Robotica (2007) volume 25, pp. 581–586. © 2007 Cambridge University Press
doi:10.1017/S0263574707003475 Printed in the United Kingdom

Setpoint regulation of continuum robots using a fixed camera

V. K. Chitrakaran*†, A. Behal‡, D. M. Dawson† and I. D. Walker†

†*Electrical and Computer Engineering, Clemson University, Clemson, SC 29634-0915, USA.*
‡*SEECS and NanoScience Technical Center, University of Central Florida, Orlando, FL 32826, USA.*

(Received in Final Form: February 12, 2007. First published online: March 23, 2007)

SUMMARY

In this paper, we investigate the problem of measuring the shape of a continuum robot manipulator using visual information from a fixed camera. Specifically, we capture the motion of a set of fictitious planes, each formed by four or more feature points, defined at various strategic locations along the body of the robot. Then, utilizing expressions for the robot forward kinematics as well as the decomposition of a homography relating a reference image of the robot to the actual robot image, we obtain the three-dimensional shape information continuously. We then use this information to demonstrate the development of a kinematic controller to regulate the manipulator end-effector to a constant desired position and orientation.

KEYWORDS: Continuum robots; Computer vision; Shape estimation.

1. Introduction

Conventional robotic manipulators are designed as a kinematic chain of rigid links that bend at discrete joints to achieve a desired motion at its end-effector. These rigid-link robots have a limited number of joints, and all the joints are actuated by devices such as motors. The maneuverability and flexibility of such devices are limited by the number of actuated joints in them. In contrast, continuum robots[1] are robotic manipulators that draw inspiration from biological appendages like elephant trunks and squid tentacles, and can bend anywhere along the length of their body. In theory, they have infinite mechanical degrees of freedom (DOF), so that their end-effector can be positioned at a desired location while concurrently satisfying work-space constraints such as tight spaces and presence of obstacles. However, from an engineering perspective, an important implication of such a design is that although such devices have a high kinematic redundancy, they are infinitely underactuated. A variety of bending motions must be generated with only a finite number of actuators. While there has been considerable progress in the area of actuation strategies for such robots,[1] the dual problem of sensing the configuration of such robots has been a challenge. From a controls perspective, a reliable position controller would require an accurate position sensing mechanism. However, internal motion sensing devices such as encoders cannot be used to determine either the shape or the end-effector position of a continuum robot, since there is no intuitive way to define links and joints on such a device. A literature survey reveals that a few indirect methods have been proposed by researchers to estimate the shape of continuum robots, such as models[2,3] that relate internal bellow pressures in fluid-filled devices, or change in tendon-length in tendon driven devices, to the position of the end-effector. However, these methods do not have accuracies comparable to position sensing in rigid link robots because of the compliant nature of continuum devices. For example, in a tendon-driven continuum robot, due to coupling of actuation between sections, various sections of the robot can potentially change shape without the encoders detecting a change in tendon length or tension. Motivated by a desire to develop an accurate strategy for realtime shape sensing in such robots, Hannan and Walker[4] implemented simple image processing techniques to determine the shape of the Elephant Trunk robotic arm at Clemson University, where images from a fixed camera were used to reconstruct the curvatures of various sections of the robot. This technique was only applicable to the case where the motion of the arm was restricted to a plane orthogonal to the optical axis of the camera. The paper, however, demonstrated conclusively that there is a large difference in curvature measurements obtained from indirect cable measurements as compared to vision-based strategy, and hence, the information obtained from *ad hoc* indirect shape measurement techniques is indeed questionable.

Vision-based techniques for shape sensing are appealing, if they can be used to reconstruct the 3D pose of the robot without applying any conditions that constrain the maneuverability of the robot. It can be shown that the correspondence between images of feature points lying on a plane, as obtained from two different cameras, is a collineation,[5] and given the matrix of collineation, the position and orientation of the second camera and the plane can be recovered relative to the first camera. In lieu of images from a second camera, given a single reference image of the plane and a knowledge of its (reference) orientation relative to the coordinate frame of a single camera, new images of the moving plane captured by the camera can be compared with this reference image to determine the changing position and orientation of the plane relative to the camera. Exploiting this technique, Chen *et al.*[6] presented the development of a kinematic controller for robot manipulators using visual feedback from a single fixed camera. In this paper, we follow a similar approach with regard to modelling the motion of various sections of a continuum robot relative to a fixed camera. Specifically, from

* Corresponding author. Email: c.vilas@gmail.com

a decomposition of the homography and from the equations describing the forward kinematics of the robot,[7] we show that the curvatures that define the shape of various sections of the robot can be fully determined. This is the primary contribution of this paper. We then make use of the various kinematic control strategies for hyperredundant robots that have appeared in the robotics literature in the past[8–10] to develop a kinematic controller that accurately positions the robot end-effector to any desired position and orientation by using a sequence of images from a single external video camera.

This paper is organized as follows. Section 2 introduces the forward kinematics for the robot. In Section 3, we illustrate how a homography-based approach can be used to continuously reconstruct the three-dimensional pose of the continuum robot, utilizing the two-dimensional images from a fixed camera. In Section 4, we use this information to develop a kinematic controller that regulates the end-effector of the robot to any desired constant reference position and orientation. Section 5 provides simulation results. Concluding remarks are presented in Section 6.

2. Continuum robot kinematics

The kinematics of a conventional, rigid-link, industrial robot can be conveniently described as a function of joint angles and link lengths using the standard Denavit–Hartenberg convention.[15] This is a systematic method of assigning orthogonal coordinate frames to the joints of the robot, such that the relative position and orientation between frames along the kinematic chain can be obtained as a product of homogeneous transformation matrices. In comparison, continuum robots resemble snakes or tentacles in their physical structure, and due to their continuous and curving shape, there is no intuitive way to define links and joints on them. The concept of curvature, pioneered by Chirikjian and Burdick[11] and appearing in the work of other researchers in the area of continuum robotics,[12,13] is a natural way to describe the kinematics of a continuum robot. One such continuum robot is the Clemson Elephant Trunk,[14] which is composed of 16 two-DOF joints divided into four sections, each section designed to bend with a constant planar curvature. Every section is cable driven, and can be actuated such that it defines a different orientation of the plane of its curvature relative to its preceding section. Due to the rigid nature of the joints, torsion is not possible within a section.

Based on the work by Hannan,[14] and further refined by Jones,[7] the fundamental idea behind development of kinematics for an individual section of this robot is to fit a virtual conventional rigid-link manipulator to its continuous curvature, and develop relationships utilizing the well-established Denavit–Hartenber procedure. Consider the sth section of the robot. Using basic geometry, the kinematics of a 2D planar curve of arc length l_s and curvature k_s can be described by three coupled movements—rotation by an angle θ_s, followed by a translation x_s, and a further rotation by angle θ_s, as shown in Fig. 1. Here, $x_s \in \mathbb{R}^3$ is the position vector of the endpoint of the curve relative to its initial point, and

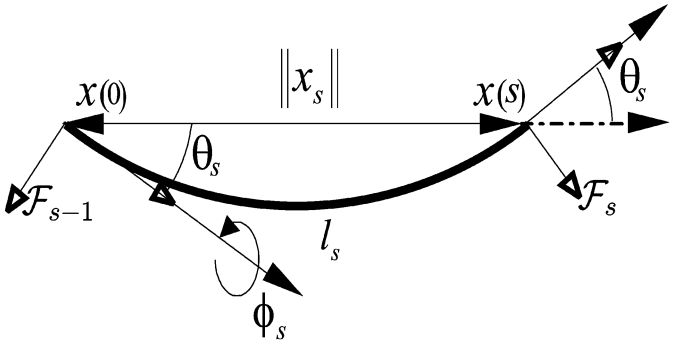

Fig. 1. A planar curve.

$$\theta_s = \frac{k_s l_s}{2} \tag{1}$$

$$\|x_s\| = \frac{l_s}{\theta_s}\sin(\theta_s). \tag{2}$$

After treating the two rotations in the curve as discrete rotational joints and the translation as a coupled discrete prismatic joint, the standard Denavit–Hartenberg procedure[15] can be applied to obtain the forward kinematics for the curve. Thus, the homogeneous transformation matrix for the planar curve, denoted by $A_{sp} \in \mathbb{R}^{4\times4}$, can be obtained as[14]

$$A_{sp} = \begin{bmatrix} \cos(k_s l_s) & -\sin(k_s l_s) & 0 & \frac{1}{k_s}\{\cos(k_s l_s) - 1\} \\ \sin(k_s l_s) & \cos(k_s l_s) & 0 & \frac{1}{k_s}\sin(k_s l_s) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3}$$

The out-of-plane rotation of the section relative to the preceding plane can be modelled as an additional rotational joint with rotation of angle ϕ_s about the initial tangent of the curve (see Fig. 1). However, a rotation about the initial tangent results in incorrect orientation of the frame defined at the other end of the curve, since the body of the robot cannot experience torsion. Therefore, in order to cancel out this torsion, the frame defined at the distal end of the curve is finally rotated by $-\phi_s$. Hence, for the 3D case, the forward kinematics for the sth section of the continuum robot can be obtained from the following homogeneous transformation matrix

$$A_s = \begin{bmatrix} R^s_{s-1} & t^s_{s-1} \\ 0 & 1 \end{bmatrix} \tag{4}$$

where

$$R^s_{s-1} = \begin{bmatrix} 1+\cos^2(\phi_s)\mathrm{ck}_s & \sin(\phi_s)\cos(\phi_s)\mathrm{ck}_s & -\cos(\phi_s)\sin(k_s l_s) \\ \sin(\phi_s)\cos(\phi_s)\mathrm{ck}_s & \cos(k_s l_s)\text{-}\cos^2(\phi_s)\mathrm{ck}_s & -\sin(\phi_s)\sin(k_s l_s) \\ \sin(k_s l_s)\cos(\phi_s) & \sin(k_s l_s)\sin(\phi_s) & \cos(k_s l_s) \end{bmatrix} \tag{5}$$

https://doi.org/10.1017/S0263574707003475 Published online by Cambridge University Press

$$t_{s-1}^{s} = \begin{bmatrix} \frac{1}{k_s}\cos(\phi_s)\mathrm{ck}_s \\ \frac{1}{k_s}\sin(\phi_s)\mathrm{ck}_s \\ \frac{1}{k_s}\sin(k_s l_s) \end{bmatrix} \tag{6}$$

and $ck_s(t) = \cos(k_s l_s) - 1$. The matrix A_s in (4) transforms the coordinates of a point defined in the coordinate frame $\mathcal{F}_s$ at the end of the sth curved section to the coordinate frame $\mathcal{F}_{s-1}$ defined at the end of the $(s-1)$th section. In the above equations, $R_{s-1}^{s} \in SO(3)$ and $t_{s-1}^{s} \in \mathbb{R}^3$ define, respectively, the rotation matrix and translation vector between the frames $\mathcal{F}_s$ and $\mathcal{F}_{s-1}$. Thus, for the entire robot with four sections, the homogeneous transformation matrix can be calculated as

$$T_0^4 = A_1 A_2 A_3 A_4. \tag{7}$$

From (7), the end-effector position and orientation in the task-space of the robot, denoted by $p(t) \in \mathbb{R}^6$, can be written as

$$p = f(q) \tag{8}$$

where $f(q) \in \mathbb{R}^6$ denotes the forward kinematics, and $q(t) \in \mathbb{R}^8$ denotes the joint-space variables for the robot defined as

$$q(t) = [\phi_1\ k_1\ \phi_2\ k_2\ \phi_3\ k_3\ \phi_4\ k_4]^{\mathrm{T}} \tag{9}$$

where $\phi_i(t)$ and $k_i(t)$ are the out-of-plane rotation and the curvature, respectively, for the ith section.

Based on (8), a differential relationship between the end-effector position and the joint-space variables can be defined as[7]

$$\dot{p} = J(q)\dot{q} \tag{10}$$

where $J(q) \triangleq \dfrac{\partial f(q)}{\partial q} \in \mathbb{R}^{6\times 8}$ is called a Jacobian matrix, and $\dot{q}(t) \in \mathbb{R}^8$ denotes the joint-space velocity vector. Note here that the determination of the Jacobian matrix requires knowledge of the joint-space vector $q(t)$. In Section 3, we describe how $q(t)$ can be constructed from images of feature points along the manipulator as obtained from the fixed camera.

3. Joint variables extraction

3.1. Camera space coordinates of feature points

Since a video camera is our position feedback device, we must develop a geometric relationship between the 3D world in which the robot resides and its 2D projection in the image plane of the camera. To this end, we define an inertial coordinate system $\mathcal{I}$, whose origin coincides with the center of a fixed camera (see Fig. 2). For the sake of simplicity, we assume that the origin of the inertial frame $\mathcal{I}$ also coincides with the origin of the robot base frame. At the end of sth section of the robot, consider a transverse plane π_s defined by four noncollinear target points denoted by $O_{si}\ \forall i = 1, 2, 3, 4$ such that the origin of the previously defined coordinate system $\mathcal{F}_s$ lies in π_s. We also consider a fixed transverse plane

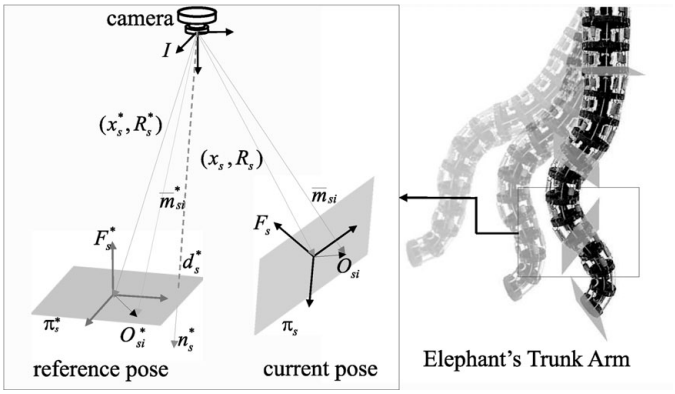

Fig. 2. Coordinate frame relationships.

denoted by π_s^*, (with four noncollinear target points denoted by $O_{si}^*\ \forall i = 1, 2, 3, 4$) and a coordinate system $\mathcal{F}_s^*$, which are defined when the end of the sth section is at a reference position and orientation relative to the fixed camera (i.e., π_s^* and $\mathcal{F}_s^*$ are defined by a reference image of the robot). Note that this reference image doesnot necessarily represent the desired position to which we want to regulate the end-effector of the robot. The 3D coordinates of the target points O_{si}, O_{si}^*, denoted by $\bar{m}_{si}(t)$, $\bar{m}_{si}^* \in \mathbb{R}^3$ in π_s and π_s^*, respectively, are expressed in the inertial coordinate system $\mathcal{I}$ as

$$\bar{m}_{si} \triangleq [\,x_{si}\ y_{si}\ z_{si}\,]^{\mathrm{T}} \tag{11}$$

$$\bar{m}_{si}^* \triangleq [\,x_{si}^*\ y_{si}^*\ z_{si}^*\,]^{\mathrm{T}}. \tag{12}$$

We define the normalized Euclidean coordinates, denoted by $m_{si}(t)$ and $m_{si}^* \in \mathbb{R}^3$ for the above target points as

$$m_{si} \triangleq \frac{\bar{m}_{si}}{z_{si}} = \left[\frac{x_{si}}{z_{si}}\ \frac{y_{si}}{z_{si}}\ 1\right]^{\mathrm{T}} \tag{13}$$

$$m_{si}^* \triangleq \frac{\bar{m}_{si}^*}{z_{si}^*} = \left[\frac{x_{si}^*}{z_{si}^*}\ \frac{y_{si}^*}{z_{si}^*}\ 1\right]^{\mathrm{T}}. \tag{14}$$

Seen through the camera, each of the points O_{si} and O_{si}^* in task space will also have projected pixel coordinates, expressed in terms of $\mathcal{I}$ denoted by $u_{si}(t)$, $v_{si}(t) \in \mathbb{R}$, and u_{si}^*, $v_{si}^* \in \mathbb{R}$, that are respectively defined as elements of $p_{si}(t)$ and $p_{si}^* \in \mathbb{R}^3$ as

$$p_{si} = [u_{si}\ v_{si}\ 1]^{\mathrm{T}} \qquad p_{si}^* = [u_{si}^*\ v_{si}^*\ 1]^{\mathrm{T}}. \tag{15}$$

The projected pixel coordinates of the target points are related to their normalized Euclidean coordinates by the pin-hole camera model,[16] such that

$$p_{si} = Am_{si} \qquad p_{si}^* = Am_{si}^* \tag{16}$$

where $A \in \mathbb{R}^{3\times 3}$ is a known, constant, and invertible intrinsic camera calibration matrix that is explicitly defined as[17]

$$A = \begin{bmatrix} fk_u & -fk_v\cot(\theta) & u_o \\ 0 & \dfrac{fk_v}{\sin(\theta)} & v_o \\ 0 & 0 & 1 \end{bmatrix} \tag{17}$$

https://doi.org/10.1017/S0263574707003475 Published online by Cambridge University Press

where u_o, $v_o \in \mathbb{R}$ denote the pixel coordinates of the principal point (i.e., the image center that is defined as the frame-buffer coordinates of the intersection of the optical axis with the image plane), $k_u, k_v \in \mathbb{R}$ represent camera scaling factors, $\theta \in \mathbb{R}$ is the angle between the axes of the imaging elements (CCD) in the camera, and $f \in \mathbb{R}$ denote the focal length of the camera.

3.2. Euclidean reconstruction

In order to develop a relationship between the coordinate system $\mathcal{I}$ defined at the center of the fixed camera and the coordinate system $\mathcal{F}_s$ defined at the end of the sth section of the robot, we define $R_s(t) \in SO(3)$ as the rotation matrix between $\mathcal{F}_s$ and $\mathcal{I}$, and $x_s(t) \in \mathbb{R}^3$ as the translation vector between $\mathcal{F}_s$ and $\mathcal{I}$, $\forall s = 1, 2, 3, 4$. Similarly, let $x_s^* \in \mathbb{R}^3$ be a constant translation vector between $\mathcal{F}_s^*$ and $\mathcal{I}$, and $R_s^* \in SO(3)$ be the constant known rotation matrix between $\mathcal{F}_s^*$ and $\mathcal{I}$. As also illustrated in Fig. 2, $n_s^* \in \mathbb{R}^3$ denotes a constant normal to the reference plane π_s^* expressed in the coordinates of $\mathcal{I}$, and the constant distance $d_s^* \in \mathbb{R}$ from $\mathcal{I}$ to plane π_s^* along the unit normal is given by

$$d_s^* = n_s^{*T} \bar{m}_{si}^*. \tag{18}$$

Note that O_{si} and O_{si}^* represent the same feature point at different geometric locations, and when expressed in the object reference frames $\mathcal{F}_s$ and $\mathcal{F}_s^*$, they have the same coordinates. Exploiting this fact, and based on the geometry between the coordinate frames $\mathcal{F}_s$, $\mathcal{F}_s^*$ and $\mathcal{I}$ depicted in Fig. 2, we can arrive at the following relationships

$$\bar{m}_{si} = x_s + R_s O_{si} \tag{19}$$

$$\bar{m}_{si}^* = x_s^* + R_s^* O_{si}. \tag{20}$$

After solving Eq. (20) for O_{si} and substituting the resulting expression into Eq. (19), the following relationships can be obtained

$$\bar{m}_{si} = \bar{x}_s + \bar{R}_s \bar{m}_{si}^* \tag{21}$$

where $\bar{R}_s(t) \in SO(3)$ and $\bar{x}_s(t) \in \mathbb{R}^3$ are the new rotational and translational variables, respectively, defined as

$$\bar{R}_s = R_s (R_s^*)^{\mathrm{T}} \qquad \bar{x}_s = x_s - \bar{R}_s x_s^*. \tag{22}$$

After utilizing Eq. (18), the relationship in Eq. (21) can now be expressed as

$$\bar{m}_{si} = \left(\bar{R}_s + \frac{\bar{x}_s}{d_s^*} n_s^{*\mathrm{T}} \right) \bar{m}_{si}^* . \tag{23}$$

After utilizing Eqs. (13) and (14), we obtain the following relationship in terms of normalized Euclidean coordinates of the feature points

$$m_{si} = \underbrace{\frac{z_{si}^*}{z_{si}}}_{\alpha_{si}} \underbrace{\left(\bar{R}_s + \frac{\bar{x}_s}{d_s^*} n_s^{*\mathrm{T}} \right)}_{H_s} m_{si}^* \tag{24}$$

where $\alpha_{si}(t) \in \mathbb{R}$ is the depth ratio, and $H_s(t) \in \mathbb{R}^{3\times3}$ denotes the Euclidean homography between the coordinate systems $\mathcal{F}_s$ and $\mathcal{F}_s^*$. Given the relationships in Eq. (16), the above relationship can be written in terms of pixel coordinates of the target points in each plane as

$$p_{si}(t) = \alpha_{si} \underbrace{(AHA^{-1})}_{G_s} p_{si}^* \tag{25}$$

where $G_s(t) \in \mathbb{R}^{3x3}$ is called the projective homography.

If all feature points in a section lie on the same plane, the distances d_s^* defined in Eq. (18) is the same for all feature points in that section. In this case, the collineation $G_s(t)$ is defined upto the same scale factor, and hence, one of its elements can be set to unity without loss of generality. Given the images of at least four coplanar features $p_{si}(t)$ on each plane π_s and the images of the corresponding reference features p_{si}^* in π_s^*, we can solve the linear set of equations[18] in Eq. (25) to determine $G_s(t)$ and $\alpha_{si}(t)$. If more than four feature points can be tracked on each plane, least-squares solution may be used to arrive at an estimate for $G_s(t)$. If the points O_{si} are not coplanar, then the estimation of $G(t)$ is a nonlinear problem that requires at least eight feature points and can be computed, for example, by using the virtual parallax algorithm.[17] Since the camera calibration matrix A in Eq. (17) is known, $H_s(t)$ can be obtained from $G_s(t)$ for each section of the manipulator. By utilizing various techniques,[5,19] $H_s(t)$ can be decomposed into rotational and translational components as in Eq. (24). Specifically, the rotation matrix $\bar{R}_s(t)$ can be computed from the decomposition of $H_s(t)$. The rotation matrix $R_s(t)$, defining the orientation of the end of the sth section of the robot relative to the camera fixed frame $\mathcal{I}$, can then be computed from $\bar{R}_s(t)$ by using Eq. (22) and the fact that R_s^* is known *a priori*.

Since $R_s(t)$ is a rotation matrix between $\mathcal{I}$ and $\mathcal{F}_s$, it can be viewed as a composition of two rotational transformations: a rotational transformation from frame $\mathcal{I}$ to $\mathcal{F}_{s-1}$, followed by a second rotational transformation from $\mathcal{F}_{s-1}$ to $\mathcal{F}_s$. Hence, we can progressively compute $R_{s-1}^s(t)$ in Eq. (5) (i.e., the rotation matrix from one section of the robot to the next), as[15]

$$R_{s-1}^s = (R_{s-1})^{\mathrm{T}} R_s, \qquad \forall s = 1, 2, 3, 4. \tag{26}$$

From Eq. (4), the joint-space variables for the sth section can, hence, be determined as

$$k_s = \frac{1}{l_s} \cos^{-1} \left(\left[R_{s-1}^s \right]_{33} \right)$$

$$\phi_s = \sin^{-1} \left(\frac{\left[R_{s-1}^s \right]_{32}}{\sin(k_s l_s)} \right) \tag{27}$$

where $l_s \in \mathbb{R}$ is the known arc length of the section, and the notation $[\cdot]_{xy}$ denotes a matrix element at row x and column y. With the knowledge of all the joint variables $q(t)$ as computed from Eq. (27), T_0^4 of Eq. (7), and consequently, the Jacobian $J(q)$ of Eq. (10) can now be calculated online.

Remark 1 We assume that the constant rotation matrix R_s^* is known. This is a mild assumption, since we assume that

https://doi.org/10.1017/S0263574707003475 Published online by Cambridge University Press

the robot can be set to a known reference configuration, or that the constant rotation matrix R_s^* can be obtained a priori using various methods (e.g., a second camera, Euclidean measurements, etc.).

4. Task-space kinematic controller

The control objective is the regulation of the end-effector of the manipulator to a desired position and orientation, denoted by $p_d \in \mathbb{R}^6$. Note that this desired configuration of the robot may be available as an image, and the technique described in the previous sections may be applied to compute p_d. The mismatch between the desired and actual end-effector Cartesian coordinates is the task-space position error, denoted by $e(t) \in \mathbb{R}^6$, and given as

$$e \triangleq p - p_d. \tag{28}$$

Utilizing the velocity kinematics in Eq. (10), and the fact that $\dot{p}_d = 0$, the open-loop error dynamics for $e(t)$ can be expressed as

$$\dot{e} = J\dot{q}. \tag{29}$$

The kinematic control input $\dot{q}(t)$ is designed[10] as

$$\dot{q} = -J^+\beta e + (I_8 - J^+J)g \tag{30}$$

where $\beta \in \mathbb{R}^{6\times 6}$ is a diagonal positive definite gain matrix, $I_n \in \mathbb{R}^{n\times n}$ denotes the $n \times n$ identity matrix, and $J^+(q)$ denotes the pseudoinverse[20] of $J(q)$, defined as

$$J^+ \triangleq J^{\mathrm{T}}(JJ^{\mathrm{T}})^{-1}. \tag{31}$$

In Eq. (30), $g(t) \in \mathbb{R}^8$ is a bounded auxiliary signal that is constructed according to a sub-task control objective such as obstacle avoidance. For example, if the joint-space configuration that avoids an obstacle in the manipulator's work space is known to be q_r, then $g(t)$ can be designed as

$$g \triangleq \gamma(q_r - q) \tag{32}$$

where $\gamma \in \mathbb{R}$ is a positive gain constant. While designing $\dot{q}(t)$ in the manner of Eq. (30), we make the assumption that the minimum singular value of the Jacobian, denoted by σ_m, is greater than a known small positive constant $\delta > 0$, such that $\max\{\|J^+(q)\|\}$ is known *a priori* and all kinematic singularities are avoided. Note that $J^+(q)$ satisfies the following equalities

$$JJ^+ = I_n \tag{33}$$

$$J(I_8 - J^+J) = 0. \tag{34}$$

After substituting the control input of Eq. (30) in Eq. (29), we obtain

$$\dot{e} = -\beta e \tag{35}$$

where Eqs. (33) and (34) have been utilized. Hence, $e(t)$ is bounded by the following exponentially decreasing envelope

$$\|e(t)\| \leq \|e(0)\| \exp(-\lambda t) \tag{36}$$

where $\lambda \in \mathbb{R}$ is the minimum eigenvalue of β.

From Eqs. (28) and (36), it is clear that $p(t) \in \mathcal{L}_\infty$. Based on the assumption that kinematic singularities are avoided, $J(t)$ is always defined and bounded. Hence, the control input $\dot{q}(t)$ is bounded since $J^+(q)$ is bounded for all possible $q(t)$, and $g(t)$ is bounded by the assumption. We make the assumption that if $p(t) \in \mathcal{L}_\infty$, then $q(t) \in \mathcal{L}_\infty$. From Eq. (29) we get $\dot{e}(t)$, $\dot{p}(t) \in \mathcal{L}_\infty$.

5. Simulations and discussion

The primary contribution of this paper is the development of a vision-based technique for measuring the shape of a continuum robot, given the expressions for the forward kinematics. For the sake of demonstration, we chose a simpler task-space kinematic controller formulated as

$$\dot{q} = -J^{\mathrm{T}}\beta e. \tag{37}$$

Substituting this control input in the error dynamics of Eq. (29) results in the same exponential stability result as in Eq. (36), except that λ is now the minimum eigenvalue of $JJ^{\mathrm{T}}\beta$. The desired task-space position of the end-effector was selected to be

$$p_d = [0.30 \;\; 0.01 \;\; 0.769 \;\; 0.0 \;\; 0.4 \;\; 0.0]^{\mathrm{T}} \tag{38}$$

where the first three elements in the vector denote the desired end-effector position specified in meters, while the remaining three elements denote the orientation defined using the Euler angle notation in radians. The initial configuration of the manipulator, denoted by $q(t_0)$, was selected as

$$q(t_0) = [0.01 \;\; 0.1 \;\; 0.01 \;\; 0.1 \;\; 0.01 \;\; 0.1 \;\; 0.01 \;\; 0.1]^{\mathrm{T}} \tag{39}$$

which is close to the relaxed configuration of the manipulator, where all sections lie extended along the principal axis. Note that the joint-space configuration of the manipulator is defined as in Eq. (9), where the units of $\phi_i(t)$ and $k_i(t)$ are radians and meters^{-1}, respectively. The diagonal elements of feedback gain matrix β were set to 30. Based on calibration parameters from an actual camera, the internal camera calibration matrix A of Eq. (17) was set as

$$A = \begin{bmatrix} 1268.16 & 0 & 257.49 \\ 0 & 1267.51 & 253.10 \\ 0 & 0 & 1 \end{bmatrix}. \tag{40}$$

The reference image of the robot was contructed from a configuration where the robot is fully extended along its backbone. The simulation results for task-space control, utilizing the simplified controller of Eq. (37) and the vision-based joint-space variable measurement described in Section 3, are shown in Figs. 3 and 4. Figure 3 shows the time evolution of task-space error in the positioning of the robot end-effector, while Fig. 4 shows the joint-space trajectories.

In a real-world implementation, multiple cameras at known positions relative to the base frame of the robot will be required to successfully track all visual markers on the robot and avoid problems of occlusion. Utilizing the technique

https://doi.org/10.1017/S0263574707003475 Published online by Cambridge University Press

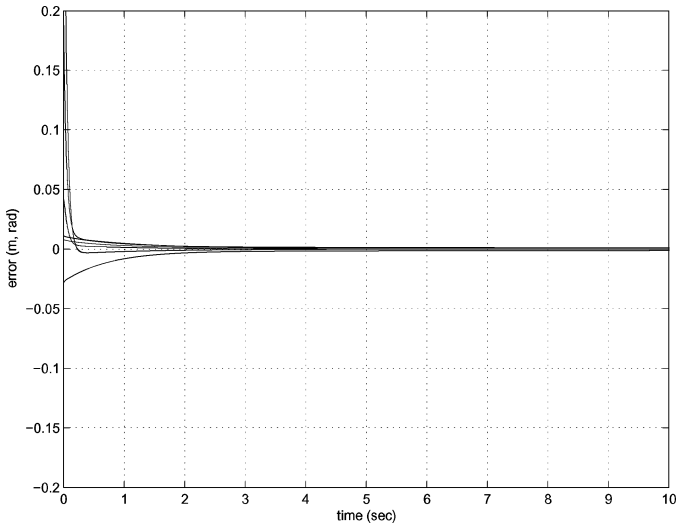

Fig. 3. End effector position error.

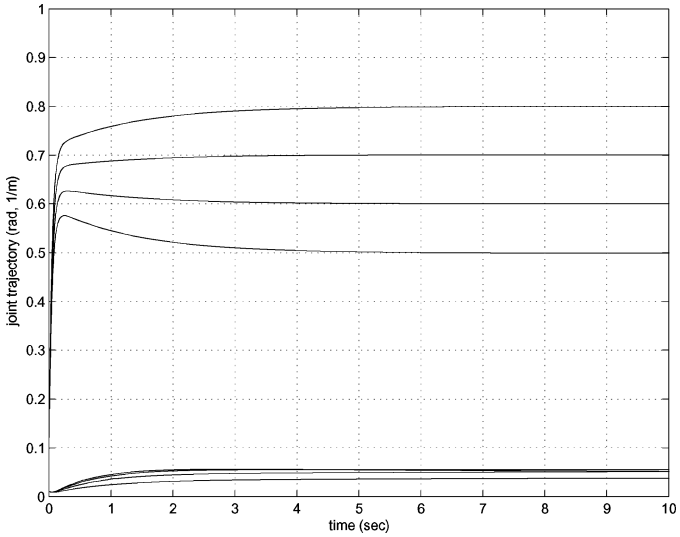

Fig. 4. Time evolution of joint trajectories.

presented in this paper, it is possible to accomplish more than just end-effector regulation. Since all joint variables are recovered from the processing of images, a joint-level controller may also be implemented which will enable complete shape control of the robot (i.e., the manipulator may be servoed to any desired shape, given an image of the manipulator at that configuration). The result may be further extended to shape tracking, if a video sequence of the desired trajectory of the robot body is available.

6. Conclusions

In this paper, we presented a kinematic controller to exponentially regulate the end-effector of a continuum robot to a desired position and orientation, using visual feedback from a fixed camera. By exploiting the homography-based techniques and the known kinematics of the robot, it was shown that the shape of the robot arm can be completely determined through the 2D images from the camera. The only requirement is that a reference orientation of the end of each section of the robot must be known relative to the camera coordinate frame.

References

1. G. Robinson and J. B. C. Davies, "Continuum Robots—A State of the Art," *Proceedings of the IEEE Conference on Robotics and Automation*, Detroit, Michigan (1999) pp. 2849–2854.
2. J. B. C. Davies, A Flexible Motion Generator, *Ph.D. Thesis* (Edinburgh: Heriot-Watt University, 1996).
3. M. W. Hannan and I. D. Walker, "Analysis and experiments with an elephant's trunk robot," *Int. J. Robot. Soc. Jpn.* **15**(8), 847–858 (2001).
4. M. W. Hannan and I. D. Walker, "Real-time shape estimation for continuum robots using vision," *Robotica* **23**(5), 645–651 (Sep. 2005).
5. O. Faugeras and F. Lustman, "Motion and structure from motion in a piecewise planar environment," *Int. J. Pattern Recog. Artif. Intell.* **2**(3), 485–508 (1988).
6. J. Chen, A. Behal, D. Dawson and Y. Fang, "2.5D Visual Servoing With a Fixed Camera," *Proceedings of the American Control Conference*, Denver, Colorado (Jun. 2003) pp. 3442–3447.
7. B. A. Jones, Kinematics and Implementation of Continuum Manipulators, *Ph.D. Thesis* (Clemson, South Carolina: Department of Electrical and Computer Engineering, Clemson University, 2005).
8. M. Kircanski and M. Vukobratovic, "Contribution to control of redundant robotic manipulators in an environment with obstacles," *Int. J. Robot. Res.* **5**(4), 112–119 (1986).
9. B. Siciliano, "Kinematic control of redundant robot manipulators: A tutorial," *J. Intell. Robot. Syst.* **3**, 201–212 (1990).
10. T. Yoshikawa, "Analysis and Control of Robot Manipulators With Redundancy," **In:** Robotics Research: The First International Symposium (MIT Press, Cambridge, Massachusetts, 1984) pp. 735–747.
11. G. S. Chirikjian and J. W. Burdick, "The kinematics of hyper-redundant robot locomotion," *IEEE Trans. Robot. Autom.* **11**(6), 781–793 (1995).
12. H. Choset and W. Henning, "A follow-the-leader approach to serpentine robot motion planning," *J. Aerospace Eng.* **12**(2), 65–73 (1999).
13. S. Ma, I. Kobayashi, S. Hirose and K. Yokoshima, "Control of a multijoint manipulator "Moray arm," *IEEE/ASME Trans. Mechatron.* **7**(3), 304–317 (2002).
14. M. W. Hannan, Theory and Experiments With an 'Elephant's Trunk' Robotic Manipulator, *Ph.D. Thesis* (Clemson, South Carolina: Department of Electrical and Computer Engineering, Clemson University, 2002).
15. M. W. Spong and M. Vidyasagar, *Robot Dynamics and Control* (Wiley, New York, 1991).
16. O. Faugeras, *Three-Dimensional Computer Vision* (MIT Press, Cambridge, Massachusetts, 2001).
17. E. Malis and F. Chaumette, "2 1/2 D visual servoing with respect to unknown objects through a new estimation scheme of camera displacement," *Int. J. Comp. Vis.* **37**(1), 79–97 (Jun. 2000).
18. R. Sukthankar, R. Stockton, and M. Mullin, "Smarter Presentations: Exploiting Homography in Camera-Projector Systems," *Proceedings of the International Conference on Computer Vision* (2001) pp. 247–253, Vancouver, Canada.
19. Z. Zhang and A. R. Hanson, "Scaled Euclidean 3D Reconstruction Based on Externally Uncalibrated Cameras," *Proceedings of the IEEE Symposium on Computer Vision*, Coral Gables, Florida (1995) pp. 37–42.
20. T. L. Boullion and P. L. Odell, *Generalized Inverse Matrices* (Wiley, New York, 1971).

https://doi.org/10.1017/S0263574707003475 Published online by Cambridge University Press