $\dot{q}(t)$ are measurable.

Assumption 2: The dynamic terms denoted by $M(q)$ and $N(q,\dot{q})$ are unknown nonlinear functions of $q(t)$ and $\dot{q}(t)$ which are second-order differentiable and satisfy the following properties

$$M(\cdot), \dot{M}(\cdot), \ddot{M}(\cdot) \in \mathcal{L}_\infty \qquad \textit{if } q(t), \dot{q}(t), \ddot{q}(t) \in \mathcal{L}_\infty \tag{2}$$

$$N(\cdot), \dot{N}(\cdot), \ddot{N}(\cdot) \in \mathcal{L}_\infty \qquad \textit{if } q(t), \dot{q}(t), \ddot{q}(t), \dddot{q}\ (t) \in \mathcal{L}_\infty. \tag{3}$$

Assumption 3: The inertia matrix $M(q)$ is symmetric and positive-definite and satisfies the following inequalities

$$m_1 \|\xi\|^2 \le \xi^T M(q)\xi \le m_2 \|\xi\|^2 \qquad \forall \xi \in \mathbb{R}^9 \tag{4}$$

where $m_1, m_2 \in \mathbb{R}$ are positive constants and $\|\cdot\|$ denotes the standard Euclidean norm.

IV. Control Design

As we have mentioned in the previous section, the dynamic modeling of extensible continuum manipulators remains an open research problem. The main focus of the control design in this section is to develop an efficient tracking controller, which can deal with the high level of uncertainty in the structure of continuum robot's dynamic model. The proposed control strategy consists of a neural network feedforward component and a nonlinear feedback component. The feedback component is based on prior work in [23], this controller is chosen because it leaves a lot of flexibility in the design of the neural network feedforward component. The neural network feedforward component is then designed, based on the back propagation algorithm in [16], to meet the boundedness requirements required by the feedback controller.

A. Feedback Controller

The control objective is to design a continuous controller, which provides asymptotic tracking of the actuator lengths and the desired actuator length trajectories in the sense that

$$q(t) \to q_d(t) \text{ as } t \to \infty. \tag{5}$$

To quantify the control objective, an error signal denoted by $e_1(t) \in \mathbb{R}^9$, is defined as follows:

$$e_1 \triangleq q_d - q. \tag{6}$$

Furthermore, an auxiliary tracking error signal $e_2(t) \in \mathbb{R}^9$ is defined as follows:

$$e_2 \triangleq \dot{e}_1 + \lambda_1 e_1 \tag{7}$$

where $\lambda_1 \in \mathbb{R}^+$ is a control gain. For the closed-loop error system development, we define a filtered tracking error signal $r(t) \in \mathbb{R}^9$ as follows:

$$r \triangleq \dot{e}_2 + \lambda_2 e_2 \tag{8}$$

where $\lambda_2 \in \mathbb{R}^+$ is a control gain.

The dynamic model of the continuum robot is highly nonlinear and has an uncertain structure; hence, the strategy developed by Xian *et al.* [23] can be utilized for the controller. This controller is chosen because it is continuous, it does not require the dynamic model of the manipulator to be known, and yet, it provides semiglobal asymptotic tracking. Specifically, the control objective described in (5) can be met with the following controller [23]

$$u(t) \triangleq (K_s + I)e_2(t) - (K_s + I)e_2(t_0) + \int_{t_0}^{t} \hat{f}(\tau)d\tau + \int_{t_0}^{t} \left(\lambda_2(K_s + I)e_2(\tau) + \beta\,\mathrm{sgn}(e_2(\tau))\right)d\tau \tag{9}$$

where $u(t) \in \mathbb{R}^9$ is the control input defined in (1), $\lambda_2 \in \mathbb{R}^+$ is a control gain, $K_s,\ \beta \in \mathbb{R}^{9\times 9}$ are positive definite diagonal control gain matrices, $\hat{f}(t) \in \mathbb{R}^9$ is the neural network feedforward component, and $\mathrm{sgn}(\cdot) : \mathbb{R}^9 \mapsto \mathbb{R}^9$ denotes the vector signum function defined as $\mathrm{sgn}(\xi) = [\mathrm{sgn}(\xi_1), \ldots, \mathrm{sgn}(\xi_9)]^T\ \forall\, \xi = [\xi_1, \ldots, \xi_9]^T \in \mathbb{R}^9$. As shown in [23], the controller presented in (9) provides semiglobal asymptotic convergence of the actuator length tracking error (i.e., $\|e_1(t)\| \to 0$ as $t \to \infty$). For brevity in this presentation the stability analysis is omitted, for a complete analysis of the controller the reader is referred to [23].

Remark 1: The design of the neural network feedforward component, $\hat{f}(t)$ is presented in the subsequent section. The only restriction imposed on the neural network component by the selection of the feedback controller in (9) is that $\hat{f}(t) \in \mathcal{L}_\infty$, i.e., the output from the neural network must be bounded.

B. Neural Network Feedforward Design

The neural network feedforward component $\hat{f}(t) \in R^9$ is computed using a two-layer network with 15 neurons. The weights are computed using a modified version of the back propagation algorithm presented

[1] Note that the desired trajectory for the lengths of each actuator could be specified directly in terms of the lengths or could be generated using the inverse kinematics described in [27].

in [16]. Given Remark 1, an important consideration regarding the design of the neural network feedforward component is that the output from the neural network must always be bounded (i.e., $\hat{f}(t) \in \mathcal{L}_\infty$). To this end, the neural network component is defined as follows:

$$\hat{f} = \hat{W}^T \bar{\sigma}\left(\hat{V}^T x\right). \tag{10}$$

where $\hat{W}(t) \in \mathbb{R}^{15\times 9}$ and $\hat{V}(t) \in \mathbb{R}^{37\times 15}$ are estimated weight matrices and $x(t) \in \mathbb{R}^{37}$ is the input vector to the neural network, which is selected as

$$x = \begin{bmatrix} 1, & q_d^T, & \dot{q}_d^T, & \ddot{q}_d^T, & \dddot{q}_d^T \end{bmatrix}^T \tag{11}$$

where $q_d(t), \dot{q}_d(t), \ddot{q}_d(t), \dddot{q}_d(t)$ were defined *a priori*. The vector activation function $\bar{\sigma}(\cdot) \in \mathbb{R}^{15} \mapsto \mathbb{R}^{15}$ is defined as follows:

$$\bar{\sigma}(\omega) = \left[\sigma(\omega_1), \sigma(\omega_2), \ldots, \sigma(\omega_{15})\right]^T \tag{12}$$

where $\omega = [\omega_1, \omega_2, \ldots, \omega_{15}]^T$ and $\sigma(s) : \mathbb{R} \mapsto \mathbb{R}$ is the sigmoid activation function defined as

$$\sigma(s) = \frac{1}{1+\exp(-s)}. \tag{13}$$

The gradient of the vector activation function, denoted by $\bar{\sigma}'(\cdot) \in \mathbb{R}^{15\times 15}$ can be expressed in closed form as follows [16]:

$$\bar{\sigma}(\omega)' = \operatorname{diag}\{\bar{\sigma}(\omega)\}\left[I - \operatorname{diag}\{\bar{\sigma}(\omega)\}\right]. \tag{14}$$

If we were to design the weight update laws according to the augmented backpropagation algorithm [16], we would use the following update rules:

$$\dot{\hat{W}} = -\kappa F \|r\| \hat{W} - F\bar{\sigma}'(\cdot)\hat{V}^T x r^T + F\bar{\sigma}(\cdot) r^T$$

$$\dot{\hat{V}} = -\kappa G \|r\| \hat{W} + Gx\left(\bar{\sigma}'^T(\cdot)\hat{W} r\right)^T$$

where $\kappa \in \mathbb{R}^+$ is selected to be a small constant, $F \in \mathbb{R}^{15\times 15}$, $G \in \mathbb{R}^{37\times 37}$ are positive definite gain matrices, $x(t)$ is the input vector defined in (11), and $r(t)$ is the filtered tracking error signal defined in (8). Here, the filtered tracking error signal $r(t)$ is required in the update laws, which requires the measurement of the acceleration $\ddot{q}(t)$, and hence, is undesirable. To ensure that the weights generated from these laws are bounded and that acceleration measurements are not required, we redefine the update laws as follows:

$$\dot{\hat{W}} = -\alpha_w \hat{W} + \gamma_1 \bar{\sigma}(\hat{V}^T x)\,\mathrm{sat}\,(e_2 + \zeta)^T \tag{15}$$

$$\dot{\hat{V}} = -\alpha_v \hat{V} + \gamma_2 x[\bar{\sigma}'(\hat{V}^T x)\hat{W}\,\mathrm{sat}\,(e_2 + \zeta)]^T \tag{16}$$

where $\alpha_v, \alpha_w \in \mathbb{R}^+$ are small constants, $\gamma_1, \gamma_2 \in \mathbb{R}^+$ are control gains which effect the learning speed, the function $\mathrm{sat}\,(\xi) : \mathbb{R}^9 \mapsto \mathbb{R}^9$ is defined as $\mathrm{sat}\,(\xi) = [\mathrm{sat}\,(\xi_1), \ldots, \mathrm{sat}\,(\xi_9)]^T \;\forall\, \xi = [\xi_1, \ldots, \xi_9]^T \in \mathbb{R}^9$ where $\mathrm{sat}\,(\xi_i) \in \mathbb{R}\;\forall\, i = 1, \ldots, 9$ is the following saturation function

$$\mathrm{sat}\,(\xi_i) = \begin{cases} -\xi_{\min}, & \text{if } \xi_i \le -\xi_{\min} \\ \xi_i, & \text{if } \xi_i > -\xi_{\min} \quad \text{or} \quad \xi_i < \xi_{\max} \\ \xi_{\max}, & \text{if } \xi_i \ge \xi_{\max} \end{cases}$$

where $\xi_{\min}, \xi_{\max} \in \mathbb{R}^+$ are constants. In (15) and (16), the auxiliary signal $\zeta(t) \in \mathbb{R}^9$ is an approximation (i.e., a dirty derivative operation) for the signal $\dot{e}_2(t)$, which is defined as follows:

$$\zeta = \frac{1}{\varepsilon}(e_2 - \eta) \tag{17}$$

where $\varepsilon \in \mathbb{R}^+$ is a small constant and the signal $\eta(t) \in \mathbb{R}^9$ is updated according to the following expression:

$$\dot{\eta} = \frac{1}{\varepsilon}(e_2 - \eta). \tag{18}$$

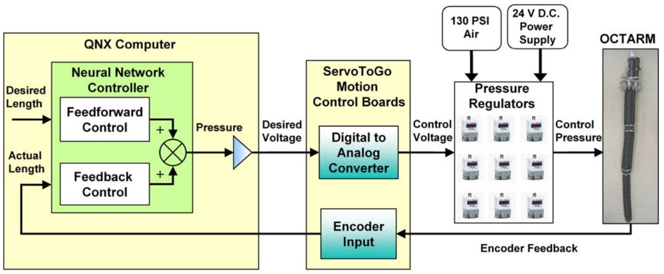

Fig. 2. Block diagram showing an overview of the OCTARM VI control.

From (10) to (18) and the fact that the input vector to the neural network is bounded, it is easy to show that the weight matrices $\hat{W}(t)$ and $\hat{V}(t)$ are bounded, and hence, the output from the neural network, $\hat{f}(t)$ is bounded.

V. Experimental Results

To verify the performance of the controller with the neural network feedforward component, the controller was implemented on the OCTARM VI continuum robot manipulator. In this section, we first provide a description of the control system for the OCTARM VI continuum robot manipulator, then experimental results are described, which illustrate the effectiveness of the neural network feedforward tracking controller.

A. *OCTARM Control System*

Fig. 2 shows an overview of the control system of the OCTARM. The control system consists of a commercial off-the-shelf Pentium III EBX form-factor single board computer (SBC) with two servotogo data acquisition boards, which provide analog and digital I/O. The computer runs the QNX Neutrino real-time operating system and QMotor 3.0 [33], a real-time control software, which facilitates online parameter tuning and data logging for the implemented control programs. Data acquisition and closed-loop control were performed at a frequency of 500 Hz. There are nine pressure regulators, one for each actuator on the manipulator. The air pressure for each actuator is determined by the neural network controller. The pressure specified by the controller is converted to a corresponding voltage level (using a linear relationship between pressure and voltage) and this drives the pressure regulators, which control the air flow to the actuators.

For closed-loop control of the OCTARM VI, manipulator accurate sensing of the actuator lengths is essential. To measure the length of each of the nine actuators, there are nine string encoders arranged around the base of section one (see Fig. 1). The cables from each of the string encoders run the entire length of the actuator they are assigned to measure. This configuration enables the length of each of the actuators on the OCTARM VI manipulator to be determined. As the string encoders have a relatively low resolution, velocities obtained through differentiation of the position measurements are noisy; hence, a variable structure velocity observer [23] was utilized to obtain estimates of the velocity.

B. *Trajectory Tracking Experiment Description*

To test the low level controller with the neural network component given in (9), a sinusoidal trajectory was selected for the actuator lengths. As the three actuators on a section are 120° apart mechanically, the desired trajectory for each actuator in a section is shifted 120° in phase from the trajectory of the previous actuator in that section. The trajectory for section one was selected as follows:

$$\begin{aligned} q_{d1k} &= l_{\min_1} + (1-\exp(-0.5t))l_1 \\ &\quad + 3(1-\exp(-0.5t))\sin\left(0.03\pi t + \frac{2}{3}\pi k\right) \end{aligned}$$

where $k = 1, 2, 3$, $q_{d1} = [q_{d11}, q_{d12}, q_{d13}] \in \mathbb{R}^3$ represents the desired trajectory for the actuators in section one, $l_{\min_1} \in \mathbb{R}$ represents the minimum lengths of the actuators in section one, and $l_1 = 8\,(\text{cm})$ is the initial extension of the actuators in section one. The initial extension was selected to keep the operating pressure close to its nominal value. For sections two and three, the initial extensions were $l_2 = 7\,(\text{cm})$, $l_3 = 5\,(\text{cm})$, and the frequency of the sinusoidal trajectory for each section was twice that of the previous section. The minimum and maximum lengths of the sections corresponding to the minimum pressure (0 psi) and maximum pressure (130 psi), respectively, are physical limitations of the actuators and were found to be $l_{\min_1} = 28\,(\text{cm}), l_{\min_2} = 26.5\,(\text{cm}), l_{\min_3} = 32.5\,(\text{cm}), l_{\max_1} = 42\,(\text{cm}), l_{\max_2} = 449\,(\text{cm}), l_{\max_3} = 54\,(\text{cm})$.

C. Control Parameter Tuning

To test the effectiveness of the neural network feedforward component, we compared the performance of the controller in (9) with and without the neural network component. The performance of the system was first tested without the neural network component. The control gains for the feedback controller were adjusted in such a manner as to obtain a fast system response, while minimizing the overshoot and oscillations, as the actuators have very low damping. The control gains, which gave the best performance were as follows:

$$\begin{aligned} K_s &= \text{diag}\{1,1,1,1,1,1,1,1,1\}, \quad \lambda_1 = 1, \quad \lambda_2 = 1 \\ \beta &= \text{diag}\{1,1,1,1,1,1,0.5,0.5,0.5\}. \end{aligned}$$

It is interesting to note that, when β and $\hat{f}(t)$ in (9) are zero, the control law is essentially a PID controller, and hence, the parameters $K_s, \lambda_1, \lambda_2$, can be tuned in a conceptually similar manner to a PID controller. However, the addition of the control gain β into the controller provides a significant improvement in the tracking performance. Also, the last three elements of the β vector, corresponding to the third section of the manipulator are smaller, as the third section requires less control effort to extend due to its relatively smaller size.

With the feedback controller now tuned, the neural network feedforward was added to the controller, while keeping the gains on the feedback controller the same. The number of neurons required for the system was determined experimentally by noting the performance achievable with a given number of neurons, and then, increasing the number of neurons until the desired tracking performance level was obtained. Initially, 5 neurons were utilized and this was subsequently increased to 15 to get better performance. Note that we cannot keep increasing the number of neurons, as we require the control algorithm to be as computationally efficient as possible for it to run in real time on the SBC. There was no offline training period utilized for the neural network feedforward component, the weight matrices $\hat{W}(t)$ and $\hat{V}(t)$ in the neural network were initialized to zero. The gains in the neural network weight update law were adjusted to be as follows:

$$\gamma_1 = 10, \quad \gamma_2 = 500, \quad \alpha_w = 0.001, \quad \alpha_v = 0.001.$$

The constants α_w and α_v (sometimes called forgetting factors) were kept small to provide stability to the weight estimation laws. The constants γ_1 and γ_2 affect the weight adaptation speed. It was interesting to note that γ_2 was required to be much larger than γ_1 to provide stable, yet fast convergence of the weights. This is because the gain γ_2 only affects the inner layer of the neural network, whereas γ_1 would affect the output layer. The constant ε must be close to zero and was selected as $\varepsilon = 0.01$.

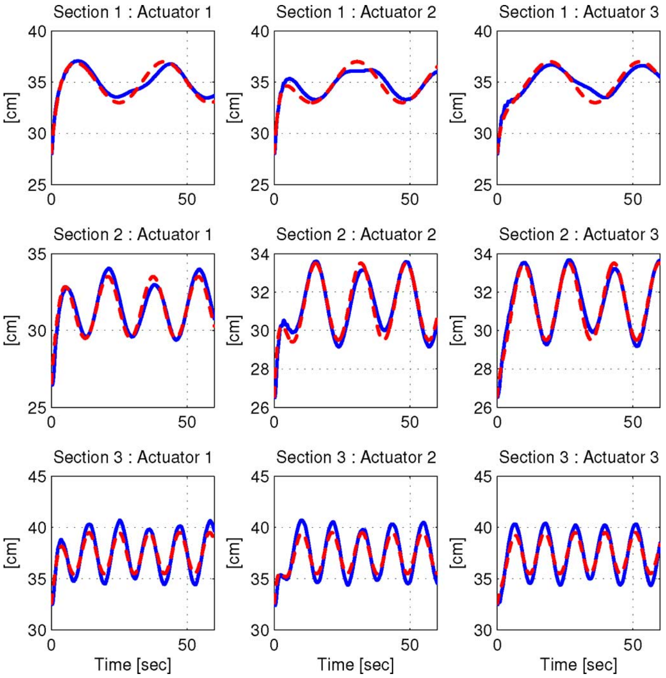

Fig. 3. Actual and desired length trajectory without neural network component, solid line represents the actual length trajectory, dashed line represents the desired length trajectory.

D. Results and Discussion

To provide a means to quantify the performance of the controller for each configuration, we computed the following measures:

$$M_e \triangleq \int_{t_0}^{t} \|e_1(\tau)\|^2 d\tau \tag{19}$$

$$M_u \triangleq \int_{t_0}^{t} \|u(\tau)\|^2 d\tau \tag{20}$$

where $M_u(t)$ is a measure of the energy expended by the controller and $M_e(t)$ is a measure of the magnitude of the tracking error over the period of operation of the system.

Figs. 3–5 show the actual and desired length trajectories, tracking error, and the input pressure for the controller without the neural network component. Figs. 6–8 show the actual and desired length trajectories, tracking error, and the input pressure, for the controller with the neural network feedforward component. It can be seen from Fig. 7 that the tracking error with the neural network feedforward component settles out to ±0.3 (cm) in under 5 s.

To compare the controller performance with and without the neural network feedforward component, the measures (19) and (20) were computed for the two configurations. Table I shows a comparison of the performance for these two controller configurations. It can be seen clearly from Table I that although the control input measure remains almost the same, the error measure shows a significant drop when the neural network feedforward component is introduced. Thus, we can conclude that improved tracking performance is achieved by adding the neural network feedforward to the controller.

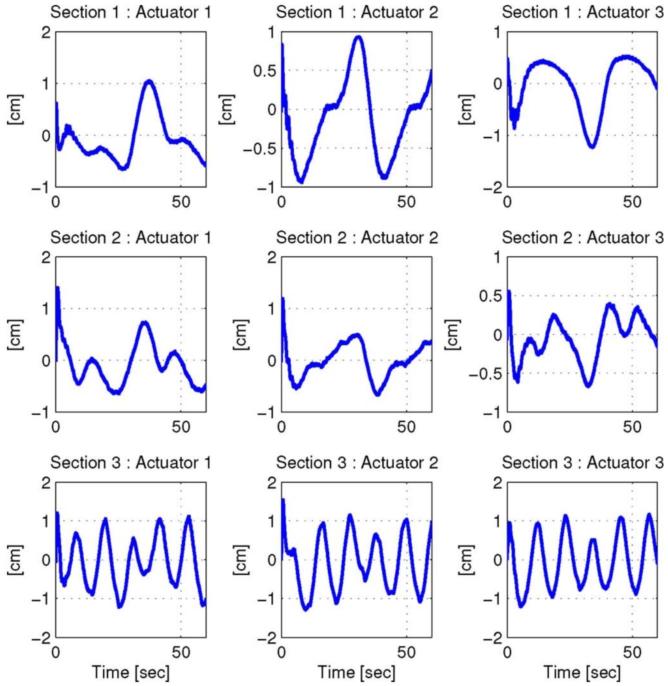

Fig. 4. Tracking error without neural network component.

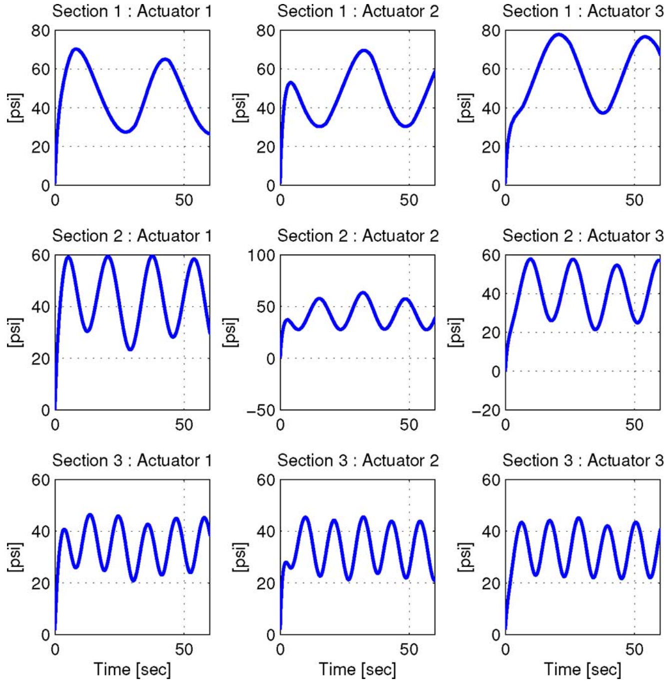

Fig. 5. Control pressure without neural network component.

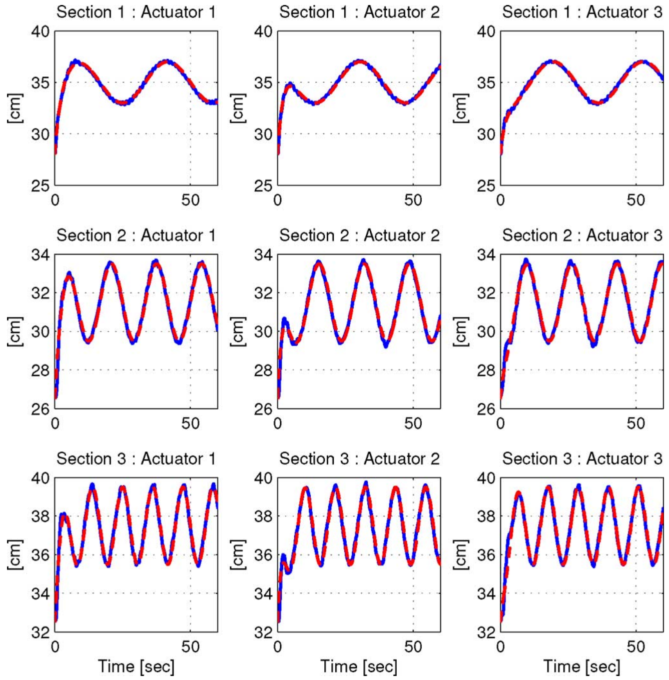

Fig. 6. Actual and desired length trajectory with neural network component, solid line represents the actual length trajectory, dashed line represents the desired length trajectory.

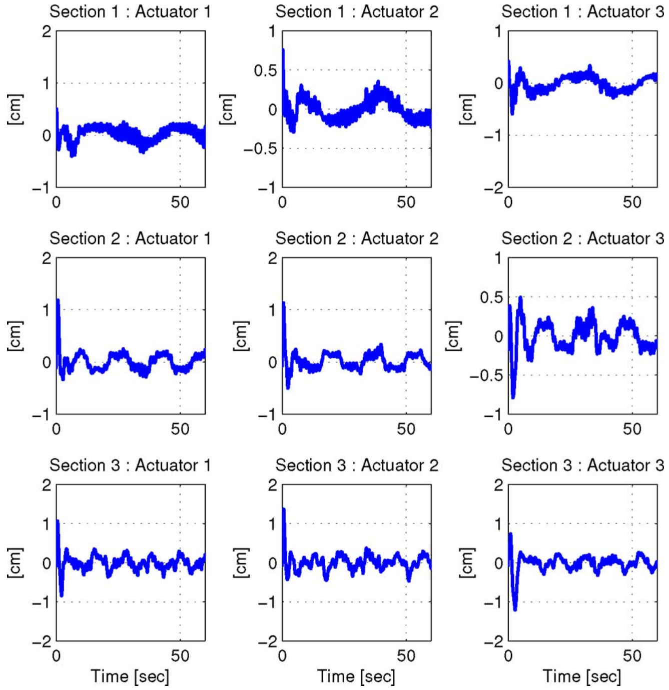

Fig. 7. Tracking error with neural network component.

The controller described in this paper has been implemented on the OCTARM VI control system replacing the original PID controller. The group has adapted the controller as the default controller for the OCTARM series of robots. It has been utilized for a number of applications, such as whole arm grasping, inspection, and tunnel navigation, where it has shown excellent performance. All of these tasks were performed by an operator commanding the continuum arm's motion using a joystick interface. The tunnel navigation task is one application, where the improved tracking performance provided by the neural network controller significantly reduces the time taken by the operator to navigate down the tunnel. Through our experimentation, we have noted that the fine motion control possible with the controller is extremely useful for the tasks mentioned earlier, as it enables all the capabilities of the continuum arm to be explored and exploited.

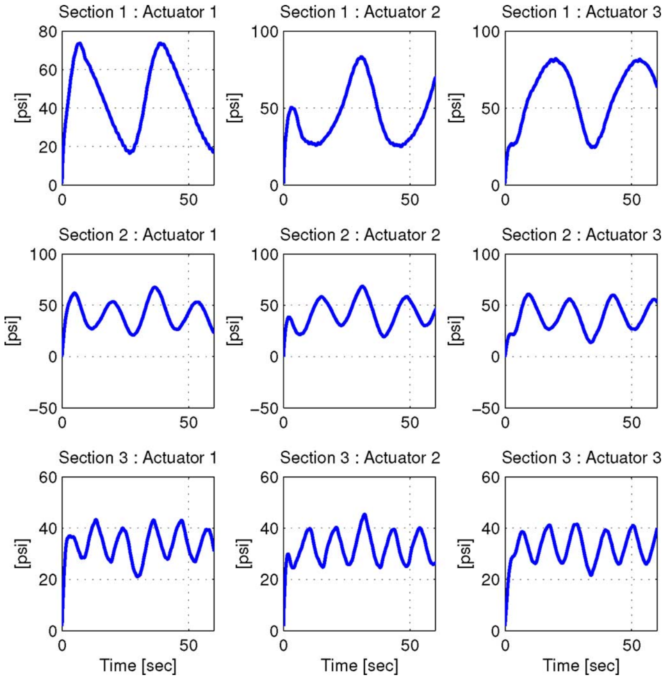

Fig. 8. Control pressure with neural network component.

TABLE I
COMPARISON OF MEASURES FOR THE CONTROLLER WITH AND WITHOUT THE FEEDFORWARD COMPONENT

	Without feedforward	**With feedforward**
M_e	145.51935	17.7180
M_u	1.0463×10^6	1.0301×10^6

VI. CONCLUSION

In this paper, we have presented a neural network-based tracking controller for continuum robot manipulators. Unlike previous works, the proposed approach does not require that an accurate dynamic model of the continuum manipulator be known. The feedforward neural network estimation scheme was used to compensate for the uncertain nonlinear dynamics of the continuum manipulator. Experimental results for the OCTARM continuum robot manipulator operating along a sinusoidal trajectory were presented to demonstrate the performance improvement provided by the proposed neural network feedforward estimation technique. As the proposed approach is not dependent on any specific model, it can be easily adapted to a wide range of continuum arm designs.

REFERENCES

[1] G. Robinson and J. B. C. Davies, "Continuum robots—A state of the art," in *Proc. IEEE Int. Conf. Robot. Automat.*, Detroit, MI, 1999, pp. 2849–2854.

[2] I. D. Walker, D. M. Dawson, T. Flash, F. Grasso, R. Hanlon, B. Hochner, W. Kier, C. Pagano, C. D. Rahn, and Q. Zhang, "Continuum robot arms inspired by cephalopods," in *Proc. 2005 SPIE Conf. Unmanned Ground Veh. Technol. IV*, Orlando, FL, Mar., pp. 303–314.

[3] R. Cieslak and A. Morecki, "Elephant trunk type elastic manipulator—A tool for bulk and liquid materials transportation," *Robotica*, vol. 17, pp. 11–16, 1999.

[4] H. Tsukagoshi, A. Kitagawa, and M. Segawa, "Active hose: An artificial elephant's nose with maneuverability for rescue operation," in *Proc. IEEE Int. Conf. Robot. Automat.*, Seoul, Korea, 2001, pp. 2454–2459.

[5] M. A. Hannan and I. D. Walker, "Kinematics and the implementation of an elephant's trunk manipulator and other continuum style robots," *J. Robot. Syst.*, vol. 20, no. 2, pp. 45–63, Feb. 2003.

[6] J. Yang, E. P. Pitarch, J. Potratz, S. Beck, and K. Abdel-Malek, "Synthesis and analysis of a flexible elephant trunk robot," *Adv. Robot.*, vol. 20, no. 6, pp. 631–659, 2006.

[7] W. McMahan, V. Chitrakaran, M. Csencsits, D. M. Dawson, I. D. Walker, B. Jones, M. Pritts, D. Dienno, M. Grissom, and C. Rahn, "Field trials and testing of the octarm continuum manipulator," in *Proc. IEEE Int. Conf. Robot. Automat.*, Orlando, FL, 2006, pp. 2336–2341.

[8] S. Hirose, *Biologically Inspired Robots*. New York: Oxford Univ. Press, 1993.

[9] F. Matsuno and K. Suenga, "Control of redundant snake robot based on kinematic model," in *Proc. 41st SICE Annu. Conf.*, Osaka, Japan, 2002, pp. 1481–1486.

[10] M. Ivanescu, "Position dynamic control for a tentacle manipulator," in *Proc. IEEE Int. Conf. Robot. Automat.*, Washington, DC, 2002, pp. 1531–1538.

[11] M. Ivanescu, N. Popescu, and D. Popescu, "A variable length tentacle manipulator control system," in *Proc. IEEE Int. Conf. Robot. Automat.*, Barcelona, Spain, 2005, pp. 3285–3290.

[12] I. A. Gravagne and I. D. Walker, "Uniform regulation for a multisection continuum manipulator," in *Proc. IEEE Int. Conf. Robot. Automat.*, Washington, DC, 2002, pp. 1519–1524.

[13] T. Suzuki, K. Shintani, and H. Mochiyama, "Control methods of hyperflexible manipulators using their dynamical features," in *Proc. 41st SICE Annu. Conf.*, Osaka, Japan, 2002, pp. 1511–1516.

[14] H. Mochiyama, E. Shimemura, and H. Kobayashi, "Control of manipulators with hyper degrees of freedom: Shape tracking using only joint angle information," *Int. J. Syst. Sci.*, vol. 30, no. 1, pp. 77–85, 1999.

[15] F. Matsuno and H. Sato, "Trajectory tracking control of snake robots based on dynamic model," in *Proc. IEEE Int. Conf. Robot. Automat.*, Barcelona, Spain, 2005, pp. 3040–3045.

[16] F. L. Lewis, S. Jagannathan, and A. Yesildirek, *Neural Network Control of Robot Manipulators and Nonlinear Systems*. London, U.K.: Taylor & Francis, Jun. 1999.

[17] S. S. Ge, T. H. Lee, and C. J. Harris, *Adaptive Neural Network Control of Robotic Manipulators*. Singapore: World Scientific, Feb. 1999.

[18] Y. H. Kim and F. L. Lewis, "Neural network output feedback control of robot manipulators," *IEEE Trans. Robot. Automat.*, vol. 15, no. 2, pp. 301–309, Apr. 1999.

[19] F. C. Sun, Z. Q. Sun, R. J. Zhang, and Y. B. Chen, "Neural adaptive tracking controller for robot manipulators with unknown dynamics," *Inst. Electr. Eng. Proc. Control Theory Appl.*, vol. 147, no. 3, pp. 366–370, 2000.

[20] H. D. Patino, R. Carelli, and B. R. Kuchen, "Neural networks for advanced control of robot manipulators," *IEEE Trans. Neural Netw.*, vol. 13, no. 2, pp. 343–354, Mar. 2002.

[21] M. K. Ciliz, "Adaptive control of robot manipulators with neural network based compensation of frictional uncertainties," *Robotica*, vol. 23, pp. 159–167, 2005.

[22] D. Braganza, D. M. Dawson, I. D. Walker, and N. Nath, "Neural network grasping controller for continuum robots," in *Proc. IEEE Int. Conf. Decision Control*, San Diego, CA, 2006, pp. 6445–6449.

[23] B. Xian, D. M. Dawson, M. S. de. Queiroz, and J. Chen, "A continuous asymptotic tracking control strategy for uncertain nonlinear systems," *IEEE Trans. Automat. Control*, vol. 49, no. 7, pp. 1206–1211, 2004.

[24] O. M. Omidvar and D. L. Elliott, Eds., *Neural Systems for Control*. Boston, MA: Academic, 1997, ch. 7.

[25] W. McMahan, B. Jones, I. D. Walker, V. Chitrakaran, A. Seshadri, and D. Dawson, "Robotic manipulators inspired by cephalopod limbs," in *Proc. CDEN Symp. Biomimicry, Bionics Biomechanics*, Montreal, Canada, 2004, pp. 1–10.

[26] M. B. Pritts and C. D. Rahn, "Design of an artificial muscle continuum robot," in *Proc. IEEE Int. Conf. Robot. Automat.*, New Orleans, LA, 2004, pp. 4742–4746.

[27] B. A. Jones and I. D. Walker, "Kinematics for multisection continuum robots," *IEEE Trans. Robot.*, vol. 22, no. 1, pp. 43–55, Feb. 2006.

[28] I. Gravagne, C. Rahn, and I. D. Walker, "Large deflection dynamics and control for planar continuum robots," *IEEE/ASME Trans. Mechatron.*, vol. 8, no. 2, pp. 299–307, Jun. 2003.

[29] Y. Yekutieli, R. Sagiv-Zohar, R. Aharonov, Y. Engel, B. Hochner, and T. Flash, "Dynamic model of the octopus arm. i. biomechanics of the octopus arm reaching movement," *J. Neurophysiol.*, vol. 94, pp. 1443–1458, 2005.

[30] E. Tatlicioglu, I. D. Walker, and D. M. Dawson, "Dynamic modeling for planar extensible continuum robot manipulators," in *Proc. IEEE Int. Conf. Robot. Automat.*, Rome, Italy, 2007, pp. 1357–1362.

[31] H. Mochiyama and T. Suzuki, "Dynamic modeling of a hyper-flexible manipulator," in *Proc. 41st SICE Annu. Conf.*, Osaka, Japan, 2002, pp. 1505–1510.

[32] H. Mochiyama, "Kinematics and dynamics of a cable-like hyper-flexible manipulator," in *Proc. IEEE Int. Conf. Robot. Automat.*, Taipei, Taiwan, 2003, pp. 3672–3677.

[33] M. Loffler, N. Costescu, and D. M. Dawson, "Qmotor 3.0 and the qmotor robotic toolkit—An advanced pc-based real-time control platform," *IEEE Control Syst. Mag.*, vol. 22, no. 3, pp. 12–26, Jun. 2002.

Swing-Up Control for a 3-DOF Gymnastic Robot With Passive First Joint: Design and Analysis

Xin Xin
and Masahiro Kaneda

***Abstract*—This paper concerns a swing-up control problem for a three-link gymnastic planar robot in a vertical plane with its first joint being passive (unactuated) and the rest being active (actuated). The objectives of this paper are to: 1) design a controller under which the robot can be brought into any arbitrarily small neighborhood of the upright equilibrium point, where all three links of the robot remain in their upright positions; and 2) attain a global analysis of the motion of the robot under the controller. To tailor the energy-based control approach to achieve the aforementioned objectives, first, this paper considers the links 2 and 3 as a virtually composite link, and proposes a coordinate transformation of the angles of active joints. Second, this paper constructs a novel Lyapunov function based on the transformation, and devises an energy-based swing-up controller. Third, this paper carries out a global analysis of the motion of the robot under the controller, and establishes some conditions on control parameters for achieving the swing-up control objective. To validate the theoretical results obtained, this paper provides simulation results for a three-link robot with its mechanical parameters being obtained from a human gymnast.**

***Index Terms*—3-DOF, energy-based control, swing-up control, the Lyapunov stability theory, underactuated robots.**

I. Introduction

Studies on underactuated mechanical systems, which possess fewer actuators than degrees of freedom (DOF), have received considerable interest in recent years [1]–[9], [14]. Since these systems usually have nonholonomic second-order constraints, their control problems are challenging [15].

The Acrobot [10], [17] is a highly simplified model of a human gymnast on a high bar, where the underactuated first joint models the gymnast's hands on the bar, and the actuated second joint models the gymnast's hips. A skilled gymnast pointed out that to achieve an effective swing, the shoulders take a more important role than do the hips. Thus, to mimic gymnastic routine more realistically and to understand the control mechanism inside the routine better, one should model the gymnast on a high bar at least as a 3-DOF underactuated robot, that is, the gymnast's shoulders should be modeled as an actuated joint besides the hips.

There are many researches on the swing-up control of 2- and 3-DOF pendulum-type robots [8], [10], [12], [13], [17], [18]. However, the complete analysis of the swing-up control of 3-DOF pendulum-type robots has not been reported [13], [18], [20].

This paper studies a swing-up control problem of a three-link planar robot in a vertical plane with its first joint being passive (unactuated), and the second and third joints being active (actuated), which is the three-link gymnastic robot studied in [13] and [18]. We call this kind of robot a passive active active (PAA) robot. Different from the existing results in [9], [13], and [18], the objectives of this paper are to: 1) design a controller under which the robot can be brought to any arbitrarily small neighborhood of the upright equilibrium point, where all three links remain in their upright positions; and 2) attain a global analysis of the motion of the robot under the controller. Once the first objective is realized, the capturing and balancing phase can be accomplished easily.

We investigate how to apply the energy-based control approach, which has been developed in the seminal works of [7], [12], and [18], to solve the swing-up control problem of PAA robot. The swing-up control problem with PAA robot can be reduced to that with the Acrobot (for example, in [21] and [22]) by using one of the two actuators to fix 1-DOF of PAA robot. However, since inherited features of this three-link robot are not utilized in such a control strategy, a skillful maneuver or high actuation efficiency may not be expected.

Usually, different underactuated systems do not share the same difficulties from the control point of view [5]. We found that the application of the energy-based control approach to controlling PAA robot is not straightforward. To tailor the energy-based control approach to PAA robot, we first consider the links 2 and 3 as a virtually composite link (VCL), and propose a coordinate transformation of the angles of joints 2 and 3. Second, we construct a new Lyapunov function based on the transformation, and provide an energy-based swing-up controller. Third, we carry out a global analysis of the motion of the PAA robot under the controller, and establish conditions on control parameters for achieving the aforementioned control objective. Finally, to validate our obtained theoretical results, we present some simulation results for the PAA robot whose mechanical parameters are obtained from a human gymnast in [19].

II. Model of the PAA Robot

Consider the PAA robot shown in Fig. 1, where for the ith ($i = 1, 2, 3$) link, m_i is its mass, l_i is its length, l_{ci} is the distance from joint i to its center of mass (COM), and I_i is the moment of inertia around its COM.

Partition the generalized coordinate vector q as

$$q = \begin{bmatrix} q_1 & q_a^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}, \quad \text{with } q_a = \begin{bmatrix} q_2 & q_3 \end{bmatrix}^{\mathrm{T}}$$

where the subscript a denotes the state "active." The motion equations of the PAA robot are [18]

$$M_{11}\ddot{q}_1 + M_{1a}\ddot{q}_a + H_1 + G_1 = 0 \qquad (1)$$

$$M_{a1}\ddot{q}_1 + M_{aa}\ddot{q}_a + H_a + G_a = \tau \qquad (2)$$

Manuscript received May 21, 2007. This paper was recommended for publication by Associate Editor E. Papadopoulos and Editor H. Arai upon evaluation of the reviewer's comments. This work was supported in part by a Grant-in-Aid Scientific Research (C) under Grant 19560452, in part by the WESCO Scientific Promotion Foundation, and in part by the Electric Technology Research Foundation of Chugoku.

The authors are with the Department of Communication Engineering, Faculty of Computer Science and System Engineering, Okayama Prefectural University, Okayama 719-1197, Japan (e-mail: xxin@c.oka-pu.ac.jp; kaneda@c.oka-pu.ac.jp).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TRO.2007.909805

1552-3098/$25.00 © 2007 IEEE