2011 IEEE/RSJ International Conference on
Intelligent Robots and Systems
September 25-30, 2011. San Francisco, CA, USA

Task-Space Control of Extensible Continuum Manipulators

Apoorva Kapadia† and Ian D. Walker

Abstract— **In this paper, we present a new approach towards the control of continuous backbone (continuum) "trunk and tentacle" robots. Development of model-based control algorithms for this new and emerging class of robots has been relatively slow due to the inherent complexity of their mathematical models. Based on the recently developed kinematics, velocity Jacobian and full dynamic model, a simple nonlinear task-space controller, established for rigid-link robots, is adapted and extended for continuum manipulators for the regulation of its tip or any location along its backbone in the task-space. This approach is applicable to all continuum robots with extension/contraction and bending capabilities. Simulation results are shown using a three-section, six degree-of-freedom planar continuum robot.**

I. Introduction

Continuum or continuous backbone robots [1], [2], [3], [4], [5] have recently been the subject of much attention [6], [7], [8]. The ability of continuum backbones to bend at any point along their structure, together with their inherent compliance, offers continuum robots the potential to perform functions not feasible with conventional robots. In particular, novel modes of grasping and manipulation, using the robots to adapt their shape and "wrap around" environmental objects of a wide variety of shapes, sizes, and physical properties [9], as seen in Figure 1, have been widely proposed. Initial results from simple experimentation with continuum robot hardware [10], [11] have underlined the strong potential of this approach to transform the nature of robotic grasping and manipulation.

In the past few years, significant progress has been made in the modeling of continuum robots. General kinematic models have been developed [12], [13], [14], and research in dynamics initially developed in [15] is active [16], [17] and ongoing [18], [19]. The results have been successfully applied to a variety of continuum robot hardware [8], [10], [20]. Additional work has established core results in the development of configuration-space controllers of continuum robots [21], [22], [23], [24]. However, works considering of manipulation using continuum robots are scarce [11], [25], [26]. This appears to be largely due to two main current deficiencies in the literature: (1) a concentration in the research thus far on configuration space models (the transformation to task-space - critical for manipulation - is non-trivial for continuum robot grasping); and (2) lack of effective control strategies for continuum robot hardware, making it difficult to realize desired motion plans in practice. In this paper, we introduce a new approach to task-space continuum robot control aimed at enabling 'whole arm' continuum grasping and manipulation.

Fig. 1. The Octarm Grasping a Cylindrical Object

The work in this paper is the first, to the best of our knowledge, to consider task-space control (i.e. control driven by feedback in the task-space) for continuum robots. We show that somewhat parallel to the case for conventional rigid-link robots, task-space feedback, when suitably chosen, is effective for continuum robots. Moreover, and in strong contrast to the case for conventional robots, our formulation enables the location of the controlled task-space coordinates to vary along the robot backbone. This is critical for enabling practical "whole arm grasping" of the type long-postulated for continuum robots.

The issue of variable task-space coordinates is an important one for continuum robot grasping and manipulation. Note that for conventional robot manipulators equipped with parallel jaw grippers, "grasping" amounts to the discrete event of closing the gripper. Grasping control is therefore a binary "success/fail" discrete event, and (subsequent) task-space manipulation is entirely an issue of controlling the end-effector coordinates in which environmental objects are fixed. For whole arm grasping however, grasping, as defined by the acquisition of sufficient physical influence over an object to constrain its movements in given direction(s), is a process, not an event. Contact is made over a range of locations on the robot, and the key coordinates that have to be controlled during the grasp, change over time. Grasping is also more closely coupled with manipulation, which is

† To whom all correspondence should be addressed.
A. Kapadia and I. D. Walker are with the Department of Electrical & Computer Engineering, Clemson University, Clemson, SC 29634-0915 ((akapadi, iwalker)@clemson.edu). This research was supported by the U.S. National Science Foundation under grants IIS-0844954 and IIS-0904116.

no longer solely a matter of controlling the configuration of the robot, under the assumption that the grasped object remains fixed with respect to the robot. In the following, we introduce and demonstrate the effectiveness of a new task-space controller for continuum robots which simultaneously combines grasping and manipulation components.

II. System Model

A. The Task-Space

As noted above, for continuum robot grasping and manipulation, the key task-space coordinates, unlike the case of conventional robots, are not typically located at the tip (distal end) of the robot. The key physical advantage of continuum robots is the ability to use any location of their structure to grasp and manipulate objects, and task-space control strategies should reflect this. This requirement not only results in added complexity in the analysis, but also in a novel controller structure.

First, we define the task-space coordinates $x(d,t) \in \mathbb{R}^m$, where $t \in \mathbb{R}^+$ represents time and $d(t) \in \mathbb{R}$ represents the arc length specifying location along the backbone of the robot. More specifically, in the context of this paper, $x(d,t) \in \mathbb{R}^3$ such that $x(d,t) = [X, Z, \psi_Y]$, where $X(t)$, and $Z(t) \in \mathbb{R}$ represent the manipulator end-effectors x-axis and z-axis coordinates respectively, and $\psi_Y \in \mathbb{R}$ represents the y-axis orientation of the manipulator. This allows the task-space to be located arbitrarily on the robot. We further assume that locations of contact between the robot and the environment can be sensed. For example, the local sensors can be embedded in the robot, or an external vision system can be used to sense the location of any point on the backbone in real time[1]. While the coordinates $x(d,t)$ are arbitrary in general, in this paper we will restrict $x(d,t)$ to the tip of the robot and to locations of contact between the robot and the environment.

The controller structure will combine two components: (a) movements of the robot which cause $\dot{x}(d,t) \neq 0$; and (b) movements of the robot which cause $\dot{x}(d,t) = 0$. Intuitively, we associate those movements in (a) with manipulation; and those in (b) with grasping about the fixed point of contact at $x(d,t)$. In practice, more complex and more interesting situations can occur, but the above intuition will suffice to demonstrate the controller herein.

B. Kinematic Model

Based on the kinematic model developed by Jones and Walker [14], the end-effector position and orientation in the task-space, $x(d,t) \in \mathbb{R}^m$ is found as

$$x \triangleq f(q) \tag{1}$$

where $f(q) \in \mathbb{R}^m$ represents the forward kinematics of the extensible continuum manipulator, and $q(t) \in \mathbb{R}^n$ represents the manipulator section lengths and curvatures. For the OctArm model used here, it should be noted that $q(t) \in \mathbb{R}^6$ and $q(t) = [d_1, d_2, d_3, \kappa_1, \kappa_2, \kappa_3]^T$, the extension lengths and curvatures for each of the three sections respectively. From (1), the differential relationships between the end-effector position and the section lengths and curvature variables can be calculated as

$$\begin{aligned} \dot{x} &= J(q)\dot{q} \\ \ddot{x} &= \dot{J}(q)\dot{q} + J(q)\ddot{q} \end{aligned} \tag{2}$$

where $\dot{q}(t)$, $\ddot{q}(t) \in \mathbb{R}^n$ represent the velocity and acceleration vectors of the section lengths and curvatures respectively, while $J(t) \in \mathbb{R}^{m \times n}$ denotes the manipulator Jacobian, which can be defined as

$$J(q) \triangleq \frac{\partial f(q)}{\partial q}. \tag{3}$$

C. Dynamic Model

In this article, the dynamic model for the 3-section Octarm, developed by Tatlicioglu *et al.* [19] is utilized . The dynamic model in presented in the familiar Euler-Lagrangian form, and is given by

$$\tau = M(q)\ddot{q} + V(q,\dot{q})\,\dot{q} + Gq + B(q) + E(q) \tag{4}$$

where $\tau \in \mathbb{R}^n$ represents the manipulator control input, $M(q) \in \mathbb{R}^{n \times n}$ represents the inertia matrix, $V(q,\dot{q}) \in \mathbb{R}^{n \times n}$ represents the effect of Centripetal-Coriolis forces, $G(q), B(q)$, and $E(q) \in \mathbb{R}^n$ represent gravitation effects, potential energy due to bending, and potential energy due to extension respectively. The model has the following properties:

Property 1: The inertia matrix $M(q)$ is symmetric and positive definite, and

$$m_1\|\xi\|^2 \leq \xi^T M(q)\xi \leq m_2\|\xi\|^2 \;\forall \xi \in \mathbb{R}^n \tag{5}$$

Property 2: The inertia and Centripetal-Coriolis matrices satisfy the following property

$$\xi^T\left(\dot{M} - 2V\right)\xi = 0 \;\forall \xi \in \mathbb{R}^n. \tag{6}$$

Note that this property is satisfied when $\dot{M} - 2V$ is skew-symmetric.

Remark 1: *The details of the Euler-Lagrangian form for the Octarm can be found in [19].*

III. Error System Formulation

Let $e(t) \in \mathbb{R}^m$ be defined as the task-space tracking error such that

$$e \triangleq x_d - x, \tag{7}$$

where $x_d(t) \in \mathbb{R}^m$ denotes the desired task-space trajectory, and the first and second time derivatives of $e(t)$ are given by $\dot{e}$, $\ddot{e} \in \mathbb{R}^m$ such that

$$\begin{aligned} \dot{e} &= \dot{x}_d - \dot{x} \\ \ddot{e} &= \ddot{x}_d - \ddot{x}. \end{aligned} \tag{8}$$

In (8), $\dot{x}_d(t)$ and $\ddot{x}_d(t) \in \mathbb{R}^m$ represents the desired task-space velocity and acceleration respectively. Additionally, $x_d(t)$ is chosen such that $x_d(t)$, $\dot{x}_d(t)$, and $\ddot{x}_d(t) \in \mathcal{L}_\infty$.

[1]These assumptions are satisfied in our lab for the types of continuum robot examined in the subsequent analysis

Let a sub-task error, denoted by $e_N(t) \in \mathbb{R}^n$ be defined such that

$$e_N \triangleq \left(I_n - J^+ J\right)(g - \dot{q}) \tag{9}$$

where $I_n \in \mathbb{R}^{n\times n}$ denotes the $n\times n$ identity matrix, $J^+(q) \in \mathbb{R}^{n\times m}$ is the pseudoinverse[2] of $J(q)$, and $g(t)$ is an auxiliary signal based on the required sub-task control objective such as, joint limit avoidance, singularity avoidance, or obstacle avoidance.

Remark 2: *A similar sub-task signal to that in the subsequent analysis was introduced in the original work on task-space control for redundant manipulators [27], and subsequently in [28]. In those works however, the connection of the auxiliary signal $g(t) \in \mathbb{R}^n$ to physically meaningful self-motions was tenuous. For continuum robot grasping however, physically meaningful signals $g(t)$ may readily be synthesized. For example, given $x(d,t)$ as the location of contact between the robot and environment, the two $g(t)$ vectors represent self-motion of the robot, maintaining the contact point, but "sliding past" and "rotating around" it, respectively.*

IV. Control Design

Based on the above error system development, and the subsequent stability analysis, the control input $\tau(t) \in \mathbb{R}^m$ is designed as follows

$$\tau = M\left(J^+\left(\ddot{x}_d + K_v\dot{e} + K_p e - \dot{J}\dot{q}\right) + \phi_N\right) + V + B + E \tag{10}$$

where K_p and $K_v \in \mathbb{R}^{m\times m}$ are constant proportional and derivative feedback gain matrices respectively. The vector in the null-space is given by $\phi_N \in \mathbb{R}^{m\times n}$ and can be denoted as

$$\begin{aligned}\phi_N = &\left(I - J^+ J\right)\left(\dot{g} + K_N \dot{e}_N\right)\\ &- \left(J^+\dot{J}J^+ + \dot{J}^+\right) J\left(g - \dot{q}\right)\end{aligned} \tag{11}$$

where $K_N^{n\times n} \in \mathbb{R}^+$ is a positive definite feedback matrix.

Theorem 1: The controller proposed in (10) guarantees that $e(t) \to 0$ as $t \to \infty$ provided the manipulator does not go through a singularity.

Proof: From, (10), the closed loop system can be given as

$$\begin{aligned}&M\ddot{q} + V\dot{q} + B + E =\\ &M\left\{J^+\left(\ddot{x}_d + K_v\dot{e} + K_p e - \dot{J}\dot{q}\right) + \phi_N\right\} + V + B + E.\end{aligned} \tag{12}$$

From the expression in (12), it can be seen that

$$\ddot{q} = J^+\left(\ddot{x}_d + K_v\dot{e} + K_p e - \dot{J}\dot{q}\right) + \phi_N, \tag{13}$$

since $M(\dot{q})$ satisfies Property 1. From (2), it can be seen that

$$\ddot{q} = J^+\left(\ddot{x} - \dot{J}\dot{q}\right) + \ddot{q}_N, \tag{14}$$

where $\ddot{q}_N(t) \in \mathbb{R}^m$ is a vector in the null space of $J(q)$. Substituting (14) in (13) results in the following expression

$$J^+\left(\ddot{e} + K_v\dot{e} + K_p e\right) = \ddot{q}_N - \phi_N. \tag{15}$$

[2]Properties of the Moore-Penrose pseudoinverse are provided in Appendix I

Premultiplying (15) by the Jacobian $J(q)$ results in

$$\ddot{e} + K_v\dot{e} + K_p e = 0, \tag{16}$$

as from the properties of pseudoinverses listed in Appendix I, $JJ^+ = I$, when $J(q)$ is of full rank, and $J\left(\ddot{q}_N - \phi_N\right) = 0$ because $\ddot{q}_N - \phi_N$ belongs in the null space of $J(q)$. The choices of K_p and K_v that ensure the frequency-domain transform $s^2 + K_v s + K_p$, where is s is the Laplace transform operator, is Hurwitz, results in the error signal $e(t) \to 0$, exponentially. ∎

In continuation of the controller synthesis, the time derivative of (9) yields

$$\dot{e}_N = \left(I - J^+ J\right)\left(\dot{g} - \ddot{q}\right) - \left(\dot{J}^+ J + \dot{J}\dot{J}\right)\left(g - \dot{q}\right), \tag{17}$$

in which, the substitution of (9) results in

$$\begin{aligned}\dot{e}_N = &\left(I - J^+ J\right)\left(\dot{g} - \ddot{q}\right) - J^+\dot{J}e_N\\ &- \left(J^+\dot{J}J^+ + \dot{J}^+\right) J\left(g - \dot{q}\right).\end{aligned} \tag{18}$$

Substituting equations (11) and (14) in (18) results in

$$\begin{aligned}\dot{e}_N = &\left(I - J^+ J\right)\dot{g} - \phi_N - J^+\dot{J}e_N\\ &- \left(J^+\dot{J}J^+ + \dot{J}^+\right) J\left(g - \dot{q}\right),\end{aligned} \tag{19}$$

which, given that $(I - J^+J)J^+ = 0$ and $\phi_N(t)$ is in the null-space of $J(q)$, can further be rewritten as

$$\dot{e}_N = -\left(I - J^+ J\right) K_N e_N - J^+\dot{J}e_N. \tag{20}$$

Theorem 2: The control law given in (10) and the null-space vector given in (11) guarantees global asymptotic sub-task tracking such that the configuration velocity goes to $g(t)$ in the null space, i.e. $\|e_N\| \to 0$ as $t \to \infty$ provided the manipulator does not achieve a singular configuration.

Proof: Let a Lyapunov function $V(t) \in \mathbb{R}^n$ be defined such that

$$V = \frac{1}{2}\|e_N\|^2. \tag{21}$$

The time derivative of (21) yields

$$\dot{V} = e_N^T\dot{e}_N, \tag{22}$$

in which the substitution of (20) results in the following expression

$$\dot{V} = -e_N^T\left(I - J^+ J\right) K_N e_N - e_N^T J^+\dot{J}e_N, \tag{23}$$

which can further be reduced to

$$\dot{V} = -e_N^T K_N e_N, \tag{24}$$

by utilizing the properties of pseudoinverses listed in Appendix I.

It can be seen that $V(t)$ in equation (21) is positive-definite, and $\dot{V}(t)$ in equation (24) is subsequently negative-definite, it goes that $\|e_N\| \to 0$ monotonically as $t \to \infty$. ∎

V. Simulation Example

A numerical simulation was performed using the MATLAB/Simulink package to outline the performance of the controller proposed in equation (10). The simulation consisted of two phases; The first phase highlighted the performance of the proposed controller, was set up to have $x(d,t)$ correspond to the tip of the robot using the mathematical model of the OctArm manipulator [7]. The second phase assumed contact of the manipulator's tip-section mid-point with an object, which it then attempted to grasp. For the purpose of this paper, all three sections were actuated, and for simplicity, the robot motion was restricted to the plane orthogonal to the effect of gravity. Thus, the robot has six degrees of freedom, that of extension/contraction and bending for each of the three links.

In the simulation the robot is modeled using the Jacobian derived from [14] and dynamic models detailed in [19] and [29] for the Octarm continuum manipulator as shown in Figure 2. The controller in the simulation has an additive white Gaussian noise with $SNR = 60$ introduced into the configuration-space information fed back into the system to simulate real-time errors due to modeling inaccuracies, sensor errors or other unaccounted-for dependencies.

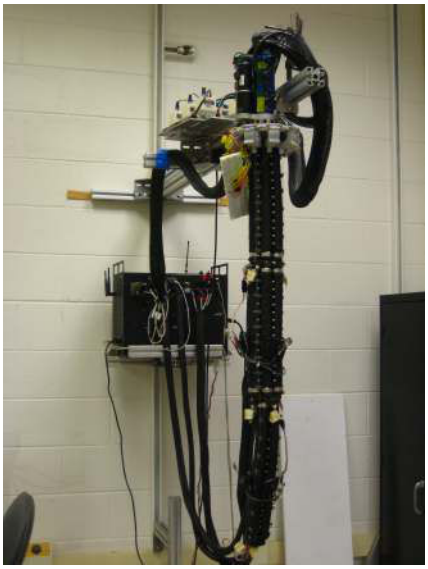

Fig. 2. The Octarm Continuum Manipulator

The nominal desired trajectories for the end-effector position are represented by an aggressive set of damped sinusoids given by

$$x_d = \begin{bmatrix} 0.7494 + 0.01sin\left(\frac{\pi t}{6}\right) \\ 0.124 + 0.05sin\left(\frac{\pi t}{6}\right) \\ 0 \end{bmatrix}. \qquad (25)$$

where the $x_d(t)$ vector represents the x-coordinate and z-coordinate trajectories, and y-coordinate orientation respectively.

The control gains used that provided these results were

$$K_p = 150I_3 \; K_v = 0 * I_3 \; K_N = I_6. \qquad (26)$$

It can be seen that the error plots shown in Figure 3 converge satisfactorily within 5 seconds when the system is controlling the tip of the manipulator. This is consistent with the dynamic characteristics of the physical Octarm, which is a pneumatically-actuated compliant device featuring a more sluggish dynamic response and higher time constants than conventional industrial manipulators [9].

At $t = 20$ $seconds$, we assume that the mid-point of the tip section of the manipulator comes into contact with an object, which is when the controller switches from regulation mode to grasping mode, with the general structure of the controller and its gains remaining unchanged. This means that after contact has been made, the "free" portion of the tip section continues to bend, attempting to "grasp" the object as seen in Figure 7. Figure 8 illustrates the section lengths of the manipulator which remain constant upon the switch to the Grasping Phase. When coupled with the changing section curvatures, this explains the change in manipulator tip positions in Figure 5 which is no longer being controlled. Figure 4 illustrates that despite the switch in tasks, the manipulator is able to track its new desired position. The manipulator control inputs for this mode are shown in Figure 6.

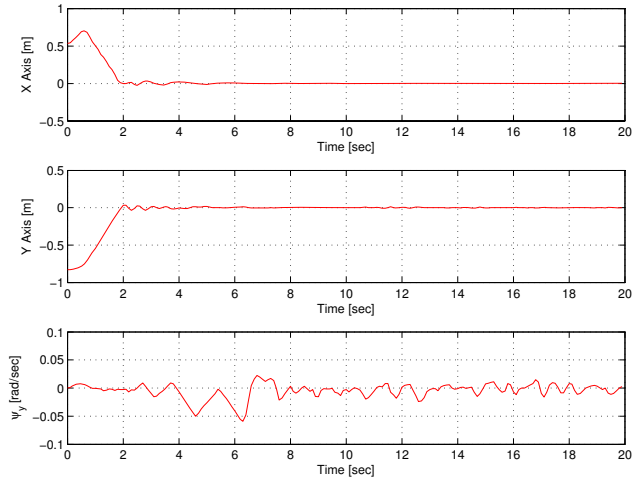

Fig. 3. Octarm Tip Control Phase Tracking Errors

VI. Conclusion

We have introduced a new model-based nonlinear task-space controller extending the concept originally developed for rigid-link manipulators to a general class of continuum robots. The application adopts the kinematics, velocity Jacobian, and dynamics recently established in the literature. By exploiting the structure inherent in the mathematical model, the controller is guaranteed to converge in spite of inherent errors due to uncertainty in the model parameters. This result is applicable to all continuum robots that extend as well as bend, and is demonstrated via two simulations using the model of a three-section extensible continuum manipulator.

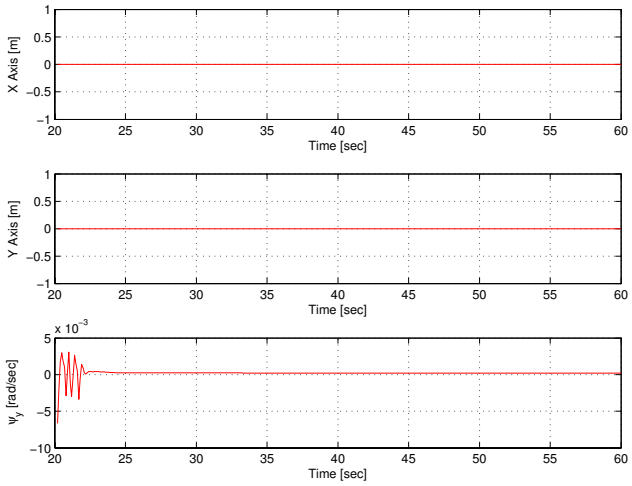

Fig. 4. Octarm Grasping Phase Tracking Errors

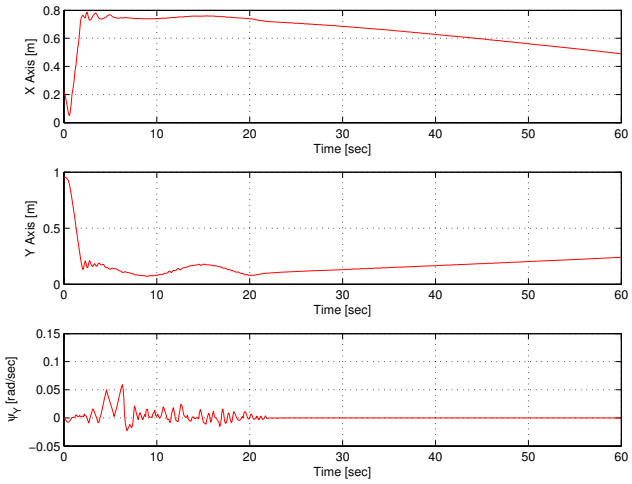

Fig. 5. Octarm Tip Section Positions

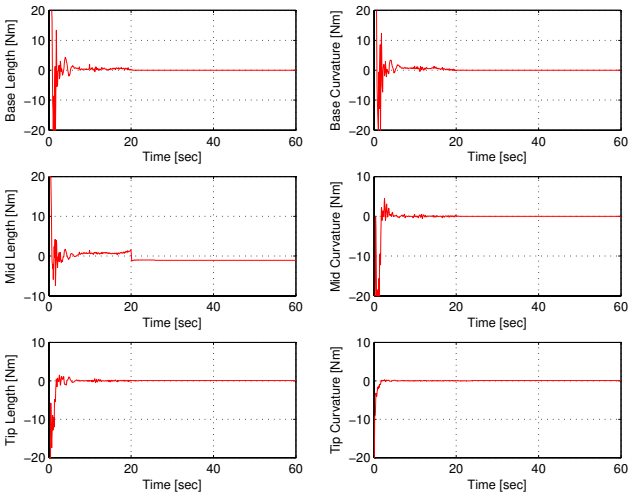

Fig. 6. Octarm Section Length and Curvature Control Inputs

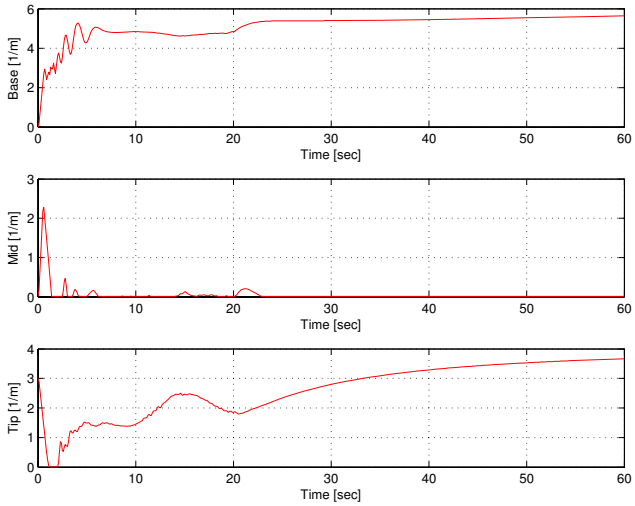

Fig. 7. OctarmSection Curvatures

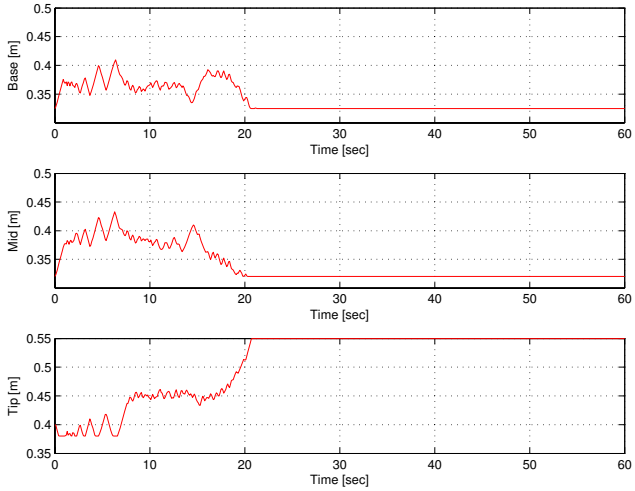

Fig. 8. Octarm Section Lengths

APPENDIX I
PSEUDOINVERSE PROPERTIES

For the development of the task-space controller, the pseudoinverse, $J^+(q)$, of the Jacobian $J(q)$ is defined as

$$J^+ \triangleq J^T \left(JJ^T\right)^{-1}, \tag{27}$$

resulting in the property

$$JJ^+ = I_m \tag{28}$$

where $I_m \in \mathbb{R}^{m\times m}$ is the standard identity matrix. As described in [30], the pseudoinverse defined in (27) satisfies the following Moore-Penrose properties

$$\begin{aligned} JJ^+J = J \quad J^+JJ^+ = J^+ \\ \left(J^+J\right)^T = J^+J \quad \left(JJ^+\right)^T = JJ^+. \end{aligned} \tag{29}$$

In addition to these properties listed above, the matrix

$(I_n - J^+J)$ also satisfies the following useful properties

$$\begin{aligned}(I_n - J^+J)(I_n - J^+J) &= I_n - J^+J \\ (I_n - J^+J)^T &= I_n - J^+J \\ J(I_n - J^+J) &= 0_{n\times m} \\ (I_n - J^+J)J^+ &= 0_{n\times m}.\end{aligned} \tag{30}$$

REFERENCES

[1] G. Robinson and J. Davies, "Continuum robots - a state of the art," in *Proc. IEEE Int. Conf. Robot. Autom.*, Detroit, MI, 1999, pp. 2849–2854.

[2] R. Buckingham and A. Graham, "Snaking around a nuclear jungle," *Ind. Robot: An Int. Journal*, vol. 32, no. 2, pp. 120–127, Feb. 2005.

[3] G. Immega and K. Antonelli, "The ksi tentacle manipulator," in *Proc. IEEE Int. Conf. Robot. Autom.*, Nagoya, Japan, 1995, pp. 3149–3154.

[4] D. Lane, B. Davies, G. Robinson, D. O'Brien, J. Sneddon, E. Seaton, and A. Elfstrom, "Practical kinematics for real-time implementation of continuum robots," *IEEE Jour. Ocean. Eng.*, vol. 24, no. 1, pp. 96–111, Jan. 1999.

[5] H. Ohno and S. Hirose, "Design of a slim slime robot and its gait of locomotion," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Maui, HI, 2001, pp. 707–715.

[6] S. Hirose, *Biologically Inspired Robots*. Oxford, UK: Oxford University Press, 1993.

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

[8] K. Suzimori, S. Iikura, and H. Tanaka, "Development of a flexible microactuator and its applications to robotic mechanisms," in *Proc. IEEE Int. Conf. Robot. Autom.*, Sacramento, CA, 1991, pp. 1622–1627.

[9] D. Braganza, D. Dawson, I. Walker, and N. Nath, "A neural network controller for continuum robots," *IEEE Trans. Robot.*, vol. 23, no. 6, pp. 1270–1277, Dec. 2006.

[10] M. Grissom, V. Chitrakaran, D. Dienno, M. Csencsits, M. Pritts, B. Jones, W. McMahan, D. Dawson, C. Rahn, and I. Walker, "Design and experimental testing of the octarm soft robot manipulator," in *Proc. SPIE Conf. Unmanned Sys. Tech.*, Kissimee, FL, 2006, pp. 109–114.

[11] N. Giri and I. Walker, "Continuum robots and underactuated grasping," in *Proc. ASME Int. Work. Underactuated Grasping*, Montreal, Canada, 2010, pp. 1–8.

[12] G. Chirikjian and J. Burdick, "A modal approach to hyper-redundant manipulator kinematics," *IEEE Trans. Robot. Autom.*, vol. 10, no. 3, pp. 343–354, Jun. 1994.

[13] I. Gravagne and I. Walker, "Kinematic transformations for remotely-actuated planar continuum robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, San Francisco, CA, 2000, pp. 19–26.

[14] B. Jones and I. Walker, "Kinematics of multisection continuum robots," *IEEE Trans. Robot.*, vol. 22, no. 1, pp. 43–57, Feb. 2006.

[15] G. Chirikjian, "Hyper-redundant manipulator dynamics: A continuum approximation," *Adv. Robot.*, vol. 9, no. 3, pp. 217–243, Jun. 1995.

[16] G. Gallot and O. I. W. Khalil, "Dynamic modeling and simulation of a 3-d eel-like robot," in *Proc. IEEE Int. Conf. Robot. Autom.*, Rome, Italy, 2007, pp. 1486–1491.

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

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

[19] E. Tatlicioglu, I. Walker, and D. Dawson, "Dynamic modeling for planar extensible continuum robot manipulators," *Int. Jour. Robot. Autom.*, vol. 24, no. 4, pp. 1087–1099, Apr. 2009.

[20] W. McMahan, B. Jones, and I. Walker, "Robotic manipulators inspired by cephalopod limbs," *Jour. Eng. Des. Inno.*, vol. 1, no. P, p. 01P2, Jun. 2005.

[21] M. Ivanescu, N. Popescu, and D. Popescu, "A variable length tentacle manipulator control system," in *Proc. IEEE Int. Conf. Robot. Autom.*, Barcelona, Spain, 2005, pp. 3274–3279.

[22] M. Ivanescu and V. Stoian, "A variable structure controller for a tentacle manipulator," in *Proc. IEEE Int. Conf. Robot. Autom.*, Nagoya, Japan, 1995, pp. 3155–3160.

[23] ——, "A controller for hyper-redundant cooperative robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Vancouver, Canada, 1998, pp. 167–172.

[24] A. Kapadia, I. Walker, D. Dawson, and E. Tatlicioglu, "A new approach to extensible continuum robot control using the sliding-mode," *Intl. J. Comp. Tech. and App.*, vol. 2, no. 4, pp. 293–300, 2011.

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

[26] G. Chen, M. Pham, and T. Redarce, "Development and kinematic analysis of a silicon rubber bending tip for colonoscopy," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Beijing, China, 2006, pp. 168–173.

[27] P. Hsu, J. Hauser, and S. Sastry, "Dynamic control of redundant manipulators," *J. Robot. Sys.*, vol. 6, no. 2, pp. 133–148, 1989.

[28] E. Zergoroglu, D. Dawson, I. Walker, and A. Behal, "Nonlinear tracking control of kinematically redundant robot manipulators," in *Proc. American Control Conf.*, Chicago, IL, 2000, pp. 2513–2517.

[29] E. Tatlicioglu, I. Walker, and D. Dawson, "New dynamic models for planar extensible continuum robot manipulators," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, San Diego, CA, 2007, pp. 1485–1490.

[30] Y. Nakamura, *Advanced Robot Redundancy and Optimization*. Reading, MA: Addison-Wesley, 1991.