Setpoint Regulation of Continuum Robots Using a Fixed Camera

Vilas K Chitrakaran[†], Aman Behal[‡], Darren M Dawson[†] and Ian D Walker[†]

[†]Electrical and Computer Engineering
Clemson Univ., Clemson, SC 29634-0915

[‡]Electrical and Computer Engineering
Clarkson Univ., Potsdam, NY 13699

email: cvilas,ddawson,ianw@ces.clemson.edu; abehal@clarkson.edu

Abstract

In this paper, we investigate the problem of measuring the shape of a continuum robot using visual information from a fixed camera. Specifically, we capture the motion of a set of fictitious planes, each formed by four 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 end-effector of the robot to a constant desired position and orientation, while at the same time using its kinematic redundancy to satisfy a subtask objective such as obstacle avoidance.

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 is limited by the number of actuated joints in them. In contrast, continuum robots [15] 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 so that their end-effector can be positioned at a desired location and orientation while concurrently satisfying constraints in their work-space such as obstacles and tight spaces. 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 (see references in [15]), the dual problem of sensing the configuration of such robots still remains 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. In [5], a model has been proposed to relate internal bellow pressures to the position of the end-effector in a fluid operated device. A model that infers position from measurements of change in tendon length in tendon driven devices has been proposed in [10]. 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 *et al.* [12] 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 is only applicable to the case where the motion of the arm is 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. In [7], it was shown that the correspondence between images of feature points lying on a plane, as obtained from two different cameras, is a collineation, 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 reference image of the plane and a knowledge of its rotation relative to the coordinate frame of the first camera, the position and orientation of the plane can be determined relative to the camera from images obtained using just the first camera alone. Exploiting this technique, in [2] Chen *et al.* 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. Then, from decomposition of the homography and from the equations

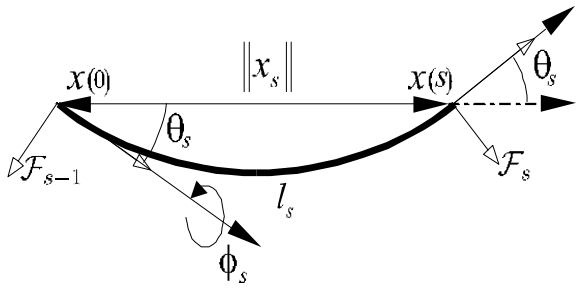

Figure 1: A planar curve

describing the forward kinematics of the robot as developed in [9], we show that the curvatures that define the shape of various sections of the robot can be fully determined. From the various kinematic control strategies for hyperredundant robots that have appeared in the robotics literature in the past (*e.g.*, [13, 16]), we use the work in [19] to develop a kinematic controller to demonstrate that the robot end-effector can be forced to any desired position and orientation using a sequence of images from a single external video camera.

This paper is organized as follows. Section 2 will introduce 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 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 a brief description of the experimental setup. Concluding remarks are given 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 [17]. 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. A natural way to describe the kinematics of a continuum robot is by using the concept of curvature [3, 8, 9, 11]. One such continuum robot is the Clemson Elephant Trunk [9] which is composed of sixteen two degree-of-freedom joints divided into four sections. For the purposes of this paper, it is sufficient to say that each section is designed to bend with a constant planar curvature. Each section can also be rotated out of plane relative to the preceding section. However, torsion is not possible within the section. Readers interested in the design details of the robot are referred to [9].

Consider the s^{th} section of the robot. Using differential 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 Figure 1. Here $x_s \in \mathbb{R}^3$ is the position vector of the endpoint of the curve relative to its initial point, and

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

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

Treating the two rotations in the curve as discrete rotational joints and the translation as a coupled discrete prismatic joint, standard Denavit-Hartenberg procedure [17] 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 follows [9]

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

The out-of-plane rotation can be modelled as an additional rotational joint with rotation of angle ϕ_s about the initial tangent of the curve (see Figure 1). Hence, for the 3D case, the forward kinematics for the s^{th} 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

$$R^s_{s-1} = \begin{bmatrix} \cos(\phi_s)\cos(k_s l_s) & -\sin(\phi_s) & -\cos(\phi_s)\sin(k_s l_s) \\ \sin(\phi_s)\cos(k_s l_s) & \cos(\phi_s) & -\sin(\phi_s)\sin(k_s l_s) \\ \sin(k_s l_s) & 0 & \cos(k_s l_s) \end{bmatrix} \tag{5}$$

$$t^s_{s-1} = \begin{bmatrix} -\frac{1}{k_s}\cos(\phi_s) + \frac{1}{k_s}\cos(\phi_s)\cos(k_s l_s) \\ -\frac{1}{k_s}\sin(\phi_s) + \frac{1}{k_s}\sin(\phi_s)\cos(k_s l_s) \\ \frac{1}{k_s}\sin(k_s l_s) \end{bmatrix}. \tag{6}$$

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 s^{th} curved section to the coordinate frame $\mathcal{F}_{s-1}$ defined at the end of the $(s-1)^{th}$ section. In the above equations, $R^s_{s-1} \in SO(3)$ and $t^s_{s-1} \in \mathbb{R}^3$ 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

$$T^4_0 = A_1 A_2 A_3 A_4. \tag{7}$$

From (7) the end-effector position and orientation in the task-space of the robot, denoted by $p(t) = \begin{bmatrix} x & y & z & \theta_x & \theta_y \end{bmatrix}^T \in \mathbb{R}^5$, can be written as follows

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

where $f(q) \in \mathbb{R}^5$ denotes the forward kinematics, and $q(t) \in \mathbb{R}^8$ denotes the joint space variables for the robot defined in the following manner

$$q(t) = \begin{bmatrix} \phi_1 & k_1 & \phi_2 & k_2 & \phi_3 & k_3 & \phi_4 & k_4 \end{bmatrix}^T. \quad (9)$$

Note that θ_z is not defined as a task-space variable due to the fact that torsion is not mechanically possible about the spine of the robot.

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

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

where $J(q) \triangleq \dfrac{\partial f(q)}{\partial q} \in \mathbb{R}^{5x8}$ 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 the following section, 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 Figure 2). Let $\mathcal{F}_0$ denote the fixed orthogonal coordinate system whose origin coincides with the base of the robot. At the end of s^{th} section of the robot, consider a transverse plane π_s defined by four non-collinear 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 denoted by π_s^*, (with four non-collinear 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 s^{th} section is at a reference position and orientation relative to the fixed camera, *i.e.*, π_s^*'s and $\mathcal{F}_s^*$'s are defined by a reference image[1] of the robot. To initiate the geometric analysis, let us express 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, in the inertial coordinate system $\mathcal{I}$ as follows

$$\bar{m}_{si} \triangleq \begin{bmatrix} x_{si} & y_{si} & z_{si} \end{bmatrix}^T \quad (11)$$

$$\bar{m}_{si}^* \triangleq \begin{bmatrix} x_{si}^* & y_{si}^* & z_{si}^* \end{bmatrix}^T. \quad (12)$$

In order to facilitate further geometric development, we define normalized Euclidean coordinates, denoted by $m_{si}(t)$, $m_{si}^* \in \mathbb{R}^3$ for the above target points as follows

$$m_{si} \triangleq \frac{\bar{m}_{si}}{z_{si}} = \begin{bmatrix} \frac{x_{si}}{z_{si}} & \frac{y_{si}}{z_{si}} & 1 \end{bmatrix}^T \quad (13)$$

$$m_{si}^* \triangleq \frac{\bar{m}_{si}^*}{z_{si}^*} = \begin{bmatrix} \frac{x_{si}^*}{z_{si}^*} & \frac{y_{si}^*}{z_{si}^*} & 1 \end{bmatrix}^T. \quad (14)$$

[1] Note that this reference image is not related to the desired position to which we want to regulate the end-effector of the robot.

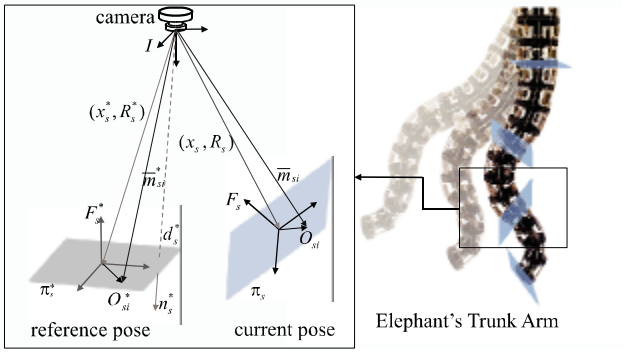

Figure 2: Coordinate frame relationships

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

$$p_{si} = \begin{bmatrix} u_{si} & v_{si} & 1 \end{bmatrix}^T \qquad p_{si}^* = \begin{bmatrix} u_{si}^* & v_{si}^* & 1 \end{bmatrix}^T. \quad (15)$$

The projected pixel coordinates of the target points are related to the normalized task-space coordinates by the pinhole lens model of [6] such that

$$p_{si} = Am_{si} \qquad p_{si}^* = Am_{si}^* \quad (16)$$

where $A \in \mathbb{R}^{3\times3}$ is a known, constant, and invertible intrinsic camera calibration matrix that is explicitly defined as [14]

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

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 s^{th} 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 a known constant rotation matrix[2] between $\mathcal{F}_s^*$ and $\mathcal{I}$. Let $x_0 \in \mathbb{R}^3$ and $R_0 \in SO(3)$ be the translation vector and the known constant rotation matrix, respectively,

[2] The subsequent development requires that the constant rotation matrix R_s^* be known. This is considered to be a mild assumption since the constant rotation matrix R_s^* can be obtained a priori using various methods (*e.g.*, a second camera, Euclidean measurements, *etc.*).

between $\mathcal{F}_0$ at the base of the robot and the camera frame $\mathcal{I}$. As also illustrated in Figure 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 Figure 2, we can arrive at the following relationships

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

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

After solving (20) for O_{si} and substituting the resulting expression into (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 new rotational and translational variables, respectively, defined as follows

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

Using (18), the relationship in (21) can now be expressed as follows

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

Utilizing (13) and (14), we obtain the following relationship in terms of normalized Euclidean coordinates of the feature points

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

where $\alpha_{si}(t) \in \mathbb{R}$ is the depth ratio, and $H_s(t) \in \mathbb{R}^{3x3}$ denotes the Euclidean homography between the coordinate systems $\mathcal{F}_s$ and $\mathcal{F}_s^*$. Given the relationships of (16), the above relationship can be written in terms of pixel coordinates of the target points in each plane in the following manner

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

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

Given the images of four points $p_{si}(t)$ on each plane π_s and the images of the corresponding reference points p_{si}^* in π_s^*, we can solve the linear set of equations in (25) to determine $G_s(t)$ and $\alpha_{si}(t)$ (*e.g.,* see [18]). Since the camera calibration matrix A in (17) is known, $H_s(t)$ can be obtained from $G_s(t)$ for each section of the manipulator. By utilizing various techniques (*e.g.*, see [7, 20]), $H_s(t)$ can be decomposed into rotational and translational components as in (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 s^{th} section of the robot relative to the camera fixed frame $\mathcal{I}$, can then be computed from $\bar{R}_s(t)$ by using (22) and the fact that R_s^* is known apriori.

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 $\mathcal{F}_s$. Since the rotation matrix R_0 between $\mathcal{I}$ and $\mathcal{F}_0$ is assumed to be known, we can progressively compute $R_{s-1}^s(t)$ in (5), *i.e.*, the rotation matrix from one section of the robot to the next, as follows [17]

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

From (4), the joint space variables for the s^{th} section can hence be determined as

$$\begin{aligned} \phi_s &= \cos^{-1}([R_{s-1}^s]_{22}) \\ k_s &= \frac{1}{l_s} \cos^{-1}([R_{s-1}^s]_{33}) \end{aligned} \tag{27}$$

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 exact knowledge of all the joint variables in $q(t)$, T_0^4 of (7), and consequently, the Jacobian $J(q)$ of (10) can be determined.

Remark 1 *If the points O_{si} are not coplanar, then the estimation of $G(t)$ is a nonlinear problem that requires at least eight points using the algorithm presented in [14].*

4 Task-Space Kinematic Controller Development

The control objective is to regulate the end-effector of the robot arm to the position and orientation of the end-effector as defined in the reference image. We quantify the mismatch between the desired and actual end-effector Cartesian coordinates as a task-space position error, denoted by $e(t) \in \mathbb{R}^5$, as follows

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

where $p_d \in \mathbb{R}^5$ denotes a desired task-space setpoint. The open loop error dynamics for $e(t)$ can be expressed as

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

where (10) was utilized, along with the fact that $\dot{p}_d = 0$. Our objective is to design a control input $\dot{q}(t)$ to ensure regulation of $e(t)$ in the sense that

$$\lim_{t \to \infty} e(t) = 0 \ . \tag{30}$$

To facilitate the control development, we define the pseudo-inverse [1] of $J(q)$ in (10), denoted by $J^+(q) \in \mathbb{R}^{8x5}$, as follows

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

$J^+(q)$ satisfies the following equality

$$JJ^+ = I_n \tag{32}$$

where $I_n \in \mathbb{R}^{nxn}$ denotes the $n \times n$ identity matrix. The matrix $(I_8 - J^+J)$, which projects vectors into null-space of $J(q)$, satisfies the property that

$$J(I_8 - J^+J) = 0. \tag{33}$$

Following the approach in [19], we can now design $\dot{q}(t)$ as

$$\dot{q} = -J^{+}\beta e + (I_8 - J^{+}J)g \tag{34}$$

where $\beta \in \mathbb{R}^{5x5}$ is a diagonal, positive definite gain matrix, and $g(t) \in \mathbb{R}^8$ is a bounded auxiliary signal that is constructed according to the sub-task control objective such as obstacle avoidance. For example, if the joint-space configuration that avoids an obstacle in the manipulator's workspace is known to be q_r, then $g(t)$ can be designed as

$$g \triangleq \gamma(q_r - q) \tag{35}$$

where $\gamma \in \mathbb{R}$ is a positive gain constant. In designing $\dot{q}(t)$ in the manner of (34), we make the assumption that the minimum singular value of the robot Jacobian, denoted by σ_m, is greater than a known small positive constant $\delta > 0$, such that $max\left\{\left\|J^{+}(q)\right\|\right\}$ is known apriori and all kinematic singularities are avoided.

After substituting the control input of (34) in (29), we obtain

$$\dot{e} = -\beta e \tag{36}$$

where (32) and (33) have been utilized. After solving (36), we can show that $e(t)$ is bounded by the following exponentially decreasing envelope

$$\|e(t)\| \leq \|e(0)\| \exp(-\lambda t) \tag{37}$$

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

Since $e(t)$ is bounded, from (36), $\dot{e}(t) \in \mathcal{L}_\infty$. From the fact that p_d is bounded by definition, we can utilize (28) and (29) to prove that $p(t)$ and $\dot{p}(t)$ are bounded for all time. From (29), with the previously stated assumption that the robot is never close to a kinematic singularity, the control input $\dot{q}(t)$ is bounded since $J(q)$ and $J^{+}(q)$ are bounded for all possible $q(t)$, and $g(t)$ is bounded by assumption. Note that if $g(t)$ is chosen as in (35), $q(t)$ is bounded since q_r is bounded by definition. In general, boundedness of $q(t)$ cannot be proved unless $g(t)$ is known to be of a similar form as chosen here.

5 Experimental Setup

In order to verify the proposed technique, an experimental testbed is being setup with the following components: (a) Clemson Elephant Trunk robot, (b) DALSA CA-D6-0256W high-speed (955 frames-per-second) grayscale CCD video camera, (c) Bitflow Roadrunner 24M framegrabber board, (d) two Intel 2.4 GHz based Personal Computers (PC) running QNX 6.2.1 Momentics realtime operating system, and (e) a teleoperator input system. Multiple cameras may be used in a checkerboard pattern, along with multiple framegrabbers to cover the whole workspace of the robot and at the same time avoid problems such as occlusions. The image processing operations will be performed by one PC (or more PCs in case of multiple cameras) and data corresponding to visually tracked feature points on the robot are communicated to the second PC through a fast ethernet link. The kinematic control, motion planning and the corresponding I/O operations associated with the robot are performed by the second PC. This PC will have standard I/O boards (with A/D, D/A, digital I/O and encoder inputs) that provides a hardware interface to the actuators and sensors on the robot, as well as to the teleoperation inputs from a pair of 3 degree-of-freedom joysticks. The control software is implemented in QMotor 3.0 [4], a PC based realtime multitasking graphical control development environment.

6 Conclusions

In this paper, we presented a kinematic controller to exponentially regulate the end-effector of the a continuum robot to a desired position and orientation using visual feedback from a fixed camera. By exploiting homography based techniques and known kinematics of the robot, it was shown that the shape of the robot arm can be completely determined from 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. Future work will include simulations and experimental verification of this technique on the Elephant Trunk arm at Clemson University.

References

[1] T. L. Boullion, and P. L. Odell, *Generalized Inverse Matrices,* Wiley, New York, 1971.

[2] J. Chen, A. Behal, D. Dawson, and Y. Fang, "2.5D Visual Servoing with a Fixed Camera," *Proceedings of the American Control Conference*, Denver, CO, pp. 3442-3447, June 2003.

[3] G. S. Chirikjian, "Theory and Applications of Hyper-Redundant Robotic Manipulators," *Ph.D. Thesis, Department of Applied Mechanics*, California Institue of Technology, June 1992.

[4] N. Costescu, M. Lo- er, M. Feemster, and D. Dawson, "QMotor 3.0 - An Object Oriented System for PC Control Program Implementation and Tuning," *Proceedings of the American Control Conference*, Arlington, VA, June 2001, pp. 4526-4531.

[5] J. B. C. Davies, "A Flexible Motion Generator," *PhD Thesis*, Heriot-Watt University, Edinburgh, 1996.

[6] O. Faugeras, *Three-Dimensional Computer Vision*, The MIT Press, Cambridge Massachusetts, 2001.

[7] O. Faugeras and F. Lustman, "Motion and Structure From Motion in a Piecewise Planar Environment," *International Journal of Pattern Recognition and Artificial Intelligence,* Vol. 2, No. 3, pp. 485-508, 1988.

[8] I. Gravagne, I. D. Walker, "On the Kinematics of Remotely-Actuated Continuum Robots," *IEEE International Conference on Robotics and Automation*, San Francisco, pp. 2544-2550, May 2000.

[9] M. W. Hannan, "Theory and Experiments with an 'Elephant's Trunk' Robotic Manipulator," *PhD Thesis, Department of Electrical and Computer Engineering*, Clemson University, 2002.

[10] M. W. Hannan, I. D. Walker, "Analysis and Experiments with an Elephant's Trunk Robot," *International Journal of the Robotics Society of Japan*, Vol. 15, No. 8, pp. 847-858, 2001.

[11] M. Hannan, I. D. Walker, "Novel Kinematics for Continuum Robots," *7th International Symposium on Advances in Robot Kinematics*, Piran, Slovenia, pp. 227-238, June 2000.

[12] M.W. Hannan, I. D. Walker, "Vision Based Shape Estimation for Continuum Robots," *IEEE International Conference on Robotics and Automation*, September 2003, accepted, to appear.

[13] M. Kircanski, and M. Vukobratovic, "Contribution to Control of Redundant Robotic Manipulators in an Environment with Obstacles," The International Journal of Robotics Research, Vol. 5, No. 4. pp. 112-119, 1986.

[14] E. Malis, F. Chaumette, "2 1/2 D Visual Servoing with Respect to Unknown Objects Through a New Estimation Scheme of Camera Displacement," *International Journal of Computer Vision*, Vol. 37, No. 1, pp. 79-97, June 2000.

[15] G. Robinson, J. B. C. Davies, "Continuum Robots - A State of the Art," *IEEE Conference on Robotics and Automation*, pp. 2849-2854, 1999.

[16] B. Siciliano, "Kinematic Control of Redundant Robot Manipulators: A Tutorial," *Journal of Intelligent and Robotic Systems*, Vol. 3, pp. 201-212, 1990.

[17] M.W. Spong and M. Vidyasagar, M., *Robot Dynamics and Control*, John Wiley and Sons, New-York, 1991.

[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.

[19] T. Yoshikawa, "Analysis and Control of Robot Manipulators with Redundancy," *Robotics Research - The First International Symposium,* MIT Press, pp. 735-747, 1984.

[20] Z. Zhang and A. R. Hanson, "Scaled Euclidean 3D Reconstruction Based on Externally Uncalibrated Cameras," *IEEE Symposium on Computer Vision*, pp. 37-42, 1995.