Model-Free Visually Servoed Deformation Control of Elastic Objects by Robot Manipulators

David Navarro-Alarcón, *Student Member, IEEE*, Yun-Hui Liu, *Fellow, IEEE*, José Guadalupe Romero, and Peng Li

***Abstract*—Despite the recent progress in physically interactive and surgical robotics, the active deformation of compliant objects remains an open problem. The main obstacle to its implementation comes from the difficulty to identify or estimate the object's deformation model. In this paper, we propose a novel vision-based deformation controller for robot manipulators interacting with unknown elastic objects. We derive a new dynamic-state feedback velocity control law using the passivity-based framework. Our method exploits visual feedback to estimate the deformation Jacobian matrix in real time, avoiding any model identification steps. We prove that even in the presence of inexact estimations, the closed-loop dynamical system ensures input-to-state stability (i.e., full dissipativity) with respect to external disturbances. An experimental study with several deformation tasks is presented to validate the theory.**

***Index Terms*—Contact modeling, deformable models, dexterous manipulation, shape control, visual servoing.**

I. Introduction

THE deformation control problem arises in applications where a mechanical manipulator needs to actively change the shape of a deformable object into a desired form [1]. It is needed in surgery to automate delicate procedures with soft tissues, e.g., suturing [2] and needle insertion [3], or in the food industry to automate the shaping of rheological food materials [4], to name a few potential applications. One of the main issues that hampers the successful implementation of these tasks is the difficulty in identifying or estimating the object's deformation properties. This information is needed since, in order to provide a desired form to the object, we need to know how the motion of the manipulator is mapped to the deformation of the body.

In this paper, we address the active deformation control problem of unknown/unmodeled compliant objects. To avoid the open-loop deformation response that arises with traditional robot-centered manipulation approaches (see e.g. [5] and [6]), in our method we exploit real-time visual feedback to explicitly servo control the object's shape. For that, we restrict our attention to physical interactions with objects that exhibit or can be modeled based on elastic deformations only. Our method is formulated for velocity-controlled robot manipulators that interact in a fully constrained fashion with the purely elastic material.

A. Related Work

As a result of its economically important applications [1], the automatic deformation of soft materials is currently an active topic of research. For example, in [7], a model-based deformation control method for rheological objects is presented. This controller proposes decomposing the deformation into elasticity and plasticity, and by using a two-phase (offline parameter identification, then model-based control) strategy, the desired shape of an object is achieved. The problem of simultaneous motion and deformation of soft objects is addressed in [8]. This study proposes a model-free PID control law for both position and effort-controlled mechanical systems. However, similar to [7], the approach in [8] only formulates the problem for a simple 1-D deformation, that is, compression of the body.

Some works formulate the deformation problem considering more degrees of freedom (DOFs). For example, in [9] an iterative controller to indirectly position multiple *points* of interest on a deformable object is reported. This robust controller is formulated for multiple manipulators, and its implementation requires *a priori* knowledge of an approximated deformation model of the object. This indirect positioning problem is also addressed in [10] and [11], where different degrees of knowledge of the deformation model is similarly required by both control methods.

Note that in [9]–[11], the object's deformation is only characterized by the explicit displacement regulation of different visual points. However, some control approaches are formulated considering more general deformation features, such as the object's contour for example. In [4], the development of an automatic forming machine for food dough is reported. Based on identified deformation parameters (a reasonable step when considering rheological materials), this paper proposes a high-level control method to give a desired shape to the dough. In [12], a control approach to deform the shape of a flexible body (modeled with the finite-element method) by multiple torque-controlled

Manuscript received December 8, 2012; revised May 3, 2013; accepted July 21, 2013. Date of publication August 15, 2013; date of current version December 2, 2013. This paper was recommended for publication by Associate Editor M. C. Cavusoglu and Editor B. J. Nelson upon evaluation of the reviewers' comments. This work is supported in part by the Hong Kong Research Grant Council under Grant 415011, in part by the Hong Kong Innovation and Technology Fund under Project ITS/475/09, in part by the Shun Hing Institute of Advanced Engineering, the Chinese University of Hong Kong, and in part by the Conacyt Mexico for scholarship 212019.

D. Navarro-Alarcon and P. Li are with the Department of Mechanical and Automation Engineering, the Chinese University of Hong Kong, Shatin N.T., Hong Kong (e-mail: dnavarro@mae.cuhk.edu.hk; pli@mae.cuhk.edu.hk).

Y.-H. Liu is with the Department of Mechanical and Automation Engineering, the Chinese University of Hong Kong, Shatin, N.T., Hong Kong and also with the State Key Laboratory of Robotics Technology and Systems, Harbin Institute of Technology, Harbin, Heilongjiang 150001, China (e-mail: yhliu@mae.cuhk.edu.hk).

J. G. Romero is with the Laboratoire des Signaux et Systemes, Université Paris-Sud 11, Gif-sur-Yvette 91192, France (e-mail: romerovelazquez@lss.supelec.fr).

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.2013.2275651

manipulators is presented. This study proposes an optimization technique to plan the motion of each manipulator; however, the actual control implementation (which is only tested with numerical simulations) also assumes *a priori* knowledge of the object's exact mathematical model. An interesting two-manipulator automatic cloth folding application is reported in [13].

B. Contribution of This Paper

To contribute to this problem, in this paper, we present a new vision-based method to servo control the deformation of a compliant object. Note that most existing visual servoing controllers for manipulators (e.g., standard methods such as [14]–[16], or adaptive-like controllers such as [17]–[20]) do not address active deformation. In our formulation, the deformation task is not restricted to the explicit displacement regulation of visual points. Instead, we propose a more general definition of a deformation feature vector, which is constructed as a nonlinear function of the visual feedback points. One of the major complications to automatically controling the object's deformation, is to estimate how the motion of the manipulator is mapped to the deformation features. Nevertheless, most control methods in the literature assume *a priori* knowledge of these deformation properties (which are usually obtained offline during testing deformations). This complicates the implementation since in many real applications, e.g., in surgery [21], it is not always possible to perform these physical interactions beforehand. To cope with this problem, we propose a model-free method that estimates the object's deformation Jacobian matrix in real time.

Compared with previously reported deformation controllers, our method does not require the exact or approximated deformation model of the body. Therefore, modeling and parameter identification steps are not needed. The method reported in [9] is similar to our controller. However, in contrast to our vision/deformation model-free formulation, the controller in [9] requires an approximated deformation model, a fully calibrated vision system, and only considers plane motion of the manipulators.

To ensure the stability of the closed-loop system, we propose an innovative dynamic-state feedback velocity control law, derived with a robust energy shaping design. The aim of this control law is to recast the deformation problem into a nonconservative Hamiltonian dynamical system [22], such that the desired deformation is achieved by the minimization of an energy-like functional. To the best of our knowledge, this is the first time a deformation controller is formulated using this methodology. We prove that even in the presence of inexact estimations of the deformation Jacobian matrix, the closed-loop system is endowed with input-to-state stability (ISS) [23], that is, full dissipativity with respect to external disturbances. An experimental study with two open-architecture robot manipulators that perform several deformation tasks is presented to validate the theory.

An early version of this study, which considers plane motion and a calibrated camera, is reported in [24]. In our new method we develop a controller to coordinate the motion of manipulators that perform *3-D* deformations in an uncalibrated manner. The extension to 3-D motions allows us to increase the number of controllable deformation DOF.

C. Organization

The rest of this paper is organized as follows. Section II presents the mathematical modeling. Section III presents the general formulation of the deformation controller. Section IV describes cases of study for specific visually servoed deformation tasks. Section V reports the conducted experimental study. Section VI gives final conclusions and future work.

II. Modeling

A. Notation

Column vectors and matrices are, respectively, denoted by small and capital bold letters, e.g., $\mathbf{v} \in \mathbb{R}^h$ and $\mathbf{M} \in \mathbb{R}^{g\times h}$. Errors are denoted by $\Delta\mathbf{v} = \mathbf{v} - \mathbf{v}^* \in \mathbb{R}^h$, for $\mathbf{v}^*$ as the desired constant reference. The identity and zero matrices are respectively denoted by $\mathbf{I}_{h\times h} \in \mathbb{R}^{h\times h}$ and $\mathbf{0}_{h\times g} \in \mathbb{R}^{h\times g}$. Time-varying quantities are accompanied with a bracket (t).

B. Robot Manipulator

Consider the motion of a serial robot manipulator. The vectors of joint displacements and end-effector constrained position are, respectively, denoted by $\mathbf{q}(t) \in \mathbb{R}^g$ and $\mathbf{x}(t) = \mathbf{x}(\mathbf{q}(t)) \in \mathbb{R}^m$, where the number of joints satisfies $g \geq m$. Note the abuse of notation by using $\mathbf{x}(\cdot)$ as both a coordinate vector and a vectorial functional (this notation is along the paper for other vectorial functionals). We denote the jth coordinate of the constrained end-effector position vector by $x_j(t)$. The differential kinematic equation of the manipulator is given by the expression

$$\dot{\mathbf{x}}(t) = \frac{\partial \mathbf{x}}{\partial \mathbf{q}}(\mathbf{q}(t))\dot{\mathbf{q}}(t) \tag{1}$$

where $\frac{\partial \mathbf{x}}{\partial \mathbf{q}}(\mathbf{q}(t)) \in \mathbb{R}^{m\times g}$ represents the standard Jacobian matrix of the kinematic chain [25].

Along this paper, we consider that the robot manipulator is provided with a low-level velocity control interface (the so-called kinematic control mode [26]). Without loss of generality, we assume that the control architecture of the manipulator allows us to *exactly* set the joint velocity input, here denoted by $\boldsymbol{\omega}(t) \in \mathbb{R}^g$, such that in the kinematic expression (1), the relation $\boldsymbol{\omega}(t) \equiv \dot{\mathbf{q}}(t)$ is always satisfied.

C. Visual Deformation Model

For the addressed problem, we consider robot manipulators with a *fully constrained* end effector in continuous interaction with an unknown elastic object. Physically, this situation refers to the case where the motion (in any direction) of a constrained tip results in a proportional deformation of the elastic body. This scenario can be found, for example, in tasks where a surgical instrument performs the deep palpation of tissues, or when frictional robot fingers compress/deform a soft object.

To servo control the body's deformation, in our approach we make use of k visual feedback points conveniently located on

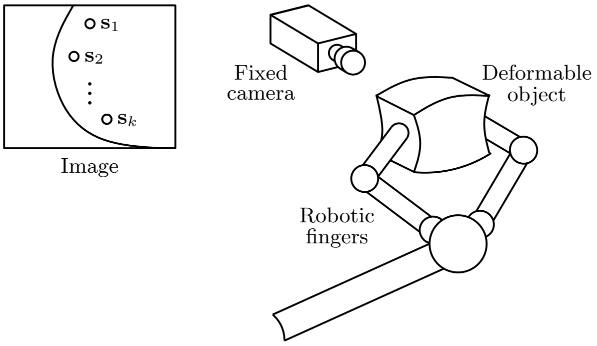


Fig. 1. Conceptual representation of a pair of robot fingers that deform an elastic object, where an uncalibrated fixed camera acquires the visual feedback of k points of interest.

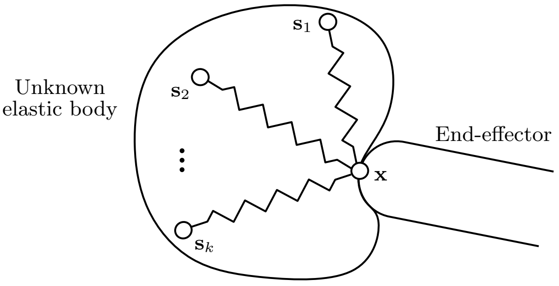


Fig. 2. Conceptual representation of the assumed deformation model, where the manipulator's end effector is virtually interconnected by spatial springs with the k visual feedback points.

the surface of the body. We denote the jth visual point by

$$\mathbf{s}_j(t) = [\,\mu_j(t) \quad \nu_j(t)\,]^\top \in \mathbb{R}^2 \tag{2}$$

where its pixel coordinates $\mu_j(t), \nu_j(t) \in \mathbb{R}$ are acquired in real time by a fixed uncalibrated camera system. For ease of notation, let us define the total visual feedback vector as follows:

$$\mathbf{s}(t) = [\,\mathbf{s}_1^\top(t) \quad \ldots \quad \mathbf{s}_k^\top(t)\,]^\top \in \mathbb{R}^{2k}. \tag{3}$$

A conceptual representation of this deformation setup is depicted in Fig. 1.

Assumption 1: For the robot manipulator that interacts with the deformable body, we assume that (locally) the coordinates of each visual feedback point $\mathbf{s}_j(t)$ can be expressed as an *unknown* smooth function of the end-effector position vector, that is, we express it as $\mathbf{s}_j(t) = \mathbf{s}_j(\mathbf{x}(t))$.

This local assumption is justified for interactions with purely elastic materials that exhibit no rheological deformations. We physically interpret this scenario as a virtual spatial-spring interconnection between a visual feedback point $\mathbf{s}_j(t)$ and the position $\mathbf{x}(t)$. See Fig. 2 for a conceptual representation.

III. Control Design

For the robot manipulator under consideration, we only control a total of m DOFs with its elastically constrained end effector; therefore, in this paper we only address deformation tasks that can be characterized by m DOF. To quantify the deformation of the body, we construct with the k visual feedback points a linearly independent feature vector $\boldsymbol{\eta}(t) = \boldsymbol{\eta}(\mathbf{s}(t)) \in \mathbb{R}^m$. Explicit definitions of this known vectorial function $\boldsymbol{\eta}(\mathbf{s}(t))$ are given in Section IV. Computing the time derivative of $\boldsymbol{\eta}(t)$ yields the following expression:

$$\dot{\boldsymbol{\eta}}(t) = \underbrace{\frac{\partial \boldsymbol{\eta}}{\partial \mathbf{s}}(\mathbf{s}(t))\frac{\partial \mathbf{s}}{\partial \mathbf{x}}(\mathbf{x}(t))}_{\mathbf{A}(\mathbf{x}(t))}\dot{\mathbf{x}}(t). \tag{4}$$

We call the matrix $\mathbf{A}(\mathbf{x}(t)) \in \mathbb{R}^{m\times m}$ the *deformation Jacobian matrix* since it relates the motion of the robot manipulator with the deformation flow $\dot{\boldsymbol{\eta}}(t)$. Note that the exact analytic form of $\mathbf{A}(\mathbf{x}(t))$ is not available since $\frac{\partial \mathbf{s}}{\partial \mathbf{x}}(\mathbf{x}(t))$ is unknown. For our analysis and control design, we make a change in the control input variable such that $\dot{\mathbf{x}}(t) = \mathbf{u}(t) \in \mathbb{R}^m$ represents the new velocity control input to the deformation plant (4). This operation is valid since the matrix $\frac{\partial \mathbf{x}}{\partial \mathbf{q}}(\mathbf{q}(t))$ is exactly known, then we can always design a kinematic controller $\boldsymbol{\omega}(t)$ which commands the manipulator to move at a desired end effector rate $\dot{\mathbf{x}}(t)$.

Problem statement: Given a desired constant deformation feature vector $\boldsymbol{\eta}^* \in \mathbb{R}^m$, design a dynamic-state feedback velocity control law $\mathbf{u}(t)$, such that the closed-loop system is passive, and the minimization of its energy functional performs the desired deformation task.

Remark 1: The kinematic expression given in (4) has been previously reported in [9]–[11]. However, none of these approaches addressed the online estimation of the unknown deformation Jacobian matrix $\mathbf{A}(\mathbf{x}(t))$. For example, Hirai and Wada [9] rightly pointed out that a model of the deformable object was indispensable to determine the motion of the manipulated points. They assumed knowledge of an approximated deformation model and derived a robust controller to cope with uncertainty. In a later work of the same group, Wada *et al.* [10] claimed that the approximated model may not be needed for small deformations, but just a "rough" kinematic relation between the robots and the positioning points. As for the method reported by Smolen and Patriciu [11], it assumed exact *a priori* knowledge of this matrix.

A. Estimating the Deformation Jacobian Matrix

In order to enforce a desired behavior to system (4), we must know how the action of the control input $\mathbf{u}(t)$ deforms the body, and thus generates the flow $\dot{\boldsymbol{\eta}}(t)$. For that, we need to exactly or very closely estimate the matrix $\mathbf{A}(\mathbf{x}(t))$, whose inverse is assumed to exist around the desired operating point. Instrumental to achieve this full rank condition is the proper (i.e., representative) selection of feedback points $\mathbf{s}_j(t)$ as well as an independent (at least locally) definition of the deformation feature vector $\boldsymbol{\eta}(t)$.

We now present, and in later sections qualitatively compare, the following offline and online numerical identification/estimation techniques:

1) Least-Squares Approximation: For this offline identification technique (see [7] for a deformation control application), we assume that around the operating point, the coordinates of the deformation feature vector $\boldsymbol{\eta}(t)$ can be closely approximated by a linearly parametrizable relation

$$\boldsymbol{\eta}(t) \approx \mathbf{W}(\mathbf{x}(t))\mathbf{b} \tag{5}$$

for a regression matrix $\mathbf{W}(\mathbf{x}(t)) \in \mathbb{R}^{m\times l}$ composed of known displacement-dependent functions, and a vector $\mathbf{b} \in \mathbb{R}^{l}$ composed of the l unknown constant parameters. Typically, this relation can be constructed based on high-order polynomials of the position vector $\mathbf{x}(t)$. By sampling N data points during offline testing deformations, we create the following "long" matrix and vector:

$$\vec{\mathbf{W}} = \left[\, \mathbf{W}^{\top}(t_1) \quad \ldots \quad \mathbf{W}^{\top}(t_N)\,\right]^{\top} \in \mathbb{R}^{mN\times l} \tag{6}$$

$$\vec{\boldsymbol{\eta}} = \left[\, \boldsymbol{\eta}^{\top}(t_1) \quad \ldots \quad \boldsymbol{\eta}^{\top}(t_N)\,\right]^{\top} \in \mathbb{R}^{mN} \tag{7}$$

which are used to obtain an estimation of the parameters

$$\widehat{\mathbf{b}} = \left(\vec{\mathbf{W}}^{\top}\vec{\mathbf{W}}\right)^{-1} \vec{\mathbf{W}}^{\top}\vec{\boldsymbol{\eta}} \in \mathbb{R}^{l}. \tag{8}$$

See [27] for a solution to estimate $\mathbf{b}$ when the matrix $\vec{\mathbf{W}}^{\top}(t)\vec{\mathbf{W}}(t)$ is singular. For a rich-enough dataset, it is reasonable to assume that $\widehat{\mathbf{b}} \approx \mathbf{b}$. Then, an estimation of the deformation Jacobian matrix is computed as $\widehat{\mathbf{A}}(t) = \frac{\partial}{\partial \mathbf{x}}\mathbf{W}(\mathbf{x}(t))\widehat{\mathbf{b}}$.

2) Broyden Update Rule: This method represents an online alternative to estimate $\mathbf{A}(\mathbf{x}(t))$. It is a common technique to control the end effector in visual servoing applications (see for example [16], [28], and [29]), but to the best of our knowledge, it has not been widely adopted to estimate the deformations of elastic bodies. In our approach, we exploit this iterative method to compute an estimation $\widehat{\mathbf{A}}(t)$ at every time instant t. This is realized by using the observations of deformations $\delta\boldsymbol{\eta}(t) = \boldsymbol{\eta}(t) - \boldsymbol{\eta}(t-\delta t)$ and controls $\delta\mathbf{x}(t) = \mathbf{x}(t) - \mathbf{x}(t-\delta t)$, where δt represents the sampling rate. On its standard (discrete-time) formulation, the recursive update rule is given as follows:

$$\widehat{\mathbf{A}}(t) = \widehat{\mathbf{A}}(t-\delta t) + \Gamma \frac{\delta\boldsymbol{\eta}(t) - \widehat{\mathbf{A}}(t-\delta t)\delta\mathbf{x}(t)}{\delta\mathbf{x}^{\top}(t)\delta\mathbf{x}(t)}\delta\mathbf{x}^{\top}(t) \qquad \forall \delta\mathbf{x} \neq \mathbf{0}_{m\times 1} \tag{9}$$

for $0 < \Gamma \leq 1 \in \mathbb{R}$ as a gain to adjust the responsiveness of the estimation. With a value of $\Gamma = 1$, by right multiplying (9) by $\delta\mathbf{x}(t)$ we can see that $\dot{\boldsymbol{\eta}}(t) \approx \widehat{\mathbf{A}}(t)\dot{\mathbf{x}}(t)$ is satisfied for "small enough" sample time δt. However, an estimation with $\Gamma \approx 1$ is susceptible to noisy and rapidly changing observations $\delta\boldsymbol{\eta}(t)$ and $\delta\mathbf{x}(t)$.

For *slow* deformations and the motion of the manipulator, the matrix $\mathbf{A}(\mathbf{x}(t))$ is expected to change slowly. In this situation, we can use "smaller" values of Γ to make the iterative estimation less responsive, i.e., $\widehat{\mathbf{A}}(t) \approx \widehat{\mathbf{A}}(t-\delta t)$. This feature is useful for slow tasks since it can filter out noise in the sensor measurements (which is common with vision systems). Note that this iterative method does not require assumptions about the unknown analytical deformation model; however, it will not identify any parameters of the system. It will only estimate a matrix $\widehat{\mathbf{A}}(t)$ which *approximates* the deformation flow $\dot{\boldsymbol{\eta}}(t)$ at the time instant t.

B. Passivity-Based Controller

Let us assume for now that $\widehat{\mathbf{A}}(t)$ is exactly estimated so that $\mathbf{A}(\mathbf{x}(t)) \equiv \widehat{\mathbf{A}}(t)$. In order to control the behavior of system (4), we propose the following *energy-motivated* dynamic-state feedback control law:

$$\mathbf{u}(t) = \widehat{\mathbf{A}}^{-1}(t)\mathbf{p}(t) \tag{10}$$

where $\mathbf{p}(t) \in \mathbb{R}^{m}$ denotes a numerical state defined as

$$\dot{\mathbf{p}}(t) = -\frac{\partial \mathcal{U}}{\partial \boldsymbol{\eta}}(\Delta\boldsymbol{\eta}(t))^{\top} - \mathbf{C}\mathbf{p}(t) \tag{11}$$

with $\mathcal{U}(\Delta\boldsymbol{\eta}(t)) \geq 0 \in \mathbb{R}$ as a potential energy function, designed with a unique equilibrium at $\Delta\boldsymbol{\eta}(t) = \boldsymbol{\eta}(t) - \boldsymbol{\eta}^{*} = \mathbf{0}_{m\times 1}$, for $\boldsymbol{\eta}^{*} \in \mathbb{R}^{m}$ as a constant reference. The symmetric and positive matrix $\mathbf{C} = \mathbf{C}^{\top} > 0 \in \mathbb{R}^{m\times m}$ represents a damping matrix. Substitution of (10) into (4) enforces the following closed-loop dynamical system:

$$\begin{bmatrix} \dot{\boldsymbol{\eta}}(t) \\ \dot{\mathbf{p}}(t) \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{m\times m} & \mathbf{I}_{m\times m} \\ -\mathbf{I}_{m\times m} & -\mathbf{C} \end{bmatrix} \begin{bmatrix} \dfrac{\partial \mathcal{H}}{\partial \boldsymbol{\eta}}(t)^{\top} \\ \dfrac{\partial \mathcal{H}}{\partial \mathbf{p}}(t)^{\top} \end{bmatrix} \tag{12}$$

where the scalar functional $\mathcal{H}(\Delta\boldsymbol{\eta}(t), \mathbf{p}(t)) \in \mathbb{R}$ is defined as

$$\mathcal{H}(\Delta\boldsymbol{\eta}(t), \mathbf{p}(t)) = \mathcal{U}(\Delta\boldsymbol{\eta}(t)) + \frac{1}{2}\mathbf{p}(t)\cdot\mathbf{p}(t). \tag{13}$$

Note that we have recasted the deformation problem into a nonconservative Hamiltonian system [22], [30] for which powerful tools can be employed to design and analyze its behavior. For this system, $\mathcal{H}(\Delta\boldsymbol{\eta}(t), \mathbf{p}(t))$ clearly qualifies as a Lyapunov function since

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{H}(\Delta\boldsymbol{\eta}(t), \mathbf{p}(t)) = -\mathbf{p}^{\top}(t)\mathbf{C}\mathbf{p}(t) \leq 0. \tag{14}$$

This simply means that the energy function, and consequently the deformation error, is nonincreasing along the trajectories of the closed-loop dynamics (12) (as with the standard passivity-based approach [31]–[33]). Convergence to the desired reference $\boldsymbol{\eta}^{*}$ is easily proved using the Krasovskii–LaSalle principle [34].

Remark 2: The motion command $\mathbf{p}(t)$ provides smooth trajectories to the robot manipulator. To see this, assume a large enough damping matrix such that $\mathbf{C} \gg \frac{\partial^2 \mathcal{U}}{\partial \boldsymbol{\eta}^2}(\Delta\boldsymbol{\eta}(t))$, and consider $\boldsymbol{v}(t) = -\mathbf{C}^{-1}\frac{\partial \mathcal{U}}{\partial \boldsymbol{\eta}}(\Delta\boldsymbol{\eta}(t))$ as a desired static-state feedback velocity command. From the following equivalent expression for $\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{p}(t)$:

$$\dot{\mathbf{p}}(t) = -\mathbf{C}\left(\mathbf{p}(t) - \boldsymbol{v}(t)\right) \tag{15}$$

we can see that $\mathbf{p}(t)$ provides a smoothened version of $\boldsymbol{v}(t)$. This is particularly useful for velocity controllers, since a rapidly changing or even discontinuous error $\Delta\boldsymbol{\eta}(t)$ might generate a high velocity command. This also helps to filter out noise from measurement of the visual feedback $\mathbf{s}(t)$.

Remark 3: The magnitude of the matrix $\mathbf{C}$ provides a way to control the dissipative properties of $\mathbf{p}(t)$, which roughly speaking, regulates the deformation speed. That is, "large" values on $\mathbf{C}$ make the convergence of $\Delta\boldsymbol{\eta}(t)$ slower.

C. Robust Passivity-Based Design

In real applications, the estimation of $\mathbf{A}(\mathbf{x}(t))$ might be close but not exact, while the measurement of the visual feedback $\mathbf{s}(t)$ usually contains noise. In this scenario, it is not clear what the stability properties of the closed-loop system will be. To this end, let us assume that the deformation flow satisfies the following relation:

$$\dot{\boldsymbol{\eta}}(t) = \widehat{\mathbf{A}}(t)\dot{\mathbf{x}}(t) + \mathbf{d}(t) \tag{16}$$

where $\mathbf{d}(t) \in \mathbb{R}^m$ is an unknown *time-varying* disturbance vector, modeling noise and uncertainty in the estimation.

Assumption 2: For slow motion of the manipulator, the time-varying disturbance is: 1) bounded $\|\mathbf{d}(t)\| < d_{\max}$ by a positive constant $d_{\max} > 0 \in \mathbb{R}$, and 2) $\mathbf{d}(t) \to \mathbf{0}_{m\times 1}$ as $\dot{\mathbf{x}}(t) \to \mathbf{0}_{m\times 1}$.

Note that 2) simply means that for a static manipulation system, i.e., for $\dot{\mathbf{x}}(t) = \mathbf{0}_{m\times 1}$, the *numerical* disturbance vector $\mathbf{d}(t)$ does not produce any deformation flow $\dot{\boldsymbol{\eta}}(t)$. To guarantee the stable execution of the deformation task in the presence of this unknown disturbance vector, we now use a robust passivity-based design, that is, we include an additional "damping" term to enforce dissipativity with respect to $\mathbf{d}(t)$ [35]. To see this, consider the following dynamic-state feedback velocity controller:

$$\mathbf{u}(t) = \widehat{\mathbf{A}}^{-1}(t)\left(\mathbf{p}(t) - c\frac{\partial\mathcal{U}}{\partial\boldsymbol{\eta}}(\Delta\boldsymbol{\eta}(t))^{\top}\right) \tag{17}$$

for $c > 0 \in \mathbb{R}^+$ as a positive damping-like feedback gain. Substitution of (17), with $\dot{\mathbf{p}}(t)$ as in (11), into (16) yields the closed-loop dynamical system

$$\begin{bmatrix} \dot{\boldsymbol{\eta}}(t) \\ \dot{\mathbf{p}}(t) \end{bmatrix} = \begin{bmatrix} -c\mathbf{I}_{m\times m} & \mathbf{I}_{m\times m} \\ -\mathbf{I}_{m\times m} & -\mathbf{C} \end{bmatrix} \begin{bmatrix} \dfrac{\partial\mathcal{H}}{\partial\boldsymbol{\eta}}(t)^{\top} \\ \dfrac{\partial\mathcal{H}}{\partial\mathbf{p}}(t)^{\top} \end{bmatrix} + \begin{bmatrix} \mathbf{d}(t) \\ \mathbf{0}_{m\times 1} \end{bmatrix}. \tag{18}$$

To analyze the stability of this perturbed system, we make use of the formalism of ISS, that is, full dissipativity with respect to external disturbances (see [23] for a comprehensive survey). For the sake of completeness, we briefly recall the definition of ISS.

Definition 1 ([22]): A dynamical system

$$\dot{\boldsymbol{\chi}}(t) = \mathbf{f}(\boldsymbol{\chi}(t), \boldsymbol{\tau}(t)) \tag{19}$$

with state variable $\boldsymbol{\chi}(t) \in \mathbb{R}^{h_1}$ and disturbance $\boldsymbol{\tau}(t) \in \mathbb{R}^{h_2}$ (for $h_1 \geq h_2$), is endowed with ISS iff there is a smooth storage function $\mathcal{Q}(\boldsymbol{\chi}(t)) : \mathbb{R}^{h_1} \mapsto \mathbb{R}$ whose time derivative $\dot{\mathcal{Q}}(\boldsymbol{\chi}(t), \boldsymbol{\tau}(t)) : \mathbb{R}^{h_1} \times \mathbb{R}^{h_2} \mapsto \mathbb{R}$ satisfies

$$\dot{\mathcal{Q}}(\boldsymbol{\chi}(t), \boldsymbol{\tau}(t)) \leq -\alpha(|\boldsymbol{\chi}(t)|) + \gamma(|\boldsymbol{\tau}(t)|), \forall \boldsymbol{\chi}(t), \boldsymbol{\tau}(t) \tag{20}$$

where $\alpha(\cdot)$ and $\gamma(\cdot)$ are scalar K_∞ functions, i.e., are continuous, strictly increasing, unbounded, and satisfy $\alpha(\mathbf{0}_{h_1\times 1}) = 0$, $\gamma(\mathbf{0}_{h_2\times 1}) = 0$.

In the following proposition, we present the main stability result of our dynamic-state feedback deformation control law.

Proposition 1: The perturbed dynamical system (18) is endowed with ISS.

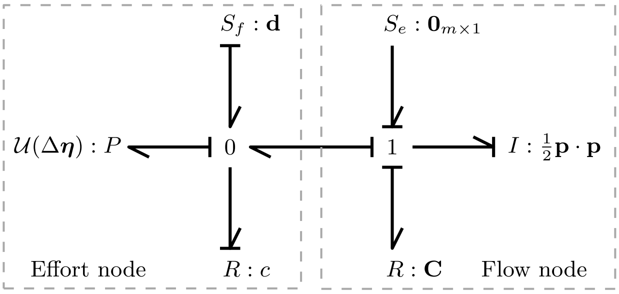

$S_f : \mathbf{d}$
$S_e : \mathbf{0}_{m\times 1}$
$\mathcal{U}(\Delta\boldsymbol{\eta}) : P$
0
1
$I : \frac{1}{2}\mathbf{p}\cdot\mathbf{p}$
Effort node
$R : c$
$R : \mathbf{C}$
Flow node

Fig. 3. Bond graph representation of the system (18). The elements P and I, respectively, represent a potential and inertial energy accumulators, R represents a resistor (dissipator), and S_e and S_f, respectively, represent an effort and flow sources. Elements connected to the 1 junction share the same flow, whereas those connected to the 0 junction share the same effort.

Proof: The time derivative of $\mathcal{H}(\Delta\boldsymbol{\eta}(t), \mathbf{p}(t))$ satisfies

$$\begin{aligned} \dot{\mathcal{H}}(t) &= -c\left\|\frac{\partial\mathcal{U}}{\partial\boldsymbol{\eta}}(\Delta\boldsymbol{\eta}(t))\right\|^2 - \mathbf{p}^{\top}(t)\mathbf{C}\mathbf{p}(t) \\ &\quad + \frac{\partial\mathcal{U}}{\partial\boldsymbol{\eta}}(\Delta\boldsymbol{\eta}(t)) \cdot \mathbf{d}(t) \\ &\leq -(c-\varepsilon)\left\|\frac{\partial\mathcal{U}}{\partial\boldsymbol{\eta}}(\Delta\boldsymbol{\eta}(t))\right\|^2 - \lambda\|\mathbf{p}(t)\|^2 + \frac{1}{\varepsilon}\|\mathbf{d}(t)\|^2 \end{aligned}$$

where the bottom expression is obtained with the ε-scaled Young's inequality, with the scalar $\lambda > 0 \in \mathbb{R}$ representing the minimum eigenvalue of $\mathbf{C}$. We see that for a feedback gain $c > \varepsilon$, where $\varepsilon \in \mathbb{R}$ is an *arbitrary* scalar, the closed-loop system (18) is endowed with ISS. ■

Remark 4: The feedback gains c and $\mathbf{C}$, respectively, determine the dissipative properties of the flow $\dot{\boldsymbol{\eta}}(t)$ and the "effort" $\dot{\mathbf{p}}(t)$ in (18) (see [36] for a detailed description of how to modify the dissipation structure on port-Hamiltonian systems). From Proposition 1, we see that only by including a damping term at the $\dot{\boldsymbol{\eta}}(t)$ level, the system is fully dissipative at both coordinates. Note that any $c > 0$ (recall that ε is an *arbitrary* scalar) suffices to enforce ISS in the presence of $\mathbf{d}(t)$. In this way, a conservative design for smooth velocity control is to rely on a highly damped dynamic-state feedback control $\mathbf{p}(t)$ (i.e., using a large matrix $\mathbf{C}$) to perform the deformation, whereas a "small" static-state feedback action $c\frac{\partial\mathcal{U}}{\partial\boldsymbol{\eta}}(\Delta\boldsymbol{\eta}(t))^{\top}$ is only needed to preserve ISS. A traditional visual servoing design, e.g., [14], only contains the latter control action; therefore, it is more susceptible to noisy measurements and (in comparison to our method) can only achieve limited types of dynamical response.

Remark 5: Fig. 3 presents a graphical representation (in bond graph notation [37], [38]) of the closed-loop system (18). From this figure, we see that the action induced by c can be interpreted as an effort-controlled dissipator, which contrast with the more common mechanical flow-controlled dissipator [39] [in (18), the latter one corresponds to $\mathbf{C}$]. In this figure, the left subsystem corresponds to the first row of (18), while the right subsystem corresponds to the second row. This simple mechanical analogy is useful to design the desired dynamical response to be performed by the closed-loop deformation control system.

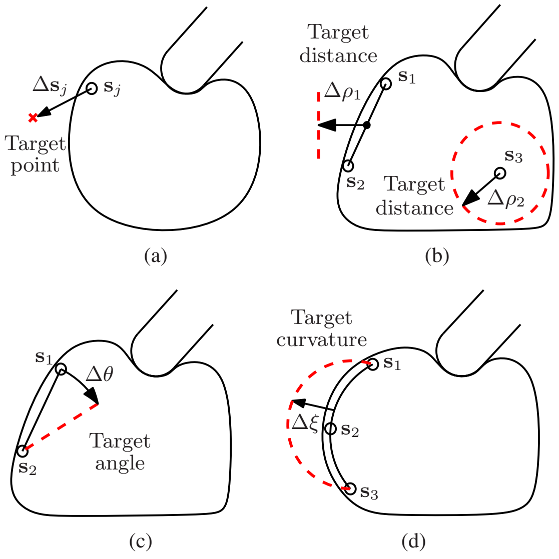


Fig. 4. Conceptual representation of different vision-based deformation features, with (a) vector of point displacement error $\Delta \mathbf{s}_j(t) = \mathbf{s}_j(t) - \mathbf{s}_j^*$, (b) scalar distance errors $\Delta\rho_j(t) = \rho_j(t) - \rho_j^*$, (c) scalar angle error $\Delta\theta(t) = \theta(t) - \theta^*$, and (d) scalar curvature error $\Delta\xi(t) = \xi(t) - \xi^*$. (a) Point-based deformation. (b) Distance-based deformation. (c) Angle-based deformation. (d) Curvature-based deformation.

IV. Case of Study

In this section, we present explicit definitions of the vision-based construction of sample deformation features.

1) *Point-based deformation:* The most basic deformation feature is the explicit displacement regulation of the visual coordinates $\mathbf{s}_j(t)$ of a point of interest. That is, two DOFs of the deformation feature vector are defined as

$$[\,\eta_i(t) \quad \eta_{i+1}(t)\,]^\top = [\,\mu_j(t) \quad \nu_j(t)\,]^\top. \tag{21}$$

Fig. 4(a) conceptually depicts this situation. Note that this type of deformation feature is the only one addressed in [9], [10], and [11].

2) *Distance-based deformation:* This is an extension of the previous case to the more general situation of controlling scalar distances $\rho_j(t) \in \mathbb{R}$ (with respect to an arbitrary reference) of features such as centroids, midpoints, absolute positions, etc. In Fig. 4(b), the distance coordinates $\rho_1(t), \rho_2(t) \in \mathbb{R}$ are computed as

$$\begin{aligned} \rho_1(t) &= \frac{1}{2}(\mu_1(t) + \mu_2(t)) \in \mathbb{R} \\ \rho_2(t) &= \|\mathbf{s}_3(t) - \boldsymbol{e}\| \in \mathbb{R} \end{aligned} \tag{22}$$

for $\boldsymbol{e} \in \mathbb{R}^2$ as an arbitrary bias vector. These coordinates simply represent the horizontal displacement of a line's midpoint [computed with $\mathbf{s}_1(t)$] and $\mathbf{s}_2(t)$ and the magnitude of a visual point $\mathbf{s}_3(t)$. In this scenario, for each distance $\rho_j(t)$, one DOF of the deformation feature vector is defined as

$$\eta_i(t) = \rho_j(t). \tag{23}$$

3) *Angle-based deformation:* Another useful deformation feature is the relative angle of a given line of interest [see Fig. 4(c)]. In our study, we define this feature by using two visual points $\mathbf{s}_1(t)$ and $\mathbf{s}_2(t)$ to describe a unit vector $\mathbf{l}(t) = \frac{\mathbf{s}_1(t) - \mathbf{s}_2(t)}{\|\mathbf{s}_1(t) - \mathbf{s}_2(t)\|} \in \mathbb{R}^2$. Then, for an arbitrary reference unit vector $\mathbf{i} \in \mathbb{R}^2$, we can compute the line's relative angle as

$$\theta_j(t) = \arccos\{\mathbf{l}(t) \cdot \mathbf{i}\} \in \mathbb{R}. \tag{24}$$

For each angle, one DOF of $\boldsymbol{\eta}(t)$ is defined as

$$\eta_i(t) = \theta_j(t). \tag{25}$$

4) *Curvature-based deformation:* A useful deformation feature is the measurement of the curvature (or roundness) of a contour of interest (e.g., one of the sides of the body). In our study, we *approximate* this value by using three visual feedback points $\mathbf{s}_1(t)$, $\mathbf{s}_2(t)$, and $\mathbf{s}_3(t)$ located on (or very close to) the contour of interest [see Fig. 4(d)]. If we assume that these points lie over the circumference of a circle, using standard geometric methods we can compute its *circumcentre* $\mathbf{o}(t) \in \mathbb{R}^2$ as follows [40]:

$$\mathbf{o}(t) = \frac{1}{2\varsigma(t)} \begin{bmatrix} \nu_2 - \nu_3 & \nu_3 - \nu_1 & \nu_1 - \nu_2 \\ \mu_3 - \mu_2 & \mu_1 - \mu_3 & \mu_2 - \mu_1 \end{bmatrix}(t) \times [\,\|\mathbf{s}_1(t)\|^2 \quad \|\mathbf{s}_2(t)\|^2 \quad \|\mathbf{s}_3(t)\|^2\,]^\top \tag{26}$$

for $\varsigma(t) = \mu_1(\nu_2 - \nu_3) + \mu_2(\nu_3 - \nu_1) + \mu_3(\nu_1 - \nu_2) \in \mathbb{R}$. Then, we can compute a quasi-curvature metric as follows

$$\xi_j(t) = \frac{1}{2}\frac{\|\mathbf{s}_1(t) - \mathbf{s}_3(t)\|}{\|\mathbf{s}_i(t) - \mathbf{o}(t)\|} \in \mathbb{R}. \tag{27}$$

For this unitless metric, a value $\xi_j(t) \approx 0$ describes flatness, while $\xi_j(t) \approx 1$ approximates a circle with radius $\frac{1}{2}\|\mathbf{s}_1(t) - \mathbf{s}_3(t)\|$. For each curve, one DOF of $\boldsymbol{\eta}(t)$ is defined as

$$\eta_i(t) = \xi_j(t). \tag{28}$$

V. Experimental Validation

A. Setup

The mechanical systems used for our experimental study are a six-DOF TX-60 Stäubli robot manipulator and a one-DOF rotational robot (see Figs. 5 and 14). The Stäubli manipulator is an open architecture robot with a PC-based controller that runs the real-time operative system VxWorks. To program the motion of its joints, the system has the *low-level interface* [41] that allows us to explicitly set the angular velocity on each of the joints. To control the motion of the one-DOF robot, we use a Galil DMC-1886 motion control board. Visual feedback is acquired with a Logitech C210 camera. The OpenCV Lucas-Kanade algorithm [42] is implemented to track the visual feedback points during the experiments.

In order to provide a deterministic real-time (RT) behavior to the control algorithms (a key feature to guarantee a *constant* and small δt), we use a Xenomai-patched RT-Linux computer [43] to process all the sensor feedback signals and compute the velocity control laws. The desired velocity command of

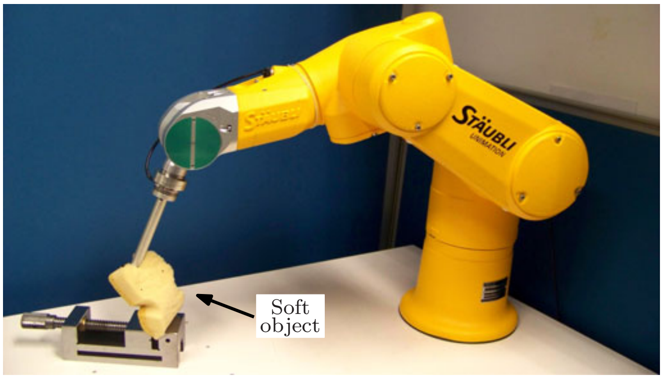


Fig. 5. Robot manipulator used for our experimental study.

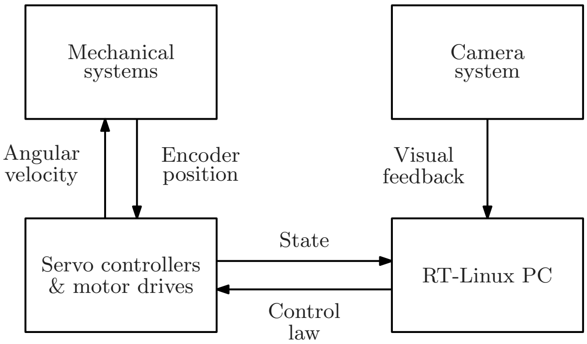


Fig. 6. Conceptual representation of the real-time control architecture for the velocity-controlled robot manipulators.

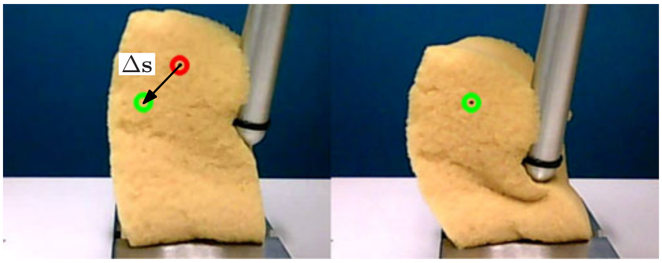


Fig. 7. Snapshots of the point-based deformation task used for the two-DOF comparative study.

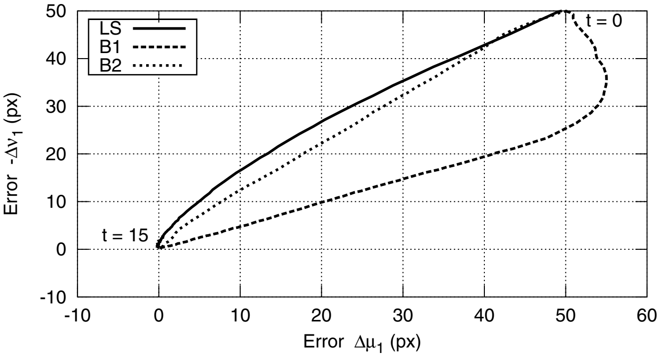


Fig. 8. Graphical comparison of the visual error trajectories of a point-based deformation, where the curve LS is obtained with the offline least-squares identification method, and the curves B1/B2 are, respectively, obtained with the online Broyden method without/with *a priori* knowledge of $\widehat{\mathbf{A}}(0)$.

the Stäubli robot is transmitted via TCP/IP to its low-level servo controller, whereas for the one-DOF mechanism we use standard PCI communication. Fig. 6 shows a conceptual representation of the control architecture. All the algorithms are implemented at a RT servo loop of 4 ms. In our experimental study, the robot manipulators physically interact with soft cleaning sponges.

To achieve the desired deformation tasks, we implement the following potential control action:

$$\frac{\partial \mathcal{U}}{\partial \boldsymbol{\eta}}(\Delta\boldsymbol{\eta}(t)) = \mathrm{sat}(3\Delta\boldsymbol{\eta}(t)) \tag{29}$$

where $-r \leq \mathrm{sat}(\cdot) \leq r$ represents a saturation function, for $r > 0 \in \mathbb{R}$. In our experiments, we use $r = 20$ pixel, $r = 0.1$ radian, and $r = 0.1$, for the point, angle, and curvature deformation features, respectively. We compute the dynamic-state feedback control action $\mathbf{p}(t)$ with a damping matrix of $\mathbf{C} = 5\mathbf{I}_{m\times m}$, and set the ISS feedback scalar to $c = 0.01$.

B. Two-Degree-of-Freedom Comparative Study

We perform this comparative study considering the control of two DOFs of the Stäubli robot (i.e., $m = 2$) positioning a single visual point, as conceptually depicted in Fig. 7. The deformation feature is defined as $[\eta_1(t), \eta_2(t)]^\top = [\mu_1(t), \nu_1(t)]^\top$.

In Section III, we describe two methods to estimate the deformation Jacobian matrix $\mathbf{A}(\mathbf{x}(t))$. We now compare the performance of our controller using both estimation methods for this simple point-based deformation task. To implement the offline estimation method, we approximate the deformation feature with the following polynomial relation:

$$\begin{aligned}\eta_1(t) &\approx b_0 + b_1x_1(t) + b_2x_1^2(t) + b_3x_1^3(t)\\ &\quad + b_4x_2(t) + b_5x_2^2(t)\\ \eta_2(t) &\approx b_6 + b_7x_1(t) + b_8x_1^2(t)\\ &\quad + b_9x_2(t) + b_{10}x_2^2(t) + b_{11}x_2^3(t)\end{aligned} \tag{30}$$

where the unknown deformation parameters vector is $\mathbf{b} = [b_0, \ldots, b_{11}]^\top \in \mathbb{R}^{12}$. To obtain the estimated vector $\widehat{\mathbf{b}} \in \mathbb{R}^{12}$, we perform offline (i.e., prior to the experiments) testing deformations around the region of interest while collecting the required data. After this, the vector $\widehat{\mathbf{b}} \in \mathbb{R}^{12}$ is computed as proposed in (8).

Fig. 8 shows the resulting error trajectories in the image plane, i.e., a curve in $(\Delta\mu_1, \Delta\nu_1)$ space. In this figure, the curve LS is obtained with the offline least-squares identification method described previously. The curve B1 is obtained with the online Broyden method when the estimated deformation Jacobian matrix $\widehat{\mathbf{A}}(0)$ is initialized with random numerical values (see also the results and discussions reported in [28]). The curve B2 is obtained with the Broyden method when *a priori* knowledge is employed by its estimator. This condition simply means that a "good-enough" initial value is required by the estimator, *only* at $t = 0$. The initial value $\widehat{\mathbf{A}}(0)$ can be obtained from a previous experiment, or from testing deformations around the starting configuration. Note that in contrast to the offline identification method, these testing deformations do not need to be performed

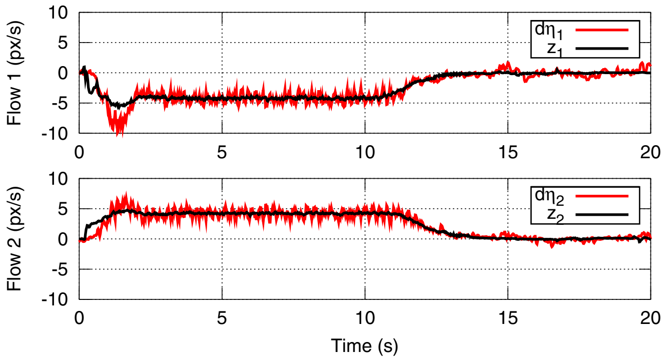


Fig. 9. Comparison of the measured visual flow $\dot{\boldsymbol{\eta}}(t)$ and the flow estimated by $\mathbf{z}(t) = \widehat{\mathbf{A}}(t)\dot{\mathbf{x}}(t)$, for the two-DOF point-based deformation task.

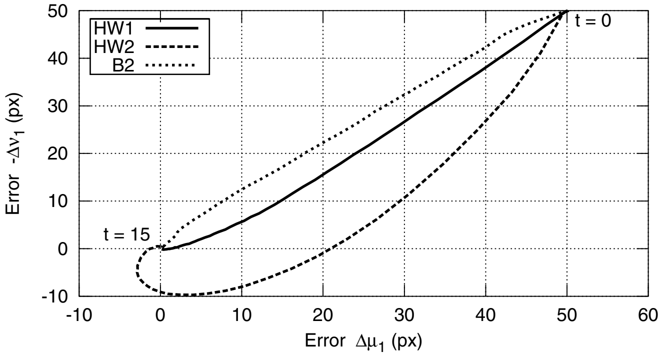


Fig. 10. Comparison of the visual error trajectories, where the curve B2 is obtained with our model-free control law, and the curves HW1 and HW2 are, respectively, obtained with [9] using a calibrated and an uncertain camera model.

all around the desired range of deformation (such that a rich data set is collected), but just locally around the starting point.

To provide a smooth motion command $\mathbf{u}(t)$ to the manipulator, we implement Broyden's iterative estimator with an update gain of $\Gamma = 0.01$, and remove noise from the signals $\delta\mathbf{x}(t)$ and $\delta\boldsymbol{\eta}(t)$ with a first-order low-pass filter. For this two-DOF point-based experiment, in Fig. 9 we present a graphical comparison of the measured visual flow $\dot{\boldsymbol{\eta}}(t)$ and the flow estimated by

$$\mathbf{z}(t) = \widehat{\mathbf{A}}(t)\dot{\mathbf{x}}(t) \in \mathbb{R}^2 \tag{31}$$

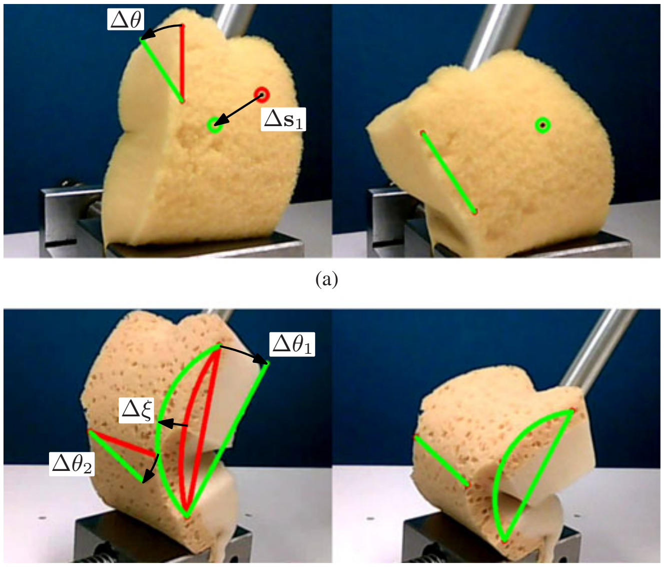
where $\widehat{\mathbf{A}}(t)$ is computed with Broyden's iterative estimator. This figure shows that the model-free online method provides a close approximation of the unknown deformation Jacobian matrix $\mathbf{A}(\mathbf{x}(t))$, when a good-enough initial value is available and the manipulator moves slowly. In the Appendix, we provide the detailed algorithm of our model-free control method. This algorithm includes a simple procedure to initialize the deformation Jacobian matrix $\widehat{\mathbf{A}}(0)$.

We now compare the performance (and distinctive features) of our deformation control law (17) with the control method proposed in [9]. Note that the controller in [9] is only formulated for plane motion of the controllable manipulators, whereas our method can cope with other deformation features and 3-D deformations (to be presented shortly). We implement the iterative Hirai–Wada method with a 2 × 2 lattice model and approximated spring constants of 0.5 (see [9] for details of this controller).

Fig. 10 shows the resulting error trajectories in the image plane. The Hirai–Wada method is robust to identification errors of the deformation model; however, in contrast to our formulation, the method in [9] does not contain an online estimation component of the camera's properties. This lack of adaptation is clearly seen in Fig. 10, where the curves HW1 and HW2 are, respectively, obtained with [9] with a calibrated and an uncertain camera model (we obtain the latter situation by simply rotating a calibrated camera by around 30°).

C. Single-Manipulator Three-Degree-of-Freedom Deformation Control

We now present our results with a 3-D deformation control of soft objects. These experiments are obtained with the Stäubli robot by controlling the three components of the Cartesian position vector $\mathbf{x}(t) \in \mathbb{R}^3$. In our setup, the deformation produced by reorienting the end effector is minimal; therefore, it is not considered for our control design. For this single-manipulator study, we have a 3-DOF deformation feature vector $\boldsymbol{\eta}(t) \in \mathbb{R}^3$ (i.e., $m = 3$).



(a)

(b)

Fig. 11. Snapshots of the single-manipulator 3-DOF deformation experiments, where the deformation feature vector is constructed as (a) $\boldsymbol{\eta}(t) = [\mu_1(t), \nu_1(t), \theta(t)]^\top$ and (b) $\boldsymbol{\eta}(t) = [\xi(t), \theta_1(t), \theta_2(t)]^\top$. (a) Point and angle deformation features. (b) Curvature and angle deformation features.

Fig. 11 shows snapshots of two deformation experiments performed with the manipulator. For the experiment shown in Fig. 11(a), we construct the deformation feature vector with three visual feedback points and define it as $\boldsymbol{\eta}(t) = [\mu_1(t), \nu_1(t), \theta(t)]^\top$. The first two coordinates represent the visual displacement of $\mathbf{s}_1(t)$, whereas the third one represents an angle. For the experiment shown in Fig. 11(b), we construct the deformation feature vector with five visual feedback points and define it as $\boldsymbol{\eta}(t) = [\xi(t), \theta_1(t), \theta_2(t)]^\top$. The first coordinate represents the proposed quasi-curvature metric, whereas the last

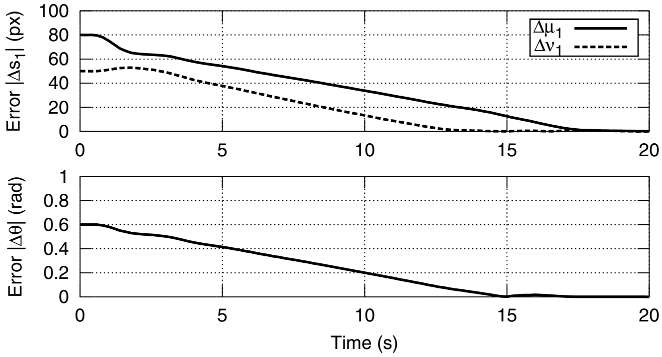


Fig. 12. Magnitude of the deformation error coordinates of the experiment shown in Fig. 11(a).

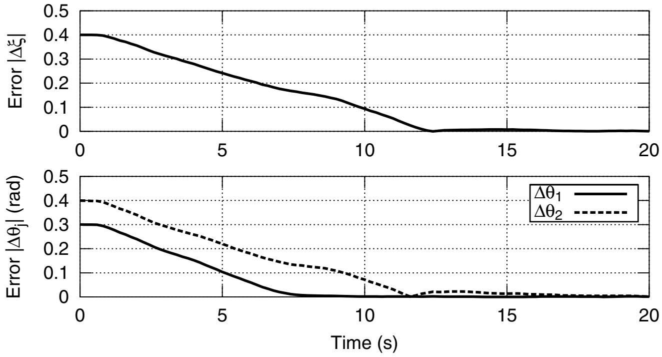


Fig. 13. Magnitude of the deformation error coordinates of the experiment shown in Fig. 11(b).

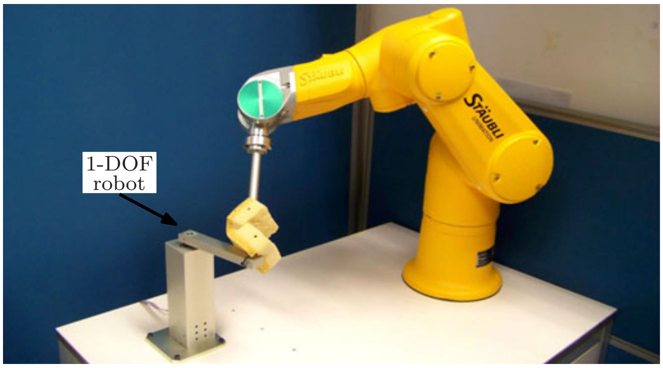


Fig. 14. Setup of the two-manipulator deformation experiments.

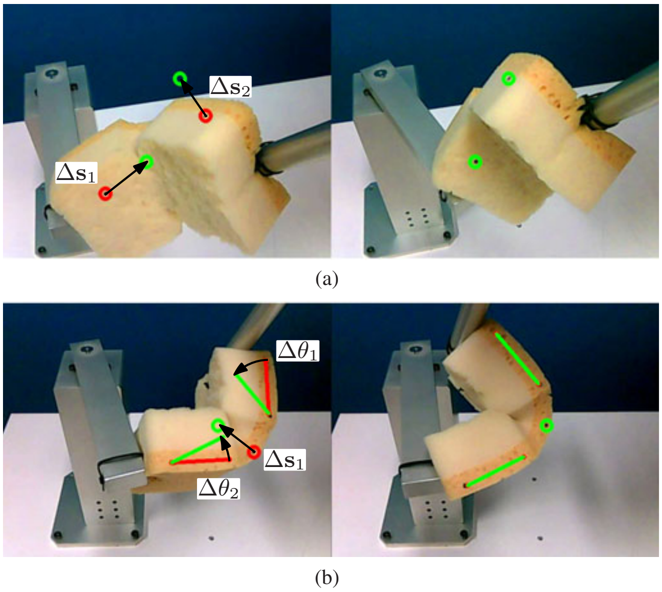


Fig. 15. Snapshots of the two-manipulator 4-DOF deformation experiments, where the deformation feature vector is constructed as (a) $\boldsymbol{\eta}(t) = [\mu_1(t), \nu_1(t), \mu_2(t), \nu_2(t)]^\top$ and (b) $\boldsymbol{\eta}(t) = [\mu_1(t), \nu_1(t), \theta_1(t), \theta_2(t)]^\top$. (a) Point deformation features. (b) Point and angle deformation features.

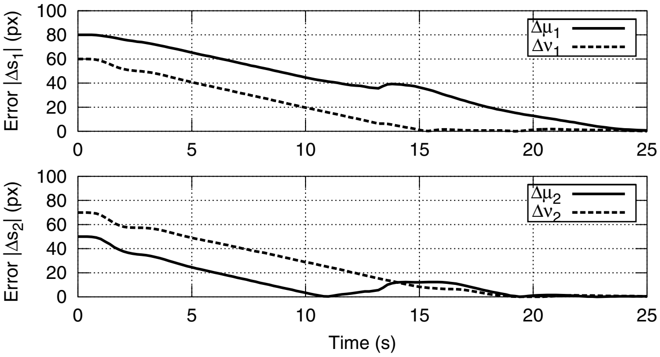


Fig. 16. Magnitude of the deformation error coordinates of the experiment shown in Fig. 15(a).

two represent angles. The deformation error trajectories of these experiments are, respectively, shown in Figs. 12–13.

D. Two-Manipulator Four-Degree-of-Freedom Deformation Control

We conduct this two-manipulator experimental study with the robotic setup shown in Fig. 14. In this study, we attach each end effector to one extreme of the soft object. Since the second manipulator only has one DOF, therefore, we extend one dimension to the total position vector $\mathbf{x}(t) \in \mathbb{R}^4$. For this two-manipulator 4-DOF experimental study, we define $\mathbf{x}(t)$ as follows

$$\mathbf{x}(t) = [\,\mathbf{x}_s^\top(t) \quad x_r(t)\,]^\top \tag{32}$$

where $\mathbf{x}_s(t) \in \mathbb{R}^3$ and $x_r(t) \in \mathbb{R}$, respectively, represent the Cartesian position of the Stäubli robot and the angular displacement of the 1-DOF rotational mechanism.

Fig. 15 shows snapshots of two deformation experiments coordinately performed by the manipulators. For the experiment shown in Fig. 15(a), we construct the deformation feature vector as $\boldsymbol{\eta}(t) = [\mu_1(t), \nu_1(t), \mu_2(t), \nu_2(t)]^\top$, where its coordinates simply represent the visual displacements of the points $\mathbf{s}_1(t)$ and $\mathbf{s}_2(t)$. For the experiment shown in Fig. 15(b), we construct the deformation feature vector with five visual feedback points and define it as $\boldsymbol{\eta}(t) = [\mu_1(t), \nu_1(t), \theta_1(t), \theta_2(t)]^\top$. The first two coordinates represent the displacements of $\mathbf{s}_1(t)$, whereas the last two represent angles of lines. The deformation error trajectories of these two experiments are, respectively, shown in Figs. 16 and 17.

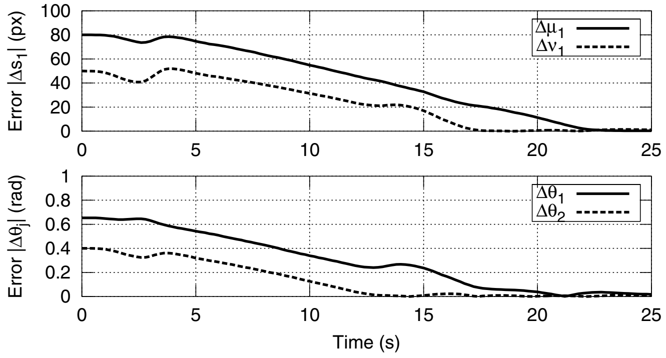


Fig. 17. Magnitude of the deformation error coordinates of the experiment shown in Fig. 15(b).

In the accompanying 18 MB video, we illustrate the performance of our model-free method with several deformation control experiments.

VI. Conclusion

In this paper, we have proposed a vision-based deformation controller that coordinates the motion of robot manipulators that deform an unknown purely elastic object. We first constructed (using multiple visual feedback points) a general feature vector to quantify the object's deformation. To avoid identifying the deformation and camera's model, we next used the Broyden update rule to obtain a numerical estimation of the (visual) deformation Jacobian matrix in real time. Finally, we transformed the deformation control problem into a nonconservative Hamiltonian dynamical system, for which we designed a novel energy-based dynamic-state feedback velocity control law.

It is important to remark that the online estimation method that we have presented requires slow motion of the robot manipulators, low-pass filtering of the observation signals, and a small gain Γ in order to provide smooth deformation trajectories. Note that this slow motion condition does not impose severe constraints to many real-life deformation tasks. For example, most robotised surgical procedures require slow motion of the mechanism for safety reasons. The experimental results that we have reported were conducted with no knowledge of the deformation or camera models. These results show the viability of our control method for uncalibrated/unmodeled real-time deformation tasks. Also, note that a critical requirement to successfully perform the task is to provide a vector $\boldsymbol{\eta}^*$ that can be physically reached with the manipulation system.

As future research, the implementation of a simplified version of this method to a real surgical application will be carried out. We are currently working on the development of a vision-based uterus deformer for a semiautomatic hysterectomy procedure. From a basic research perspective, we want to formulate the problem with multiple vision systems or endoscopic images (so as to more accurately control three-dimensional deformations) and with torque-controlled manipulators (to incorporate adaptive algorithms, e.g., [44] and [45]).

Appendix
Algorithm of Our Model-Free Control Method

```
if $\widehat{\mathbf{A}}(0)$ is unknown then
    Initialise $\widehat{\mathbf{A}}(0) \leftarrow$ random values
    repeat
        Command slow motion $\dot{\mathbf{x}}(t) \leftarrow \mathbf{v}_{slow}$
        Measure and filter observations $\delta\boldsymbol{\eta}(t)$ and $\delta\mathbf{x}(t)$
        Compute estimated Jacobian $\widehat{\mathbf{A}}(t) \leftarrow$ (9)
    until Flow error $\|\mathbf{z}(t) - \dot{\boldsymbol{\eta}}(t)\| < small\_error$
end if
repeat                                                    ▷ Control loop
    Measure feedback signals $\mathbf{s}(t)$ and $\mathbf{x}(t)$
    Compute numerical state $\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{p}(t) \leftarrow$ (11)
    Compute and filter observations $\delta\boldsymbol{\eta}(t)$, $\delta\mathbf{x}(t)$
    Compute estimated Jacobian $\widehat{\mathbf{A}}(t) \leftarrow$ (9)
    Compute velocity control law $\mathbf{u}(t) \leftarrow$ (17)
    Map $\mathbf{u}(t) \mapsto \boldsymbol{\omega}(t)$ and command joint motion
until Error $\|\Delta\boldsymbol{\eta}(t)\| < e_{min}$
```

We define the following $small\_error > 0 \in \mathbb{R}$, $\mathbf{v}_{\mathrm{slow}} \in \mathbb{R}^m$, and $e_{\min} > 0 \in \mathbb{R}$ as task-specific parameters. In our experimental study, we obtained good results for the initialization of $\widehat{\mathbf{A}}(0)$ when the velocity command $\mathbf{v}_{\mathrm{slow}}$ simultaneously moved *all* coordinates of the end-effector constrained displacements $\mathbf{x}(t)$. The positive scalar $small\_error$ determines when $\widehat{\mathbf{A}}(t)$ provides a close enough initial approximation of the deformation flow $\dot{\boldsymbol{\eta}}(t)$. The positive scalar $e_{\min}$ determines the minimum admissible deformation error.

References

[1] D. Henrich and H. Wörn Eds., *Robot Manipulation of Deformable Object* (Series Advanced manufacturing). New York, NY, USA: Springer-Verlag, 2000.

[2] M. Saha and P. Isto, "Manipulation planning for deformable linear objects," *IEEE Trans. Robot.*, vol. 23, no. 6, pp. 1141–1150, Dec. 2007.

[3] V. Mallapragada, N. Sarkar, and T. Podder, "Toward a robot-assisted breast intervention system," *IEEE/ASME Trans. Mechatronics*, vol. 16, no. 6, pp. 1011–1020, Dec. 2011.

[4] S. Tokumoto and S. Hirai, "Deformation control of rheological food dough using a forming process model," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2002, vol. 2, pp. 1457–1464.

[5] D. Sun, J. K. Mills, and Y. Liu, "Position control of robot manipulators manipulating a flexible payload," *Int. J. Robot. Res.*, vol. 18, no. 3, pp. 319–332, Mar. 1999.

[6] F. Ding, J. Huang, Y. Wang, T. Fukuda, and T. Matsuno, "Adaptive sliding mode control for manipulating deformable linear object with input saturation," in *Proc. Int. Conf. Mechatron. Autom.*, 2012, pp. 1862–1867.

[7] M. Higashimori, K. Yoshimoto, and M. Kaneko, "Active shaping of an unknown rheological object based on deformation decomposition into elasticity and plasticity," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2010, pp. 5120–5126.

[8] M. Shibata and S. Hirai, "Soft object manipulation by simultaneous control of motion and deformation," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2006, pp. 2460–2465.

[9] S. Hirai and T. Wada, "Indirect simultaneous positioning of deformable objects with multi-pinching fingers based on an uncertain model," *Robotica*, vol. 18, no. 1, pp. 3–11, Jan. 2000.

[10] T. Wada, S. Hirai, S. Kawamura, and N. Kamiji, "Robust manipulation of deformable objects by a simple PID feedback," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2001, vol. 1, pp. 85–90.

[11] J. Smolen and A. Patriciu, "Deformation planning for robotic soft tissue manipulation," in *Proc. Int. Conf. Adv. Comput.-Human Interact.*, 2009, pp. 199–204.

[12] J. Das and N. Sarkar, "Autonomous shape control of a deformable object by multiple manipulators," *J. Intell. Robot. Syst.*, vol. 62, pp. 3–27, Apr. 2011.

[13] M. Cusumano-Towner, A. Singh, S. Miller, J. O"Brien, and P. Abbeel, "Bringing clothing into desired configurations with limited perception," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2011, pp. 3893–3900.

[14] S. Hutchinson, G. Hager, and P. Corke, "A tutorial on visual servo control," *IEEE Trans. Robot. Autom.*, vol. 12, no. 5, pp. 651–670, Oct. 1996.

[15] F. Chaumette and S. Hutchinson, "Visual servo control—Part I: Basic approaches," *IEEE Robot. Autom. Mag.*, vol. 13, no. 4, pp. 82–90, Dec. 2006.

[16] F. Chaumette and S. Hutchinson, "Visual servo control—Part II: Advanced approaches," *IEEE Robot. Autom. Mag.*, vol. 14, no. 1, pp. 109–118, Mar. 2007.

[17] H. Wang, Y.-H. Liu, and D. Zhou, "Adaptive visual servoing using point and line features with an uncalibrated eye-in-hand camera," *IEEE Trans. Robot.*, vol. 24, no. 4, pp. 843–857, Aug. 2008.

[18] H. Wang, Y.-H. Liu, and W. Chen, "Uncalibrated visual tracking control without visual velocity," *IEEE Trans. Control Syst. Technol.*, vol. 18, no. 6, pp. 1359–1370, Nov. 2010.

[19] H. Wang, Y.-H. Liu, W. Chen, and Z. Wang, "A new approach to dynamic eye-in-hand visual tracking using nonlinear observers," *IEEE/ASME Trans. Mechatronics*, vol. 16, no. 2, pp. 387–394, Apr. 2011.

[20] Y. Shen, D. Sun, Y.-H. Liu, and K. Li, "Asymptotic trajectory tracking of manipulators using uncalibrated visual feedback," *IEEE/ASME Trans. Mechatronics*, vol. 8, no. 1, pp. 87–98, Mar. 2003.

[21] M. C. Cavusoglu, J. Rotella, W. S. Newman, S. Choi, J. Ustin, and S. S. Sastry, "Control algorithms for active relative motion cancelling for robotic assisted off-pump coronary artery bypass graft surgery," in *Proc. 12th Int. Conf. Adv. Robot.*, Jul. 2005, pp. 431–436.

[22] J. E. Marsden and T. Ratiu, *Introduction to Mechanics and Symmetr* (Texts in Applied Mathematics). New York, NY, USA: Springer-Verlag, 1994.

[23] E. D. Sontag, "Input to state stability: Basic concepts and results," in *Nonlinear and Optimal Control Theory*, P. Nistri and G. Stefani, Eds. New York, NY, USA: Springer-Verlag, 2006, pp. 163–220.

[24] D. Navarro-Alarcon, Y.-H. Liu, J. G. Romero, and P. Li, "Visually servoed deformation control by robot manipulators," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2013, vol. 1, pp. 5239–5244.

[25] J. J. Craig, *Introduction to Robotics: Mechanics and Control*, 2nd ed. Boston, MA, USA: Addison-Wesley, 1989.

[26] D. Whitney, "Resolved motion rate control of manipulators and human prostheses," *IEEE Trans. Man-Mach. Syst.*, vol. 10, no. 2, pp. 47–53, Jun. 1969.

[27] A. N. Tikhonov and V. Y. Arsenin, *Solutions of Ill Posed Problems*. New York, NY, USA: Winston and Sons, 1977.

[28] K. Hosoda and M. Asada, "Versatile visual servoing without knowledge of true Jacobian," in *Proc. IEEE Int. Conf. Intell. Robots Syst.*, 1994, vol. 1, pp. 186–193.

[29] M. Jagersand, O. Fuentes, and R. Nelson, "Experimental evaluation of uncalibrated visual servoing for precision manipulation," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1997, vol. 4, pp. 2874–2880.

[30] A. van der Schaft, *L_2-Gain and Passivity Techniques in Nonlinear Control*, 2nd ed. London, U.K.: Springer-Verlag, 2000.

[31] M. Takegaki and S. Arimoto, "A new feedback method for dynamic control of manipulators," *ASME J. Dyn. Sys., Meas., Control*, vol. 102, pp. 119–125, Jun. 1981.

[32] S. Arimoto, *Control Theory of Non-linear Mechanical Systems: A Passivity-based and Circuit-theoretic Approach*. New York, NY, USA: Oxford Univ. Press, 1996.

[33] R. Ortega, J. Loria Perez, P. Nicklasson, and H. Sira-Ramirez, *Passivity-based Control of Euler-Lagrange Systems*, 1st ed. New York, NY, USA: Springer-Verlag, 1998.

[34] M. Vidyasagar, *Nonlinear Systems Analysis*, 2nd ed. Englewood Cliffs, NJ, USA: Prentice–Hall, 1993.

[35] J. G. Romero, A. Donaire, and R. Ortega, "Robust energy shaping control of mechanical systems," *Syst. Control Lett.*, vol. 62, no. 9, pp. 770–780, Sep. 2013.

[36] R. Ortega, A. J. van der Schaft, B. Maschke, and G. Escobar, "Interconnection and damping assignment passivity-based control of port-controlled Hamiltonian systems," *Automatica*, vol. 38, no. 4, pp. 585–596, Apr. 2002.

[37] H. M. Paynter, *Analysis and Design of Engineering Systems*. Boston, MA, USA: MIT Press, 1961.

[38] G. Golo, A. S. van der, P. C. Breedveld, and B. M. Maschke, "Hamiltonian formulation of bond graphs," in *Nonlinear and Hybrid Systems in Automotive Control*, R. Johansson and A. Rantzer, Eds. New York, NY, USA: Springer-Verlag, 2003, pp. 351–372.

[39] D. Jeltsema and J. Scherpen, "Multidomain modeling of nonlinear networks and systems," *IEEE Control Syst. Mag.*, vol. 29, no. 4, pp. 28–59, Aug. 2009.

[40] C. Kimberling, *Triangle Centers and Central Triangles* (Series Congressus Numerantum). Winnipeg, MB, Canada: Utilitas Mathematica, 1998.

[41] F. Pertin and J.-M. Bonnet-des-Tuves, "Real time robot controller abstraction layer," in *Proc. Int. Symp. Robot.*, 2004, p. 71.

[42] G. Bradski, "The OpenCV library," *Dr. Dobb's J. Softw. Tools*, vol. 25, no. 11, pp. 122–125, Nov. 2000.

[43] P. Gerum, "Xenomai—Implementing a RTOS emulation framework on GNU/Linux," White Paper, Xenomai, Apr. 2004.

[44] J. Slotine and W. Li, "On the adaptive control of robot manipulators," *Int. J. Robot. Res.*, vol. 6, no. 3, pp. 49–59, Sep. 1987.

[45] Y.-H. Liu, H. Wang, C. Wang, and K. K. Lam, "Uncalibrated visual servoing of robots using a depth-independent interaction matrix," *IEEE Trans. Robot.*, vol. 22, no. 4, pp. 804–817, Aug. 2006.

David Navarro-Alarcón (S'06) was born in Mexico. He received the M.Sc. degree in robotics and advanced manufacturing from the Centre for Research and Advanced Studies, National Polytechnic Institute, Saltillo, Mexico, in 2009 and is currently working toward the Ph.D. degree in mechanical and automation engineering with the Chinese University of Hong Kong, Shatin, N.T., Hong Kong.

His research interests include force/vision-based control of mechanical systems and medical robotics.

Yun-Hui Liu (S'90–M'92–SM'98–F'09) received the B.Eng. degree in applied dynamics from the Beijing Institute of Technology, Beijing, China, in 1985; the M.Eng. degree in mechanical engineering from Osaka University, Osaka, Japan, in 1989; and the Ph.D. degree in mathematical engineering and information physics from the University of Tokyo, Tokyo, Japan, in 1992.

He was with the Electrotechnical Laboratory, Ministry of International Trade and Industry, Ibaraki, Japan, from 1992 to 1995. Since February 1995, he has been with the Chinese University of Hong Kong (CUHK), Shatin, N.T., Hong Kong, where he is currently a Professor with the Department of Mechanical and Automation Engineering. He is also a Professor with the State Key Lab of Robotics Technology and System, Harbin Institute of Technology, Harbin, China, and the Director of Joint Centre for Intelligent Sensing and Systems, National University of Defense Technology, Hunan, China, and CUHK. He has published more than 200 papers in refereed journals and refereed conference proceedings and was listed in the Highly Cited Authors (Engineering) by Thomson Reuters in 2013. His research interests include visual servoing, medical robotics, multifingered robot hands, mobile robots, sensor networks, and machine intelligence.

Dr. Liu has received numerous research awards from international journals and international conferences in robotics and automation and government agencies. He is the Editor-in-Chief of *Robotics and Biomimetics*, which is a new journal to be published by Springer in 2014, and an Editor of *Advanced Robotics*. He served as an Associate Editor of the IEEE TRANSACTION ON ROBOTICS AND AUTOMATION and General Chair of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems.

José Guadalupe Romero was born in Tlaxcala, Mexico, in 1983. He received the B.S. degree in electronic engineering from the University of Zacatecas, Zacatecas, Mexico, in 2006 and the M.Sc. degree in robotics and advanced manufacturing from the Centre for Research and Advanced Studies, National Polytechnic Institute, Saltillo, Mexico, in 2009. He is currently working toward the Ph.D. degree with the Laboratoire des Signaux et Systèmes, Université Paris-Sud 11, Gif-sur-Yvette, France.

His research interests include the fields of robotics and robust nonlinear control, focussing on mechanical systems.

Peng Li received the B.Eng. degree in mechanical engineering from North Eastern University, Shenyang, China, in 2004 and the Ph.D. degree in mechatronics from the Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, in 2010.

From 2009 to 2010, he was with the Chinese University of Hong Kong, Shatin N.T., Hong Kong, as a Research Assistant, where, since 2010, he has been a Postdoctoral Researcher with the Department of Mechanical and Automation Engineering. His research interests include developing surgical robots, medical devices, and pipeline robots.