Deformation control of a 3D soft object using RGB-D visual servoing and FEM-based dynamic model

Mandela Ouafo Fonkoua, François Chaumette, and Alexandre Krupa

***Abstract*—In this letter, we present a visual control framework for accurately positioning feature points belonging to the surface of a 3D deformable object to desired 3D positions, by acting on a set of manipulated points using a robotic manipulator. Notably, our framework considers the dynamic behavior of the object deformation, that is, we do not assume that the object is in its static equilibrium during the manipulation. By relying on a coarse dynamic Finite Element Model (FEM), we have successfully formulated the analytical relationship expressing the motion of the feature points to the six degrees of freedom (6 DOF) motion of a robot gripper. From this modeling step, a novel closed-loop deformation controller is designed. To be robust against model approximations, the whole shape of the object is tracked in real-time using an RGB-D camera, thus allowing to correct any drift between the object and its model on-the-fly. Our model-based and vision-based controller has been validated in real experiments. The results highlight the effectiveness of the proposed methodology.**

***Index Terms*—Deformation control, physics-based model, visual servoing.**

I. Introduction

THE manipulation of deformable objects has become a major concern in robotics, recently gaining substantial attention within the robotics research community. This growing interest is due to its important role in enhancing robot capabilities in various fields, such as surgical robotics, home and personal robotics, and manufacturing robotics where tasks involving the manipulation of soft and flexible materials are required [1]. Despite advancements in soft robotics, the autonomous manipulation of compliant objects persists as an ongoing research challenge [1]. In contrast to rigid objects, deformable objects exhibit infinite degrees of freedom and highly nonlinear dynamics, introducing significant challenges.

Accurately controlling the position of points belonging to a deformable object is one of the fundamental operations for its manipulation [2]. In this work, we focus on the indirect positioning of a 3D elastic object that consists in driving points of its surface (feature points) to desired 3D positions by acting on a set of manipulated points through a robot gripper.

Manuscript received: March 15, 2024; Accepted May 27, 2024.

This paper was recommended for publication by Editor Pascal Vasseur upon evaluation of the Associate Editor and Reviewers' comments.

This work was supported by the BIFROST (313870) project funded by The Research Council of Norway. Experiments presented in this paper were carried out thanks to a platform of the Robotex 2.0 French research infrastructure.

The authors are with Inria, Univ. Rennes, CNRS, IRISA, Campus de Beaulieu, 35042 Rennes, France (e-mail: mandela.ouafo-fonkoua@inria.fr; francois.chaumette@inria.fr; alexandre.krupa@inria.fr).

Digital Object Identifier (DOI): see top of this page.

To autonomously control the deformation of a soft object using a robot, it is necessary to establish a relationship between the robot motion and the change in the object deformation. In the literature, this relation is obtained either by using a physics-based or geometric-based deformation model, or by an online estimation of a local deformation model from past sensor measurements.

A. Model-free deformation control

Recent literature on deformation control of soft bodies commonly employs a deformation Jacobian [3], [4], [5]. This Jacobian establishes the relationship between the robot motion and the deformation of the object. It is typically numerically estimated based on sensor measurements. In model-free approaches, the control problem is generally formulated in sensor measurement space. Instead of using a deformation Jacobian, other model-free approaches use machine learning to directly map the sensor measurements to the control velocities of the robot end-effectors [6], [7], [8]. After obtaining the model, one can employ a controller such as model predictive control (MPC) [9] for executing the deformation task. These approaches do not require an identification of the physical parameters of the object being manipulated. However, the success of these approaches heavily depends on the accuracy of initial estimates for the interaction, and there are potential issues of losing generality or overtraining in specific situations, which present difficulties for learned models. Moreover, these model-free techniques are sensitive to measurement noise and lack the capability to guarantee beforehand the feasibility of the desired deformation. The risk of damaging the soft object, especially in critical scenarios like robot-assisted surgery, remains a major concern in such instances.

B. Model-based deformation control

To control the deformation of soft objects, methods relying on physics-based model have been proposed to predict the behavior of the objects of interest. Wada et al. [10], [11] introduced the problem of indirect simultaneous positioning for soft objects. They developed a control law using a PID controller and a Mass-Spring-Model (MSM). However, the validation of this approach has been limited to simulations involving small deformations and planar motion. In a more recent development based on MSM, Makiyeh et al. [12] introduced a deformation control strategy to indirectly position a single point on a soft

object by controlling only the translations of the robot end-effector. Kinio et al. [13] employed a H_∞ controller and a FEM model of the object to carry out the deformation control task. Ficuciello et al. [14] and Duenser et al. [15] also considered FEM to control the shape of a soft object by formulating the problem as an optimization problem. However their control approaches are open-loop, which reduces their robustness to model and parameter uncertainties. In a more recent development based on FEM, Koessler et al. [16] proposed a model-based controller with visual feedback to indirectly position points belonging to a soft linear object. However their approach cannot perform deformation at high speed since it relies on the quasi-static hypothesis which considers that the object is always in its equilibrium state during the deformation.

FEM-based approaches are also considered in soft robotics: [17] uses a reduced-order FEM to control a robotic arm, and [18] applies FEM-based gain-scheduling control for positioning the end-effector of a soft trunk robot. As described below, a different approach is proposed in this paper.

Other types of physics-based models are mesh-free: Fason et al. [19] used a mesh-free particle method, known as the reproducing kernel particle method, from which an output regulator is designed to move some points on the object to desired 2D locations, this approach being limited to planar motion. Geometric approaches are another type of model-based approach. In this category, Shetab-Bushehri et al. [20], [21] used a model called As-Rigid-As Possible (ARAP) to model deformable bodies and to automatically control their shapes. Geometric models only take into account the shape of the object and can not capture complex behaviors related to the object properties.

C. Contributions

In this letter, we propose a vision-based and model-based deformation control framework. Our main contributions are:

- A new analytical form of the relation between the motion of a robot gripper and the motion of feature points is derived based on FEM. Contrary to previous work using FEM, our developments are not based on quasi-static assumptions, which allows for better robustness against dynamic effects.
- From the relation obtained, a novel control law is designed, incorporating a feedback term and a feedforward term whose values are computed from the derived dynamic model.
- An experimental validation of the proposed control framework with two distinct soft objects made of foam.

The remainder of this letter is organized as follows: Section II outlines the proposed approach, Section III details the experimental evaluation, and finally Section IV concludes the letter with a discussion on the approach and on future work.

II. Methodology

The main challenge addressed in this letter is to generate the velocities of a robot gripper to guide 3D feature points on a 3D elastic object to desired locations in a closed-loop manner. This is achieved through the utilization of a dynamic physical model and a stationary RGB-D camera. We assume that the robot gripper remains rigidly attached to the object throughout the entire manipulation process, and we consider that the transformation matrix between the RGB-D camera frame $\{\mathcal{C}\}$ and the robot base frame $\{\mathcal{B}\}$ is known thanks to a preliminary hand-eye calibration. This is why all the following equations are expressed in the camera frame. As already said, contrary to previous approaches [16], [14], [15] we do not assume that the object is in a quasi-static equilibrium during its deformation. To make it clear, we describe in Section II-A1 the modeling obtained using this quasi-static assumption while the dynamic case is described in Section II-A2.

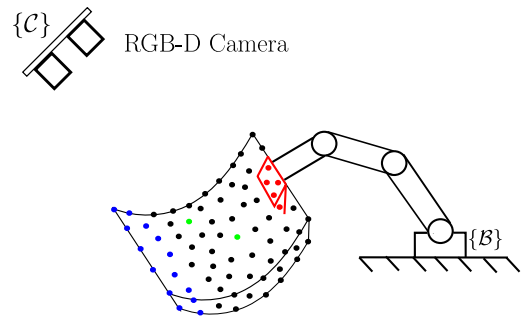

Fig. 1: System configuration. The points depicted on the object are nodes of the volumetric mesh of the 3D object. The red points are nodes attached to the gripper, the green points are the feature points, the blue points are static or free to move, depending on the chosen configuration, and the black points are the other points.

A. Modeling

A detailed introduction on how to model deformable objects using FEM can be found in [22], [23]. In continuum mechanics, 3D deformable objects are modelled by non-linear partial differential equations of elasticity. Applying FEM to the non-linear equations of motion results in the following simplified equation,

$$\boldsymbol{M}(\boldsymbol{x})\ddot{\boldsymbol{x}} = \boldsymbol{f}_e(t) - \boldsymbol{f}(\boldsymbol{x}, \dot{\boldsymbol{x}}) \tag{1}$$

where $\boldsymbol{x} \in \mathbb{R}^{3n}$ is the vector of generalized degrees of freedom that corresponds to the 3D positions of the nodes of a volumetric 3D mesh describing the object topology, $\dot{\boldsymbol{x}}$ the vector of velocities, $\ddot{\boldsymbol{x}}$ the vector of accelerations, $\boldsymbol{M}(\boldsymbol{x}) \in \mathbb{R}^{3n\times 3n}$ is the mesh mass matrix. $\boldsymbol{f}$ represents the internal elastic forces applied to the deformable object depending on the current state and $\boldsymbol{f}_e$ gathers external forces applied on the object. We split the external forces in three parts: the known external forces $\boldsymbol{f}_p$ (like gravity for example), the external forces due to known constraints $\boldsymbol{f}_c$ (like fixing a part of the object, for example), and external forces due to actuators $\boldsymbol{f}_a$ (which in our case is due to the displacement of the gripper attached to the object). We can thus rewrite (1) as follows,

$$\boldsymbol{M}(\boldsymbol{x})\ddot{\boldsymbol{x}} = \boldsymbol{f}_p(t) + \boldsymbol{f}_c(t) + \boldsymbol{f}_a(t) - \boldsymbol{f}(\boldsymbol{x}, \dot{\boldsymbol{x}}) \tag{2}$$

In order to be applicable for our control purposes, (2) needs to be integrated (numerically) over time, to find a relation

between the velocities of the manipulated points and the velocities of the feature points. For that, we employ a time-stepping implicit scheme, specifically the backward Euler method, to ensure unconditional stability of the numerical integration [22].

In the following sections, we unveil the resulting model for both quasi-static and dynamic cases.

1) ***Quasi-static case****:* In this case, (2) becomes,

$$\boldsymbol{f}_p(t) + \boldsymbol{f}_c(t) + \boldsymbol{f}_a(t) - \boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{0} \tag{3}$$

The internal elastic forces $\boldsymbol{f}(\boldsymbol{x})$ are non-linear. To address this, we use a Taylor series expansion to compute a unique linearization of $\boldsymbol{f}(\boldsymbol{x})$ per time step h. This yields:

$$\boldsymbol{f}(\boldsymbol{x}^{t+h}) = \boldsymbol{f}(\boldsymbol{x}^t) + \frac{\partial \boldsymbol{f}(\boldsymbol{x}^t)}{\partial \boldsymbol{x}} \Delta \boldsymbol{x} \tag{4}$$

with $\Delta \boldsymbol{x} = \boldsymbol{x}^{t+h} - \boldsymbol{x}^t$. Combining (3) and (4) leads to,

$$\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}} \Delta \boldsymbol{x} = \boldsymbol{f}_p^{t+h} + \boldsymbol{f}_c^{t+h} + \boldsymbol{f}_a^{t+h} - \boldsymbol{f}(\boldsymbol{x}^t) \tag{5}$$

$$= \Delta \boldsymbol{f}_p + \Delta \boldsymbol{f}_c + \Delta \boldsymbol{f}_a + \boldsymbol{f}_p^t + \boldsymbol{f}_c^t + \boldsymbol{f}_a^t - \boldsymbol{f}(\boldsymbol{x}^t) \tag{6}$$

where $\Delta \boldsymbol{f}_i = \boldsymbol{f}_i^{t+h} - \boldsymbol{f}_i^t,\ i \in \{p, c, a\}$. Using (3) at time step t, (6) becomes,

$$\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}} \Delta \boldsymbol{x} = \Delta \boldsymbol{f}_p + \Delta \boldsymbol{f}_c + \Delta \boldsymbol{f}_a \tag{7}$$

To find the relationship between the manipulated points and the feature points, like in [16] we partition (7). First we define $\boldsymbol{x}_f = \boldsymbol{S}_f \boldsymbol{x}$, $\boldsymbol{x}_m = \boldsymbol{S}_m \boldsymbol{x}$, $\boldsymbol{x}_c = \boldsymbol{S}_c \boldsymbol{x}$, and $\boldsymbol{x}_o = \boldsymbol{S}_o \boldsymbol{x}$, with $\boldsymbol{S}_f$, $\boldsymbol{S}_m$, $\boldsymbol{S}_c$, and $\boldsymbol{S}_o$ being selection matrices. The indices f, m, c, and o correspond respectively to the feature points (the green points in Fig. 1), the manipulated points (the red points attached to the robot gripper in Fig. 1), the static points if any (the blue points in Fig. 1), and the other points (the black points in Fig. 1). Now we can write $\boldsymbol{x}$ as follows,

$$\boldsymbol{x} = \boldsymbol{S}_f^T \boldsymbol{x}_f + \boldsymbol{S}_m^T \boldsymbol{x}_m + \boldsymbol{S}_o^T \boldsymbol{x}_o + \boldsymbol{S}_c^T \boldsymbol{x}_c \tag{8}$$

Let $\boldsymbol{A_S} = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}$ and $\boldsymbol{b_S} = \Delta \boldsymbol{f}_p + \Delta \boldsymbol{f}_c$, then by applying the partition in (7), dropping small variation Δ for dotted time derivatives, we obtain:

$$\begin{bmatrix} \boldsymbol{S}_f \\ \boldsymbol{S}_m \\ \boldsymbol{S}_o \end{bmatrix} \boldsymbol{A_S} \begin{bmatrix} \boldsymbol{S}_f^T \boldsymbol{S}_m^T \boldsymbol{S}_o^T \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{x}}_f \\ \dot{\boldsymbol{x}}_m \\ \dot{\boldsymbol{x}}_o \end{bmatrix} = \begin{bmatrix} \boldsymbol{S}_f \\ \boldsymbol{S}_m \\ \boldsymbol{S}_o \end{bmatrix} \dot{\boldsymbol{b}}_{\boldsymbol{S}} + \begin{bmatrix} \boldsymbol{S}_f \\ \boldsymbol{S}_m \\ \boldsymbol{S}_o \end{bmatrix} \dot{\boldsymbol{f}}_a \tag{9}$$

since $\dot{\boldsymbol{x}}_c = \boldsymbol{0}$ for the static points. In the quasi-static case, characterized by low velocities and small displacements between consecutive time steps, $\dot{\boldsymbol{b}}_{\boldsymbol{S}}$ can be considered as $\boldsymbol{0}$. Furthermore, as actuation forces are exclusively applied to the manipulated points, (9) is transformed to:

$$\begin{bmatrix} \boldsymbol{A}_{ff} & \boldsymbol{A}_{fm} & \boldsymbol{A}_{fo} \\ \boldsymbol{A}_{mf} & \boldsymbol{A}_{mm} & \boldsymbol{A}_{mo} \\ \boldsymbol{A}_{of} & \boldsymbol{A}_{om} & \boldsymbol{A}_{oo} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{x}}_f \\ \dot{\boldsymbol{x}}_m \\ \dot{\boldsymbol{x}}_o \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{S}_m \dot{\boldsymbol{f}}_a \\ \boldsymbol{0} \end{bmatrix} \tag{10}$$

where $\boldsymbol{A}_{ij} = \boldsymbol{S}_i \boldsymbol{A_S} \boldsymbol{S}_j^T$ with $i, j \in \{f, m, o\}$. From (10) we can deduce,

$$\dot{\boldsymbol{x}}_f = -\boldsymbol{A}_f^{-1} \boldsymbol{A}_m \dot{\boldsymbol{x}}_m \tag{11}$$

where $\boldsymbol{A}_f = \boldsymbol{A}_{ff} - \boldsymbol{A}_{fo} \boldsymbol{A}_{oo}^{-1} \boldsymbol{A}_{of}$ and $\boldsymbol{A}_m = \boldsymbol{A}_{fm} - \boldsymbol{A}_{fo} \boldsymbol{A}_{oo}^{-1} \boldsymbol{A}_{om}$. We thus obtain a linear relationship as established in [16], [11], [21].

2) ***Dynamic case****:* In this scenario, applying the backward Euler method to integrate (2) results in,

$$\boldsymbol{M}(\dot{\boldsymbol{x}}^{t+h} - \dot{\boldsymbol{x}}^t) = h(\boldsymbol{f}_p^{t+h} + \boldsymbol{f}_c^{t+h} - \boldsymbol{f}(\boldsymbol{x}^{t+h}, \dot{\boldsymbol{x}}^{t+h})) + h\boldsymbol{f}_a^{t+h} \tag{12}$$

$$\boldsymbol{x}^{t+h} = \boldsymbol{x}^t + h\dot{\boldsymbol{x}}^{t+h} \tag{13}$$

From (12) one can see that computing the internal elastic forces $\boldsymbol{f}$ requires the position and the velocities at time-step $t+h$, which are unknown at the current time step t. To proceed, we take the first order approximation of the internal forces $\boldsymbol{f}$ [24]:

$$\boldsymbol{f}(\boldsymbol{x}^{t+h}, \dot{\boldsymbol{x}}^{t+h}) = \boldsymbol{f}(\boldsymbol{x}^t, \dot{\boldsymbol{x}}^t) + \frac{\partial \boldsymbol{f}(\boldsymbol{x}^t, \dot{\boldsymbol{x}}^t)}{\partial \boldsymbol{x}} d\boldsymbol{x} + \frac{\partial \boldsymbol{f}(\boldsymbol{x}^t, \dot{\boldsymbol{x}}^t)}{\partial \dot{\boldsymbol{x}}} d\dot{\boldsymbol{x}} \tag{14}$$

with $d\boldsymbol{x} = \boldsymbol{x}^{t+h} - \boldsymbol{x}^t$ and $d\dot{\boldsymbol{x}} = \dot{\boldsymbol{x}}^{t+h} - \dot{\boldsymbol{x}}^t$. Combining (12), (13) and (14) leads to,

$$\left(\boldsymbol{M} + h\frac{\partial \boldsymbol{f}}{\partial \dot{\boldsymbol{x}}} + h^2 \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\right) d\dot{\boldsymbol{x}} =$$

$$- h^2 \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}} \dot{\boldsymbol{x}}^t + h\left(\boldsymbol{f}_p^{t+h} + \boldsymbol{f}_c^{t+h} - \boldsymbol{f}(\boldsymbol{x}^t, \dot{\boldsymbol{x}}^t)\right) + h\boldsymbol{f}_a^{t+h} \tag{15}$$

Before applying partition to (15), we obtain from (8):

$$d\dot{\boldsymbol{x}} = \boldsymbol{S}_f^T d\dot{\boldsymbol{x}}_f + \boldsymbol{S}_m^T d\dot{\boldsymbol{x}}_m + \boldsymbol{S}_o^T d\dot{\boldsymbol{x}}_o \tag{16}$$

since $\dot{\boldsymbol{x}}_c = \boldsymbol{0}$ for static points. Then, let

$$\boldsymbol{A_D} = \boldsymbol{M} + h\frac{\partial \boldsymbol{f}}{\partial \dot{\boldsymbol{x}}} + h^2 \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}} \tag{17}$$

$$\boldsymbol{b_D} = -h^2 \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}} \dot{\boldsymbol{x}}^t + h\left(\boldsymbol{f}_p^{t+h} + \boldsymbol{f}_c^{t+h} - \boldsymbol{f}(\boldsymbol{x}^t, \dot{\boldsymbol{x}}^t)\right) \tag{18}$$

Applying the partition to (15), and using (16), we obtain:

$$\begin{bmatrix} \boldsymbol{S}_f \\ \boldsymbol{S}_m \\ \boldsymbol{S}_o \end{bmatrix} \boldsymbol{A_D} \begin{bmatrix} \boldsymbol{S}_f^T \boldsymbol{S}_m^T \boldsymbol{S}_o^T \end{bmatrix} \begin{bmatrix} d\dot{\boldsymbol{x}}_f \\ d\dot{\boldsymbol{x}}_m \\ d\dot{\boldsymbol{x}}_o \end{bmatrix} = \begin{bmatrix} \boldsymbol{S}_f \\ \boldsymbol{S}_m \\ \boldsymbol{S}_o \end{bmatrix} \boldsymbol{b_D} + h \begin{bmatrix} \boldsymbol{S}_f \\ \boldsymbol{S}_m \\ \boldsymbol{S}_o \end{bmatrix} \boldsymbol{f}_a^{t+h} \tag{19}$$

Since there is no external actuation forces applied on the feature points and on the other points, (19) simplifies to:

$$\begin{bmatrix} \boldsymbol{A}_{ff} & \boldsymbol{A}_{fm} & \boldsymbol{A}_{fo} \\ \boldsymbol{A}_{mf} & \boldsymbol{A}_{mm} & \boldsymbol{A}_{mo} \\ \boldsymbol{A}_{of} & \boldsymbol{A}_{om} & \boldsymbol{A}_{oo} \end{bmatrix} \begin{bmatrix} d\dot{\boldsymbol{x}}_f \\ d\dot{\boldsymbol{x}}_m \\ d\dot{\boldsymbol{x}}_o \end{bmatrix} = \begin{bmatrix} \boldsymbol{b}_f \\ \boldsymbol{b}_m + h\boldsymbol{S}_m \boldsymbol{f}_a^{t+h} \\ \boldsymbol{b}_o \end{bmatrix} \tag{20}$$

where $\boldsymbol{A}_{ij} = \boldsymbol{S}_i \boldsymbol{A_D} \boldsymbol{S}_j^T$, $\boldsymbol{b}_i = \boldsymbol{S}_i \boldsymbol{b_D}$, with $i, j \in \{f, m, o\}$. Let us note that $\boldsymbol{A}_{ij}$ share the same expressions as in the previous section. However, their values completely differ since they are obtained from $\boldsymbol{A_D}$, not from $\boldsymbol{A_S}$. Finally, from (20) we deduce,

$$d\dot{\boldsymbol{x}}_f = -\boldsymbol{A}_f^{-1} \boldsymbol{A}_m d\dot{\boldsymbol{x}}_m + \boldsymbol{A}_f^{-1} (\boldsymbol{b}_f - \boldsymbol{A}_{fo} \boldsymbol{A}_{oo}^{-1} \boldsymbol{b}_o) \tag{21}$$

Note that $\boldsymbol{A}_f$ and $\boldsymbol{A}_{oo}$ are always invertible since (7) and (15) have unique solutions, given that $\boldsymbol{A_S}$ and $\boldsymbol{A_D}$ are non-singular matrices. This is because the mass matrix $\boldsymbol{M}$ and the stiffness matrix $\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}$ are positive definite [23], and the damping term $\frac{\partial \boldsymbol{f}}{\partial \dot{\boldsymbol{x}}}$ is modeled as Rayleigh damping. Since $d\dot{\boldsymbol{x}}_f = \dot{\boldsymbol{x}}_f^{t+h} - \dot{\boldsymbol{x}}_f^t$

and $d\dot{\boldsymbol{x}}_m = \dot{\boldsymbol{x}}_m^{t+h} - \dot{\boldsymbol{x}}_m^t$, it follows from the previous equation that,

$$\dot{\boldsymbol{x}}_f^{t+h} = -\boldsymbol{A}_f^{-1}\boldsymbol{A}_m\dot{\boldsymbol{x}}_m^{t+h} + \boldsymbol{\gamma} \tag{22}$$

where $\boldsymbol{\gamma} = \boldsymbol{A}_f^{-1}(\boldsymbol{b}_f - \boldsymbol{A}_{fo}\boldsymbol{A}_{oo}^{-1}\boldsymbol{b}_o + \boldsymbol{A}_m\dot{\boldsymbol{x}}_m^t) + \dot{\boldsymbol{x}}_f^t$ is a term that depends on the internal elastic forces applied on the object and on the velocities of the manipulated and feature points at the previous time-step. In summary, $\boldsymbol{\gamma}$ encapsulates the cumulative effects of previous actions applied to the manipulated object. Note that this term does not appear in the quasi-static case.

B. Control

Let us now define $\boldsymbol{x}_f^*$ as the desired 3D position of the feature points $\boldsymbol{x}_f$. Our control objective is to drive $\boldsymbol{x}_f$ exponentially toward $\boldsymbol{x}_f^*$ by acting on manipulated points $\boldsymbol{x}_m$ through the robot gripper.

To model the interaction between the robot gripper and the deformable object, as in [25] we consider the part of the object rigidly attached to the robot gripper as a rigid body, whose center of gravity G coincides with the center of the robot gripper. On Fig. 1 one can clearly see that the manipulated points (red points) compose this rigid body. Let $\mathcal{A}_m$ be the set of manipulated points, $\boldsymbol{\omega}_G \in \mathbb{R}^3$ the angular velocity of the robot gripper, and $\boldsymbol{v}_G \in \mathbb{R}^3$ its linear velocity. For a point $P_i \in \mathcal{A}_m$ in the rigid part of the deformable object, we can write its velocity as follows,

$$\dot{\boldsymbol{x}}_{P_i} = \boldsymbol{v}_G - (\boldsymbol{x}_{P_i} - \boldsymbol{x}_G) \times \boldsymbol{\omega}_G \tag{23}$$

For all points $P_i \in \mathcal{A}_m$ we have:

$$\dot{\boldsymbol{x}}_m = \begin{bmatrix} \dot{\boldsymbol{x}}_{P_1} \\ \dot{\boldsymbol{x}}_{P_2} \\ \vdots \\ \dot{\boldsymbol{x}}_{P_M} \end{bmatrix} = \begin{bmatrix} \boldsymbol{I}_{3\times3} & -[\boldsymbol{x}_{P_1} - \boldsymbol{x}_G]_\times \\ \boldsymbol{I}_{3\times3} & -[\boldsymbol{x}_{P_2} - \boldsymbol{x}_G]_\times \\ & \vdots \\ \boldsymbol{I}_{3\times3} & -[\boldsymbol{x}_{P_M} - \boldsymbol{x}_G]_\times \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_G \\ \boldsymbol{\omega}_G \end{bmatrix} = \boldsymbol{J} \begin{bmatrix} \boldsymbol{v}_G \\ \boldsymbol{\omega}_G \end{bmatrix} \tag{24}$$

where $\boldsymbol{I}_{3\times3}$ is the identity matrix of size 3×3, M is the number of manipulated points, and $[\boldsymbol{u}]_\times$ is the skew symmetric matrix related to the vector $\boldsymbol{u}$.

We substitute (24) in (22) to obtain the following expression:

$$\dot{\boldsymbol{x}}_f = \boldsymbol{J_D} \begin{bmatrix} \boldsymbol{v}_G \\ \boldsymbol{\omega}_G \end{bmatrix} + \boldsymbol{\gamma} \tag{25}$$

where $\boldsymbol{J_D} = -\boldsymbol{A}_f^{-1}\boldsymbol{A}_m\boldsymbol{J}$ is the deformation Jacobian. To fulfill our control objective, using (25) we propose the following control law,

$$\begin{bmatrix} \boldsymbol{v}_G \\ \boldsymbol{\omega}_G \end{bmatrix} = -\boldsymbol{J_D}^+ \left(\lambda \left(\boldsymbol{x}_f - \boldsymbol{x}_f^*\right) + \boldsymbol{\gamma}\right) \tag{26}$$

with $\boldsymbol{J_D}^+$ the pseudo-inverse of $\boldsymbol{J_D}$ and λ a positive gain. Using the quasi-static model to design a control law leads to a control scheme akin to (26), with the strong distinction that the deformation Jacobian $\boldsymbol{J_D}$ is not the same, and there is no feedforward term.

III. Experiments

A. Experimental Setup and Implementation

Fig. 2 illustrates the experimental setup employed to validate our methodology. It features a 7-DOF Franka Emika Panda robot with its gripper rigidly attached to the soft body during manipulation. The setup also includes a static RGB-D Intel RealSense-D435 camera for the visual tracking of the object shape and markers located on it that represent the feature points. For the software part (see Fig. 3), we used the ViSP library [26] both to acquire/process the RGB-D frames from the camera and to control the robot. The terms essential for computing our control law can not be obtained directly from the object itself but rather from a simulator utilizing the FEM to solve the dynamics equations of a deformable elastic object. For that, we used SOFA [27]. As illustrated in Fig. 3, we can have access from SOFA to all the quantities required to compute the proposed control law (26). Finally, all computations were executed on a laptop with Intel® CoreTM i7 CPU @ 2.70GHz × 20, leading to a control rate at 60 Hz.

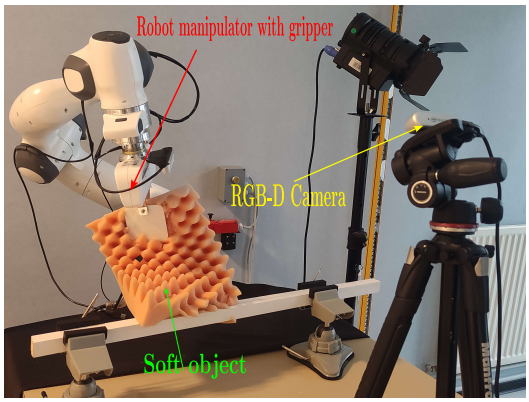

Fig. 2: Experimental setup.

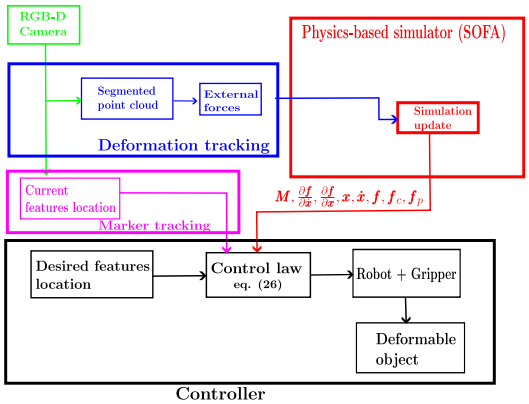

Fig. 3: Block diagram of our approach: visual deformation tracking in blue, visual marker tracking in pink, physics-based simulator in red and the closed-loop control law in black.

B. Deformation Tracking using an RGB-D camera

Tracking the deformations of the soft object is an important part of our approach. To obtain a good estimate of the

deformation Jacobian $\boldsymbol{J_D}$ and the additional term $\boldsymbol{\gamma}$, it is crucial that the object model closely mirrors the configuration of the real object. To achieve this alignment, we segment the current point cloud of the object that is provided by the RGB-D camera and use it to constrain the model to match the shape of the real object. This strategy enhances the robustness to uncertainties in the physical parameters of the object and mitigates errors accumulated during simulation, which could otherwise lead to an unrealistic object representation.

The problem of tracking deformation consists of fitting the segmented point cloud of the object with its volumetric mesh [28]. To obtain this volumetric mesh, first, we need a closed surface mesh, which can be obtained either through 3D reconstruction techniques or from a 3D CAD model. Subsequently, a coarse volumetric mesh is generated by filling the surface mesh with tetrahedra, a process facilitated using the Gmsh tool [29].

Let us denote by $\boldsymbol{Y} = \{\boldsymbol{y}_i\}_{i=1}^{n_{\boldsymbol{Y}}}$ the segmented point cloud of the object at the current image/depth frame k. To obtain $\boldsymbol{Y}$, the object is first segmented from the RGB image using a color thresholding algorithm. The resulting segmented image is intersected with the aligned depth map to obtain the segmented depth map of the object which is back-projected in the camera frame to obtain the segmented point cloud of the visible surface of the object. Let $\boldsymbol{X} = \{\boldsymbol{x}_i\}_{i=1}^{n_{\boldsymbol{X}}}$ be the set of vertices of the volumetric mesh at the previous image/depth frame $k-1$. The goal is then to find the positions of points in $\boldsymbol{X}$ at the current image/depth frame k, by fitting $\boldsymbol{X}$ to $\boldsymbol{Y}$ following the same procedure as described in [28], where forces are computed to be applied on the volumetric mesh to deform it. By relying only on forces computed using the method described in [28], excessive oscillations may occur in the tracking. To prevent this, we also add damping forces computed as described in [30]. Those forces can be regarded as visual constraints. In addition to the visual constraints, we incorporate other known constraints, such as fixed position constraints for the static points, and the position and orientation of the robot gripper. The physics-based simulator (in our case, SOFA) then estimates the deformation of the volumetric mesh by solving the dynamic equations under these constraints before the next image/depth frame $k+1$ arrives.

In addition to the volumetric mesh of the object, SOFA requires its physical parameters namely the Young modulus E and the Poisson ratio ν. Achieving accurate estimations of these parameters involves conducting meticulous material testing with the appropriate equipment. An alternative, albeit less precise method [12], involves utilizing an RGB-D sensor, a force sensor mounted on the robot, and a physics-based simulator (SOFA) for identifying E and ν. Notably, our approach does not necessitate a precise estimation of E and ν, a rough estimate is sufficient, and the identification technique proposed in [31] was satisfactory enough.

C. Results

The effectiveness of our deformation control scheme was evaluated using two distinct soft objects, as illustrated in Fig. 4. Their mesh was respectively composed of 378 and 450 points. To validate our methodology, we measured the 3D positional error between the desired positions of the feature points and their actual positions during the time evolution of the robotic task, aiming to exponentially drive this positional error close to zero. In the following, we present results on the control of one and two feature points. In all the subsequent experiments, the control gain λ in (26) has been set to $\lambda = 1.5$. The results are also presented in the accompanying video.

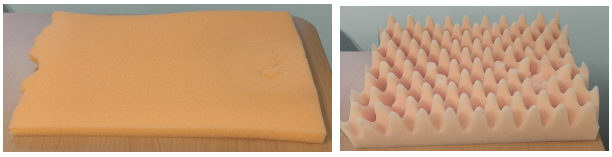

Fig. 4: Soft objects used in our experiments.

The first experiment is presented in Fig. 5. In that case, there are no static object points and only the 3D position of one feature point (in green) is controlled. It has been selected far away from the gripper to enhance the influence of dynamic effects. This green point represents the projection of the feature point in the image, using a fiducial marker rigidly attached to the object so that it is easily tracked using the ViSP Libray [26]. It is requested to reach first the 3D position depicted in red and then the 3D position depicted in yellow. The error of the positioning task is directly measured by the RGB-D camera. The measured depth exhibits oscillations within a 2 mm range, attributed to the depth measurement noise inherent to this type of camera. Consequently, convergence is considered when the error norm is within 2 mm. The blue, green, yellow, and red solid lines in Figs. 5a and 5c represent respectively the evolution of the positioning error along ($\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{z}$) axes of the camera frame and the error norm.

The results obtained with our complete control law (26) are presented on Figs. 5a and 5b while the results obtained with the control law derived with the quasi static assumption are depicted on Figs. 5c and 5d. Our control law clearly exhibits better performance compared to the alternative method in terms of time-to-convergence, and it demonstrates reduced sensitivity to dynamic effects, such as oscillations induced by gravity and rapid and noisy movements of the gripper observed when using the quasi-static assumption. The dynamic performance of our complete controller is also more apparent in the accompanying video.

The second experiment is presented in the first column of Fig. 6, which again consists of positioning one point on the surface of the object at a desired 3D location. Now, the bottom of the object is rigidly attached to a bar and, consequently, constrained to be static. This is to enhance the compliant behavior of the object during the manipulation task. The green and yellow points in Fig. 6a, as well as the blue, green, yellow, and red solid lines in Fig. 6i have the same meaning as in the previous experiment. We can note that the proposed control law rapidly decreases the error norm exponentially, reducing it from approximately 150 mm to close to 0 mm, which validates the modeling proposed in the previous section. We can notice a convergence in less than 3 $seconds$.

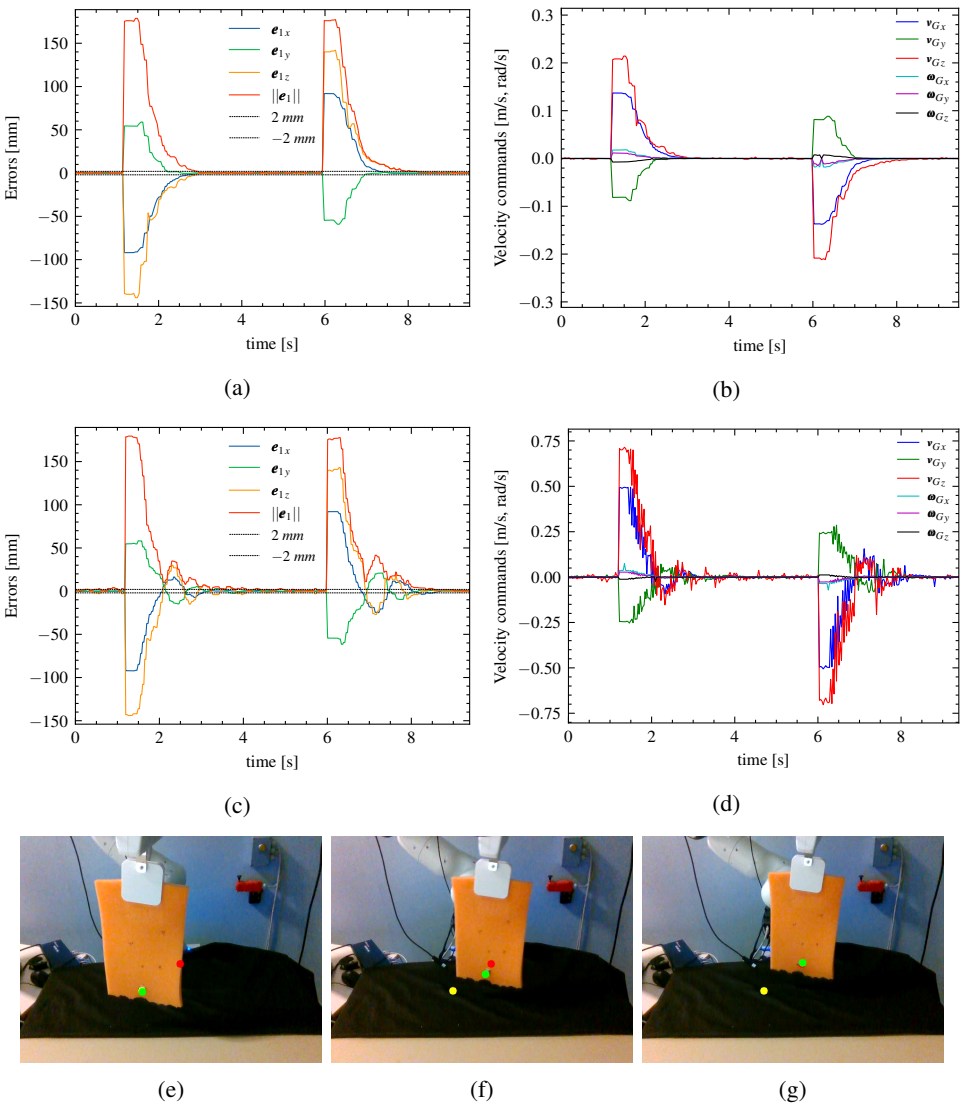

Fig. 5: First experiment: (a-b) Evolution of the positioning errors and robot gripper control velocity in the dynamic case. (c-d) Evolution of the positioning errors and robot gripper control velocity in the quasi-static case. (e-g) Snapshot of the experiment. From left to right: initial configuration, intermediate configuration, configuration at the end of the first task.

Since our gripper has 6 DOF, it is possible to control the 3D position of 2 object points. The second column of Fig. 6 presents the results of such a task. In Fig. 6e the green points represent the projection of the feature points in the image as in the previous scenario. The yellow and red points represent respectively the projection in the image of the desired locations of the left green point and right green point. Fig. 6j shows the evolution of the 3D positioning errors along the ($\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{z}$) axes of the camera frame of each feature point. The plots $\boldsymbol{e}_{1x}$, $\boldsymbol{e}_{1y}$, and $\boldsymbol{e}_{1z}$ correspond to the left feature point, while the plots $\boldsymbol{e}_{2x}$, $\boldsymbol{e}_{2y}$, and $\boldsymbol{e}_{2z}$ correspond to the right feature point. The plot $||\boldsymbol{e}||$ depicts the evolution of the error norm in the positioning of the two points over time. Like in the previous experiment, we can notice a fast exponential convergence of the different errors close to $0\,mm$ in less than 3 $seconds$. Figs. 6k and 6l depict the evolution of the velocities of the robot-gripper over time during the manipulation. In the case of one point, our controller exhibits almost no rotational velocity since there are only 3 DOF to control. This is in contrast to the case with two points where gripper rotation is required to achieve the task. It is important to note that the manipulator can only drive the features to physically reasonable targets, depending also on the grasping configuration.

The results for the second object depicted on Fig. 4 are presented in the accompanying video. Due to its more complex shape and perception limitations, we can note a more shakier behavior. In addition, the video also demonstrates the robustness of our control scheme against external disturbances.

IV. Conclusion

In this letter, we presented a physics-based approach to control the 6 DOF of a robot gripper using an RGB-D camera to position feature points belonging to a soft object by manipulating distant points. We also combined real-time visual tracking of the object for minimizing the gap between

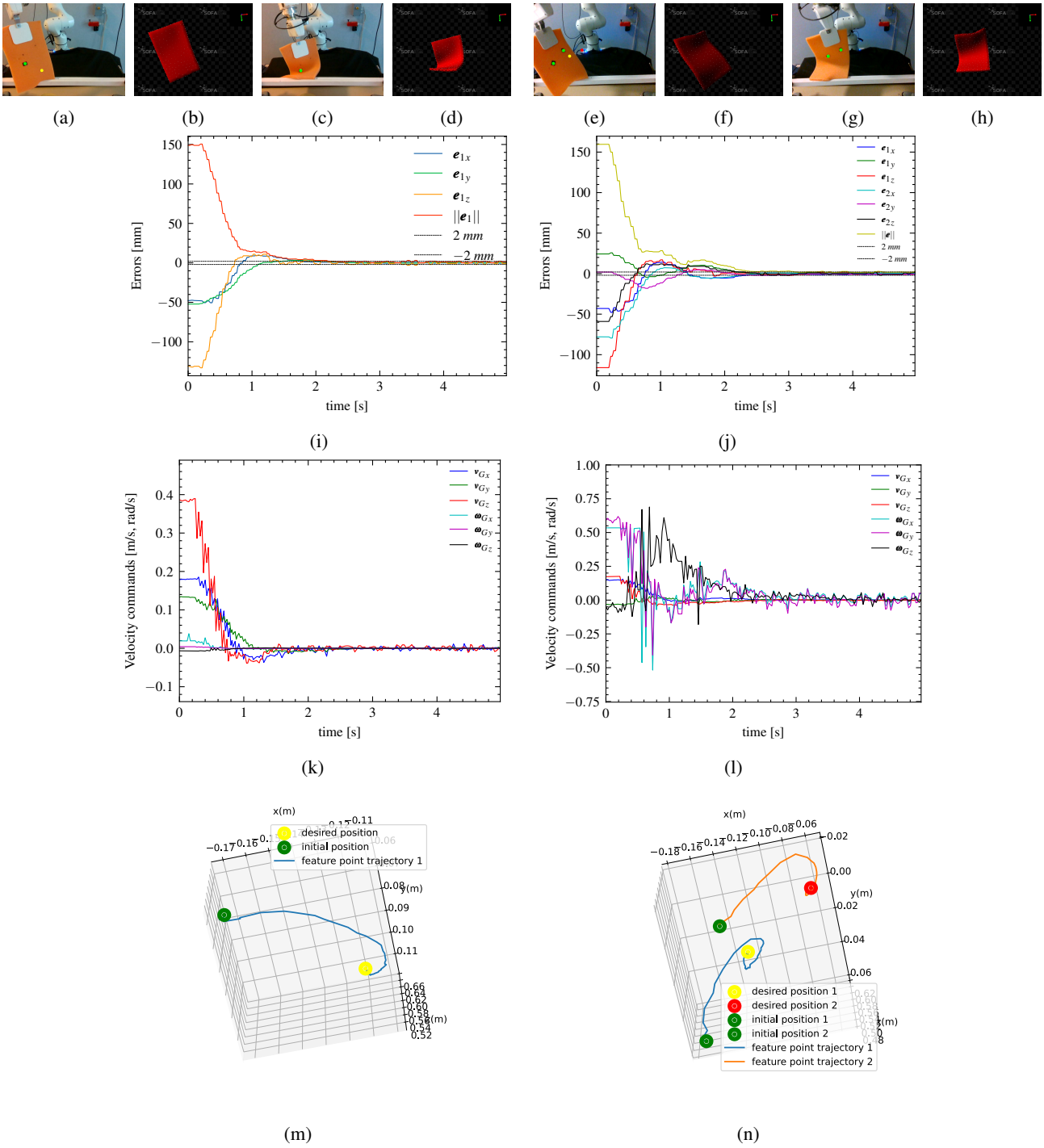

Fig. 6: Second and third experiment: the first column show the results for positioning one point while the second column show the results for positioning two points. (a), (e) Soft object in initial state. (b), (f) Model of the soft object in its initial state within SOFA. (c), (g) Soft object after deformation process. (d), (h) Model of the soft object after deformation process within SOFA. (i), (j) 3D errors in mm measured over time for the one-point and two-points positioning task respectively. (k), (l) linear velocities in m/s and angular velocities in rad/s of the robot gripper for the one-point and two-points positioning task respectively. (m), (n) 3D trajectory in the camera frame of the 3D feature points from their initial position in green to their desired position in yellow and red for the one-point and two-points positioning task respectively.

the FEM model used and the deformations observed from the RGB-D camera. Based on the FEM model, we derived the analytical relationship between the motion of the robot gripper and the motion of the feature points without considering the classical quasi-static assumption. Using that relation we proposed a visual servoing framework to automatically control the deformation of a soft object. The proposed approach was evaluated in real experiments involving two deformable objects

with different topology, and appealing results were presented involving large deformations. Experiments also show that it is beneficial to take into account the dynamic effect in the control law to compensate for the transient dynamic behavior of the object, leading to a far better behavior than the one obtained using the quasi-static assumption.

As future work, we will focus on the control of the whole shape of the object using a dual-arm robot. Since the shape control of deformable objects is an under-actuated problem, it will be necessary to define more advanced features that could provide a controllable, low-dimensional representation of the overall shape. To conclude, by analogy with classical visual servoing where the interaction matrix related to a point of a rigid object is involved for designing more complex visual features such as moments [32], we have introduced in this work its counterpart for a point belonging to a deformable object. From this knowledge we plan to elaborate high-level features for soft object manipulation in future work.

ACKNOWLEDGMENT

The authors would like to thank Samuel Felton for his help during the development of experimental validations.

REFERENCES

[1] J. Zhu, A. Cherubini, C. Dune, D. Navarro-Alarcon, F. Alambeigi, D. Berenson, F. Ficuciello, K. Harada, J. Kober, X. Li, J. Pan, W. Yuan, and M. Gienger, "Challenges and outlook in robotic manipulation of deformable objects," *IEEE Robotics & Automation Magazine*, vol. 29, no. 3, pp. 67–77, 2022.

[2] S. Hirai, "Indirect simultaneous positioning of deformable objects without physical parameters or time-derivatives," in *Mechatronics for Safety, Security and Dependability in a New Era*. Oxford: Elsevier, 2007, pp. 81–86.

[3] D. Navarro-Alarcon, H. M. Yip, Z. Wang, Y.-H. Liu, F. Zhong, T. Zhang, and P. Li, "Automatic 3-d manipulation of soft objects by robotic arms with an adaptive deformation model," *IEEE Transactions on Robotics*, vol. 32, no. 2, pp. 429–441, 2016.

[4] R. Lagneau, A. Krupa, and M. Marchal, "Automatic shape control of deformable wires based on model-free visual servoing," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5252–5259, 2020.

[5] D. Navarro-Alarcón, Y.-H. Liu, J. G. Romero, and P. Li, "Model-free visually servoed deformation control of elastic objects by robot manipulators," *IEEE Transactions on Robotics*, vol. 29, no. 6, pp. 1457–1468, 2013.

[6] Z. Hu, T. Han, P. Sun, J. Pan, and D. Manocha, "3-D Deformable Object Manipulation Using Deep Neural Networks," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4255–4261, Oct. 2019.

[7] B. Thach, B. Y. Cho, A. Kuntz, and T. Hermans, "Learning visual shape control of novel 3d deformable objects from partial-view point clouds," in *Int. Conf. on Robotics and Automation (ICRA)*, 2022, pp. 8274–8281.

[8] Z. Hu, P. Sun, and J. Pan, "Three-dimensional deformable object manipulation using fast online gaussian process regression," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 979–986, 2018.

[9] C. Shin, P. W. Ferguson, S. A. Pedram, J. Ma, E. P. Dutson, and J. Rosen, "Autonomous tissue manipulation via surgical robot using learning based model predictive control," in *Int. Conf. on Robotics and Automation (ICRA)*, 2019, pp. 3875–3881.

[10] T. Wada, S. Hirai, and S. Kawamura, "Indirect simultaneous positioning operations of extensionally deformable objects," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, vol. 2, 1998, pp. 1333–1338.

[11] T. Wada, S. Hirai, S. Kawamura, and N. Kamiji, "Robust manipulation of deformable objects by a simple pid feedback," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, vol. 1, 2001, pp. 85–90.

[12] F. Makiyeh, M. Marchal, F. Chaumette, and A. Krupa, "Indirect positioning of a 3d point on a soft object using rgb-d visual servoing and a mass-spring model," in *International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 2022, pp. 235–242.

[13] S. Kinio and A. Patriciu, "A comparative study of H_∞ and PID control for indirect deformable object manipulation," in *IEEE Int. Conf. on Robotics and Biomimetics (ROBIO)*, 2012, pp. 414–420.

[14] F. Ficuciello, A. Migliozzi, E. Coevoet, A. Petit, and C. Duriez, "FEM-Based Deformation Control for Dexterous Manipulation of 3D Soft Objects," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, Oct. 2018, pp. 4007–4013.

[15] S. Duenser, J. M. Bern, R. Poranne, and S. Coros, "Interactive Robotic Manipulation of Elastic Objects," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, Oct. 2018, pp. 3476–3481.

[16] A. Koessler, N. R. Filella, B. Bouzgarrou, L. Lequièvre, and J.-A. C. Ramon, "An efficient approach to closed-loop shape control of deformable objects using finite element models," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, May 2021, pp. 1637–1643.

[17] R. K. Katzschmann, M. Thieffry, O. Goury, A. Kruszewski, T.-M. Guerra, C. Duriez, and D. Rus, "Dynamically closed-loop controlled soft robotic arm using a reduced order finite element model with state observer," in *2019 2nd IEEE International Conference on Soft Robotics (RoboSoft)*, 2019, pp. 717–724.

[18] K. Wu and G. Zheng, "Fem-based gain-scheduling control of a soft trunk robot," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3081–3088, 2021.

[19] R. Fanson and A. Patriciu, "Model based deformable object manipulation using linear robust output regulation," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2010, pp. 496–501.

[20] M. Shetab-Bushehri, M. Aranda, Y. Mezouar, and E. Özgür, "As-rigid-as-possible shape servoing," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3898–3905, 2022.

[21] ——, "Lattice-based shape tracking and servoing of elastic objects," *IEEE Transactions on Robotics*, vol. 40, pp. 364–381, 2024.

[22] J. Bonet, A. J. Gil, and R. D. Wood, *Nonlinear Solid Mechanics for Finite Element Analysis: Dynamics*. Cambridge University Press, 2021.

[23] Thomas J. R. Hughes, *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Prentice Hall, 1987.

[24] D. Baraff and A. Witkin, *Large Steps in Cloth Simulation*, 1st ed. New York, NY, USA: Association for Computing Machinery, 2023.

[25] E. Coevoet, Y. Adagolodjo, M. Lin, C. Duriez, and F. Ficuciello, "Planning of Soft-Rigid Hybrid Arms in Contact With Compliant Environment: Application to the Transrectal Biopsy of the Prostate," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4853–4860, Apr. 2022.

[26] E. Marchand, F. Spindler, and F. Chaumette, "ViSP for visual servoing: a generic software platform with a wide class of robot control skills," *IEEE Robotics and Automation Magazine*, vol. 12, no. 4, pp. 40–52, December 2005.

[27] F. Faure, C. Duriez, H. Delingette, J. Allard, B. Gilles, S. Marchesseau, H. Talbot, H. Courtecuisse, G. Bousquet, I. Peterlik, and S. Cotin, "SOFA: A Multi-Model Framework for Interactive Physical Simulation," in *Soft Tissue Biomechanical Modeling for Computer Assisted Surgery*, Y. Payan, Ed. Springer, June 2012, vol. 11, pp. 283–321.

[28] A. Petit, V. Lippiello, G. A. Fontanelli, and B. Siciliano, "Tracking elastic deformable objects with an rgb-d sensor for a pizza chef robot," *Robotics and Autonomous Systems*, vol. 88, pp. 187–201, 2017.

[29] C. Geuzaine and J.-F. Remacle, "Gmsh: A 3-D Finite Element Mesh Generator with Built-in Pre- and Post-Processing Facilities," *Int. Journal for Numerical Methods in Engineering*, vol. 79, pp. 1309–1331, Sept. 2009.

[30] T. Tang and M. Tomizuka, "Track deformable objects from point clouds with structure preserved registration," *The Int. Journal of Robotics Research*, vol. 41, no. 6, pp. 599–614, May 2022.

[31] A. Sengupta, R. Lagneau, A. Krupa, E. Marchand, and M. Marchal, "Simultaneous tracking and elasticity parameter estimation of deformable objects," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2020, pp. 10 038–10 044.

[32] F. Chaumette, "Image moments: a general and useful set of features for visual servoing," *IEEE Transactions on Robotics*, vol. 20, no. 4, pp. 713–723, 2004.