



**HAL**  
open science

## FEM-based Deformation Control for Dexterous Manipulation of 3D Soft Objects

Fanny Ficuciello, Alessandro Migliozi, Eulalie Coevoet, Antoine Petit,  
Christian Duriez

► **To cite this version:**

Fanny Ficuciello, Alessandro Migliozi, Eulalie Coevoet, Antoine Petit, Christian Duriez. FEM-based Deformation Control for Dexterous Manipulation of 3D Soft Objects. IROS 2018 - IEEE/RSJ International Conference on Intelligent Robots and Systems, Oct 2018, Madrid, Spain. 10.1109/IROS.2018.8593512 . hal-01889783

**HAL Id: hal-01889783**

**<https://inria.hal.science/hal-01889783v1>**

Submitted on 8 Oct 2018

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# FEM-based Deformation Control for Dexterous Manipulation of 3D Soft Objects

F. Ficuciello<sup>1</sup>, A. Migliozi<sup>1</sup>, E. Coevoet<sup>3</sup>, A. Petit<sup>2</sup> and C. Duriez<sup>3</sup>

**Abstract**—In this paper, a method for dexterous manipulation of 3D soft objects for real-time deformation control is presented, relying on Finite Element modelling. The goal is to generate proper forces on the fingertips of an anthropomorphic device during in-hand manipulation to produce desired displacements of selected control points on the object. The desired motions of the fingers are computed in real-time as an inverse solution of a Finite Element Method (FEM), the forces applied by the fingertips at the contact points being modelled by Lagrange multipliers. The elasticity parameters of the model are preliminarily estimated using a vision system and a force sensor. Experimental results are shown with an underactuated anthropomorphic hand that performs a manipulation task on a soft cylindrical object.

## I. INTRODUCTION

Robotic manipulation of deformable objects is a challenging problem in many tasks such as food handling [1], manipulation of ropes, cables or wires [2], cloths folding [3], or in robotic surgery for operations on organs or tissues [4]. While research on grasping and manipulation of 3D rigid objects has gained a certain maturity, dealing with non-rigid objects is a much more complex task, for which the community has been recently driven important efforts.

Indeed, controlling the deformations of a soft object towards a desired shape faces the potential infinite number of degrees of freedom to be controlled. Among the wide variety of recent works in the literature aiming at addressing this difficult problem, we can distinguish, as in many robotic tasks, two classes: model-based and learning-based approaches.

Model-based methods rely on the design of a deformation model, with more or less realistic mechanical properties, represented by stiffness or elasticity coefficients over Finite Element models, spring systems [5], [6], [7] or particle systems [8] or by heuristics [9] etc... The problem can be handled as a planning or an open loop deformation control scheme [8], [10], [11] for which the deformation properties have to be known or estimated beforehand in a calibration step. Servo-control solutions, also known as shape servoing, have also been proposed [7], [12], [5], [6] which enable an online estimation of the deformation parameters along with

the control process, by processing the visual feedback of the system.

Learning-based strategies, which employ higher level representations of the deformations, have also been popular. Usually in a learning from demonstration manner, a mapping between observed deformations and the actuation of the manipulator can be learnt and used online for manipulation [13], with reinforcement learning [14], tangent space mapping [15], or deep learning [16] frameworks. In an online learning manner, a Gaussian Process Regression process has been recently employed in [17] to model and estimate the deformation parameters of the object in a visual-servo control. However, these approaches, which have the advantage of not requiring any specific modelling and calibration, often depend on learning phases which can hardly generalize and can suffer from inaccuracies.

Besides, these aforementioned works consider simple manipulators and end-effectors such as grippers to control the deformations. Regarding dexterous manipulation with a multifingered hand, several contributions have been proposed to manipulate a rigid object with compliant hands [18], [19], [20]. Nevertheless, to our knowledge, the converse task of manipulating a compliant soft object with a rigid multifingered hand remains an open problem.

In this work, we propose a model-based approach, relying on volumetric elastic Finite Element modelling, for the manipulation of a deformable object with a robotic multifingered hand. By sticking closely to the physics, we suggest to control the deformations in an accurate and physically consistent manner, using an efficient inversion method of the model. Indeed, inverse resolution techniques for Finite Element Methods (FEM) have recently impressively improved in terms of both physical accuracy and real-time performance, which had been a main limitation so far. The benefit and prospects of this solution have been demonstrated for the control of soft robots [21], [22]. Thus, adapting this framework to our active deformation control task with a robotic anthropomorphic hand appears as an appealing solution. Our strategy also takes advantage of the availability of a vision-based offline calibration technique for the estimation of the elasticity parameters [23]. As a preliminary work, we restrict the approach to an open-loop control scheme.

The paper is organized as follows. Section II presents the deformation model employed for the manipulated object, as well as the interaction model with the robotic hand. The method for the estimation of the mechanical properties is also described. In Section III the inverse simulation framework to

The research has been partially supported by RoDyMan project under ERC AdG-320992, and by MUSA National Italian Grant.

<sup>1</sup> F. Ficuciello and A. Migliozi are with PRISMA Lab, University of Naples Federico II, via Claudio 21, 80125, Naples, Italy, email: {fanny.ficuciello}@unina.it, {migliozi.alessandro}@gmail.com .

<sup>2</sup> A. Petit is with Mimesis team: INRIA, 1, place de l'Hopital 67000 Strasbourg, France {antoine.a.petit}@inria.fr

<sup>3</sup>E. Coevoet and C. Duriez are with Defrost team: INRIA, CNRS, University of Lille and Ecole Centrale de Lille, France {eulalie.coevoet, christian.duriez}@inria.fr

perform deformation control with contact point actuation is introduced. The kinematic model and control of the used robotic hand is presented in Section IV. Finally some results on simulated and real world manipulation tasks of a simple soft objects are shown in Section V.

## II. FEM DEFORMATION AND INTERACTION MODELLING

In this work, the manipulated object is constituted by a continuous isotropic material whose elasticity can be modeled using a volumetric linear Finite Element Method (FEM). An FEM model consists in tessellating a volume of the object into a mesh of elements (tetrahedrons) connecting a set  $\mathbf{x} = \{\mathbf{x}_j\}_{j=1}^{n_x}$  of 3D vertices. The behavior of the manipulated object can be formulated as a quasi-static equilibrium between internal forces inside the deformable structure  $\mathbf{f}(\mathbf{x})$  and external forces due to gravity and interactions with the environment.

In the case of a robotic anthropomorphic dexterous hand manipulating a deformable object, these external forces are the contact forces exerted on the object by the hand. In this study gravity is neglected and the contacts between the hand and the object occur at the fingertips. Our approach is based on the formulation developed for soft robot control in [21], for which a set of end-effector points defined on a robot can be controlled by actively deforming the robot by a set of actuators, through an inverse FEM problem.

In an analogous manner, we can consider here the fingertips as actuators, and we define as Lagrange multipliers  $\lambda_a$  as being the force intensities applied on a limited number of points located on the object. The directions of the efforts are given by a matrix  $\mathbf{H}_a$ . Considering that no load is applied at the end-effector points (or control points) on the object ( $\lambda_e = 0$ ), the quasi-static formulation of the equilibrium can be written as:

$$\mathbf{f} + \mathbf{H}_a^T \lambda_a = 0. \quad (1)$$

In order to model the internal forces  $\mathbf{f}$ , we resort to the linear elasticity, through Hooke's law, and to the infinitesimal strain theories [24].  $\mathbf{f}$  can then be linearly related to the displacements  $\hat{\mathbf{u}}$  of the vertices of the mesh:

$$\mathbf{f} = \mathbf{K} \hat{\mathbf{u}} \quad (2)$$

$\mathbf{K}$  being the stiffness matrix, depending on two elastic parameters of the material, the Young modulus  $E$  and Poisson ratio  $\nu$ . This formulation is modified by adopting a corotational approach [25] to accommodate for rotation transformations, by warping the stiffness with respect to a rotation matrix corresponding to the rotational component of the element deformations. For further investigation on this topic the reader can refer to [26].

### A. Estimation of the elasticity parameters

A requirement for a reliable physics-based manipulation is the estimation of both  $E$  and  $\nu$ . We employ the approach proposed in [23], which consists in minimizing a fitting error between the simulated deformations and the deformations

captured by an RGB-D sensor (Asus Xtion). The simulation relies on the input force measured by means of a force sensor mounted on a robotic arm and on a simple interaction model between the object, the robotic tool and the underlying flat surface. The set-up is shown in Fig. 1. The deformations are



Fig. 1. Setup used for object mechanical parameters estimation.

generated by applying an effort, as point-wise as possible, on the deformable object using the robot. They are observed using the RGBD camera, once static equilibrium is reached.

The simulated deformations are computed using the FEM model presented above, with the measured contact force as input, assuming the contact to be punctual and performed on a known vertex on the surface of the mesh. The flat rigid surface exerts penalty elastic forces on the vertices in contact with it, determined through classical collision detection techniques. This model of the scene is implemented in the open-source SOFA simulator [27]. Given a preliminary ICP-based rigid alignment between the mesh and the segmented point cloud provided by the RGBD camera, a fitting function  $e_{param}$ , as a deviation between the simulation and the observed point cloud, can be defined as:

$$e_{param}(E, \nu) = dist(sim(E, \nu, \mathbf{f}^m(\mathbf{x}_c)), Y) \quad (3)$$

$\mathbf{f}^m$  is the measured contact force exerted on the known object contact point  $\mathbf{x}_c$ , while  $Y$  is the acquired point cloud. This deviation is shown in Figure 2. For a pertinent error function, we employ the matching technique presented in [26] between the segmented point cloud and the visible part of the deformed mesh, and conversely. Based on the two sets of mesh-to-point cloud and point cloud-to-mesh correspondences, the error function is calculated as:

$$e_{param}(E, \nu) = \frac{1}{n_{X_V}} \sum_{i=0}^{n_{X_V}} (\mathbf{x}_i^{sim} - \text{NN}_Y(\mathbf{x}_i^{sim}))^2 + \frac{\alpha}{n_Y} \sum_{j=0}^{n_Y} (\mathbf{y}_j - \text{NN}_{X_V}(\mathbf{y}_j))^2 \quad (4)$$

where  $X_V = \{\mathbf{x}_i^{sim}\}_{i=0}^{n_{X_V}}$  are the vertices of the visible part of the mesh which is deformed by simulation.  $Y = \{\mathbf{y}_j\}_{j=0}^{n_Y}$  are the vertices of the segmented point cloud, and  $\alpha$  is a

tuning parameter between both distances.  $\text{NN}_Y(\mathbf{x}_i^{\text{sim}})$  and  $\text{NN}_{X_V}(\mathbf{y}_j)$  define the correspondences between  $\mathbf{x}_i^{\text{sim}}$  and  $\mathbf{y}_j$ , using nearest neighbor searches respectively within the sets  $Y$  and  $X_V$ .

Minimizing  $e_{\text{param}}$  is a non-linear optimization problem, which we propose to solve using a gradient-free Nelder-Mead method [28] with respect to  $(E, \nu)$ . The results of the estimation for the considered application are represented in Fig. 3, converging respectively towards  $7489\text{N/m}$  and  $0.05$  for the Young modulus and the Poisson ratio.

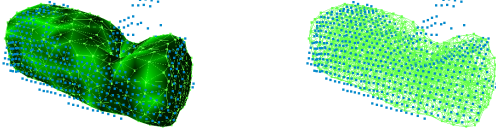


Fig. 2. Simulated (green mesh and vertices) and observed deformation (point cloud represented by blue dots).

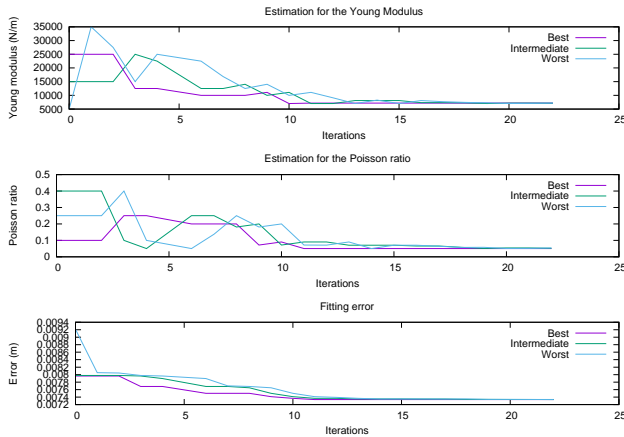


Fig. 3. Nelder-Mead process for elasticity parameters estimation: Young modulus (top), Poisson ratio (middle). Starting from 3 initial configurations, the convergence of the best, intermediate, and worst candidates, given the fitting error (bottom), is shown.

### III. INVERSE SIMULATION FOR DEFORMATION CONTROL

As introduced in section II, our problem, which consists in computing the contact forces that deform the object towards a known desired deformation, can be seen as an inverse simulation problem.

The framework described in [21] is thus employed by replacing the actuators with the contact forces exerted by the fingertips, assuming that the contacts are permanent on the object. As suggested in equation (1), Lagrange multipliers are used to model the contact force intensities  $\lambda_a$  that the fingertips apply on the object.

In this application linear actuation is considered, thus the directions  $\mathbf{H}_a$  of the forces are known, each actuator applying a force in only one direction with respect to a reference frame chosen at the contact. Here the directions

are simply the normals to the object surface at the contact points.

Regarding the manipulating hand, the range of efforts and the displacement limits that the fingers can apply to the object depend on the hand actuation system and the kinematics.

When controlling whole set of vertices of the mesh modelling the object, reaching a desired deformation can be intractable. In order to reduce the dimension of the problem, a set of end-effectors (or control) points are selected on the object. The objective is then to minimize the shift  $\delta_e$  between the end-effector positions and the desired positions  $\mathbf{p}_{des}$ , with respect to the force  $\lambda_a$  or the motion  $\delta_a$  applied at the fingertips at the contact points.

This can be formulated as a convex optimization problem solving  $\min(\frac{1}{2}\delta_e^T \delta_e)$ , which depends on the stiffness  $\mathbf{K}$ .  $\mathbf{K}$  is often large so an optimization in the motion space would be computationally expensive. To perform this optimization in real-time, the problem is projected in the constraint space using the Schur complement. The deviations  $\delta$  can then be written as:

$$\delta_i = h^2 \mathbf{H}_i \mathbf{K}^{-1} \mathbf{H}_j^T \lambda_j + \delta_j^{free} \quad (5)$$

where  $i$  and  $j$  can either represent the actuators  $a$  (the fingertips) or the end-effector  $e$ ,  $\delta_j^{free}$  being the deviation when no load is applied by the fingertips on the object.  $h$  is the integration time interval.

Based on equation (5),  $\frac{1}{2}\delta_e^T \delta_e$  is then minimized using QP programming, given the limit courses of the fingertips. Through this minimization technique, a solution for  $\lambda_a$  can be found, even when the desired position is out of the workspace. In such a case, the algorithm will find the point that minimizes the distance with the desired position while respecting the limits introduced for the stroke of the fingers of the hand. The motion  $\delta_a$  of the fingertips can finally be obtained from  $\lambda_a$ , using equation (5).

Let us note that for a potential speed-up, a multi-rate strategy can be used, combining two loops, one at a low frequency (FEM computations) and another at a higher frequency (around 600Hz, for QP resolution and actuator control) [29]

For a complete description of this framework, the reader can refer to [21], [22].

In our application, we choose three end-effectors, represented by green squares in Fig. 4. The output (actuation efforts at the fingertips at the contact points with the object) are computed with respect to the target input, both labeled in the Figure.

### IV. DEXTEROUS MANIPULATION WITH AN ANTHROPOMORPHIC HAND

The manipulation of a cylindrical object grasped by an underactuated robotic hand is performed to test the deformation control strategy. The SCHUNK 5-Finger Hand (S5FH) [30] is an anthropomorphic underactuated hand with size, kinematic structure and cosmetic appearance very similar to the human hand. The S5FH has 20 joints and 9 DOFs. The

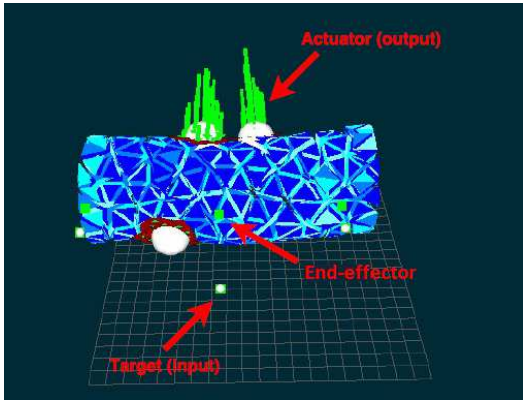


Fig. 4. The scene in SOFA.

whole technical data, hardware and software specifications can be found in [30] and [31]. The number of motors is significantly lower than the number of joints, thus joint motion couplings are regulated by means of mechanical synergies defined via mechanical transmissions.

Let  $q$  be the vector of the 20 joint angles describing the robotic hand configuration and  $x_e$  the Cartesian position of the fingertips. The differential kinematics between the mechanical synergies subspace and the Cartesian space, is represented by  $\dot{x}_e = J_{h_m} \dot{m}$ , where  $J_{h_m} \in \mathbb{R}^{n_x \times n_m}$  is the mechanical synergies Jacobian and is computed as  $J_{h_m} = J_h S_m$  such that  $\dot{x}_e = J_h S_m \dot{m} = J_h \dot{q}$ , where  $x_e \in \mathbb{R}^{n_x}$ , with  $n_x = 15$ , is the position vector of the five fingertips,  $J_h \in \mathbb{R}^{n_x \times n_q}$  is the S5FH hand Jacobian.  $S_m \in \mathbb{R}^{n_q \times n_m}$  is the matrix of the mechanical synergies and maps motor velocities into joint velocities,  $S_m \dot{m} = \dot{q}$ . In the manipulation control the Schunk hand motors are actuated according to the schema in Fig. 6.

The desired deformation is generated by moving the end-effectors, represented in Fig. 4, interactively in the GUI of SOFA or through a task planning module providing predefined trajectories. According to this scheme, from the

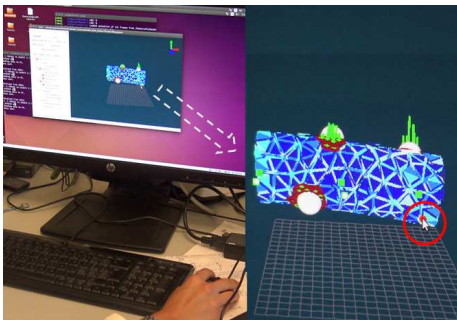


Fig. 5. Desired deformation assigned by the user using the graphic interface.

defined desired deformation the FEM inverse problem can be solved. The solution of the problem, namely the  $\lambda_a$  actuators effort to apply, is translated in the desired fingertip positions of the robotic hand. The hand control is performed in the Cartesian space with a closed-loop inverse kinematic

algorithm (CLIK) that uses the transpose of the mechanical synergies Jacobian [32].

Underactuation implies solving an inverse kinematics problem where the Jacobian is a high-rectangular matrix, which is typical when the number of equations is greater than the number of the unknown variables. In this case, the solution to the inverse problem does not exist if the position vector of the fingertips  $x_e \notin \mathcal{R}(J_{h_m})$ , but only an approximate solution can be found. Indeed, underactuation reduces the subspace dimension of controllable internal forces.

Let  $E$  be a basis for the subspace of controllable internal forces that maps the controlled joint displacements into internal forces activated on the object. According to the mechanical synergies matrix,  $E$  is reduced to  $E_m = ES_m$ .

Thus, if we define  $e$  as the deformation error between the desired deformation  $p_{des}$  defined by the operator and the measured one on the real object, this error accounts for three different sources: convex optimization  $e_{com}$ , model parameters approximations  $e_{mod}$ , and hardware features (underactuation and sensors uncertainties)  $e_{hand}$ , such as:

$$e = e_{com} + e_{mod} + e_{hand}. \quad (6)$$

The deformation control is here an open-loop action. Nevertheless, we minimize the modeling errors using a parameter estimation strategy for  $(E, \nu)$ , and the hand mechanical limitation using a feedback of the measured motor positions. Indeed, the hand control is a closed loop strategy that minimize the error due to the underactuation. The results shown in the next section demonstrate that the whole scheme works well in real-time and that the achieved results are quite promising.

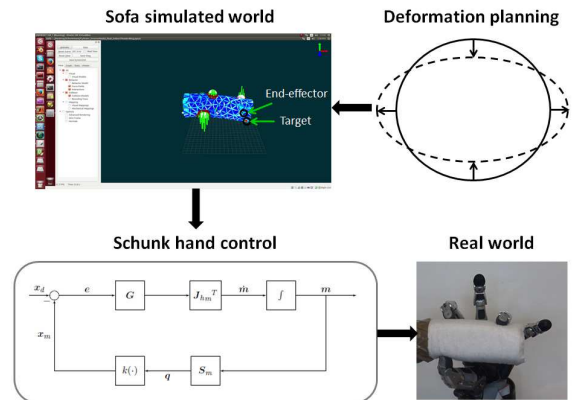


Fig. 6. Deformation control scheme.

## V. EXPERIMENTS

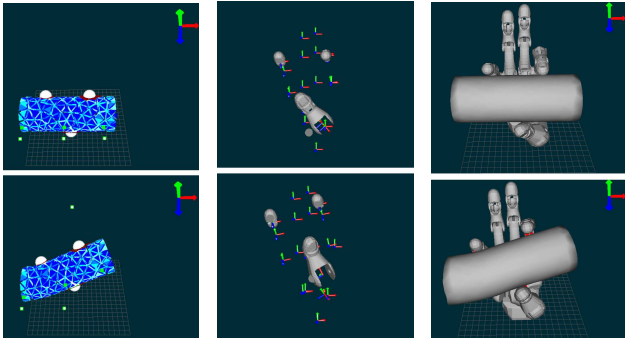
### A. In simulation

The proposed approach can be illustrated in a pure simulation scenario. A result of the inverse simulation method, giving the forces and displacements of the fingertips at the contact points, in order to reach desired displacements of the 3 end-effector points, can be seen in Fig.7(a).

In order to simulate the interaction between the anthropomorphic robotic hand and the deformable object, we

have integrated the hand into SOFA, which is used here to implement both direct and inverse FEM problems. The hand kinematics have been modelled using the real finger dimensions, joint ranges and kinematic dependencies. Based on the computed displacements of the fingertips, the inverse kinematics provide the joint displacements of the hand, as shown in Fig. 7(b). The origin of the reference frame in the scene is chosen at the wrist center of the hand.

In order to reproduce the desired deformation in direct simulation, we model the collisions with the object by using the CAD model of each part of the hand and classical collision detection and penalty response techniques available in SOFA, as seen in Fig. 7(c).



(a) Inverse simulation. (b) Inverse kinematics. (c) Direct simulation.

Fig. 7. In these figures the three steps of the deformation control algorithm are simulated in SOFA.

### B. Real world manipulation

In the real scenario with the S5FH hand, a socket based communication protocol is carried out to interface the inverse simulation output with the SCHUNK SVH driver. At each time step the inverse simulation computes the desired contact points displacements ( $\delta_a$ ) on the deformable object (according to the estimated actuator efforts  $\lambda_a$ ), which are then transmitted to the hand controller through the Robot Operating System (ROS). The Robot Operating System (ROS) is used to control the SCHUNK 5-Finger Hand. A SVH Driver suite has been developed by FZI Forschungszentrum Informatik for the low level interface and enables an easy control of the hand using a customized library written in C++ [31].

The deformation control scheme has been tested for two different groups of fingers, thumb-index-middle and thumb-index-ring. To compare the desired and the actual deformation of the manipulated object we have measured the fingertips error between the desired position provided by SOFA and the one computed from direct kinematics. For the configuration thumb-index-middle the results are reported in Fig. 8. The fingers behave as three actuators that produce deformations. The initial position of the actuators in SOFA are planned taking into account the hand kinematics limitations, thus in such a way that the position of the desired fingertips are in the workspace of the hand. Despite this choice, a constant centroid error, see Fig. 8, is present at the beginning of the experiments.

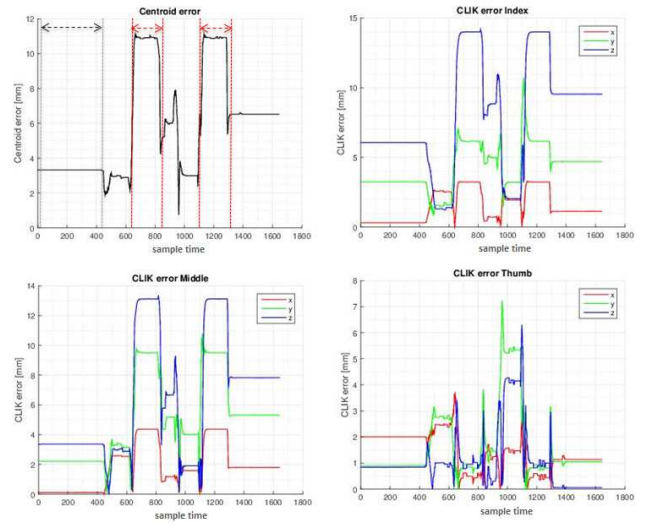


Fig. 8. Hand configuration with thumb-index-middle fingers in contact: the centroid error and the error of the single fingers are reported.

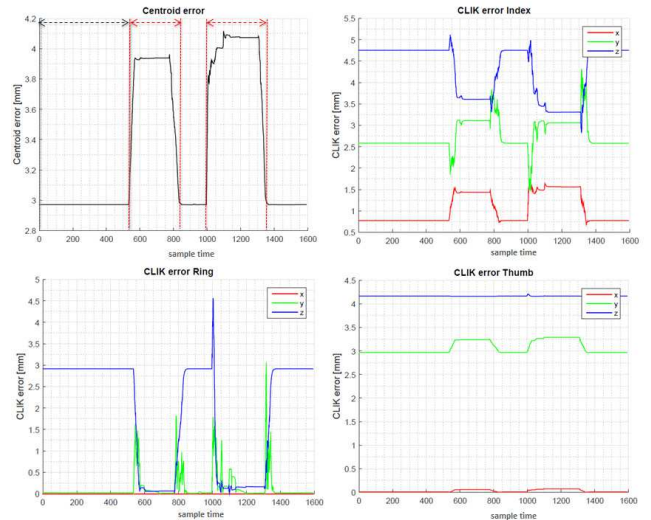


Fig. 9. Hand configuration with a fixed thumb and middle-ring fingers in contact: the centroid error and the error of the single fingers are reported.

This error is due to approximations introduced by underactuation and sensors uncertainties that can cause a different kinematic position reached by the hand with respect to the planned one. The error is constant because at the beginning the hand is in contact with the object but is not moving. The red arrows highlight the two time slots when a target is assigned by the operator for one of the end-effectors. As it is possible to observe, the error increases due to the fact that the desired fingertip positions provided by SOFA are not in the hand workspace, thus the CLIK algorithm provides a minimized error solution. To provide an idea on how underactuation can influence the error, we have reported the experimental setup obtained with another hand configuration using the same number of contacts (three fingers) but with a fixed thumb. Fig. 9 reveals that the overall error is smaller because of the fixed thumb that reduces the error due to hand

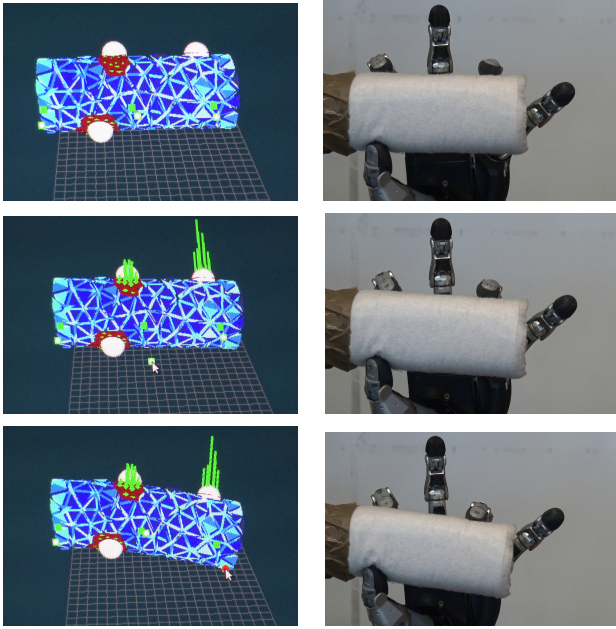


Fig. 10. A comparison between the desired deformation planned in the sofa scene and the deformation controlled on the real object is shown.

underactuation.

A comparison between the desired configuration in simulation, and the real world scene, in the initial undeformed configuration and in two configurations where a desired deformation is imposed are reported in Fig. 10.

In order to have a quantitative evaluation of the sources of error, as described in equation (6), the displacements of the control points on the real object have been measured by tracking two markers with a camera (Fig. 11), using a ROS wrapper for Alvar, an open source AR tag tracking library.

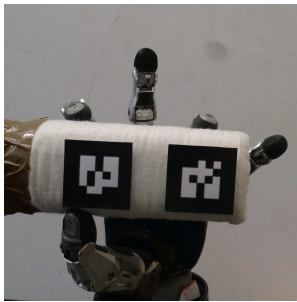


Fig. 11. Setup for the deformation tracking.

This way we can estimate the error between the displacements of the control points in the simulated framework and on the object in the real scenario. The results obtained with the thumb-index-ring configuration are reported in Fig. 12. A desired displacement is imposed by the operator on one of the end-effectors by gradually dragging the goal with the mouse in the SOFA GUI.

As previously outlined in Sect. IV, this error is the sum of three sources: the error between the positions reached

by the end-effectors (blue line in Fig. 12) and the goal positions (green line in Fig. 12) which is caused by the convex optimization  $e_{com}$ , and other two errors,  $e_{mod}$  and  $e_{hand}$ , caused by the differences between the simulated deformations (blue line in Fig. 12) and real deformations on the object (red line in Fig. 12).

In Fig. 13 the error due to the optimization in FEM inverse problem is shown. In Fig. 14 the error between the real and simulated deformation is reported. This error is given by the approximation of the physical parameters of the model and of the hand mechanical structure, that includes underactuation and position errors, mainly due to sensor uncertainties.

According to the plots represented in Figs. 8-9 and Figs. 14-13 a significant error between the simulation and the real object is caused by the limitations of the hand, due to underactuation.

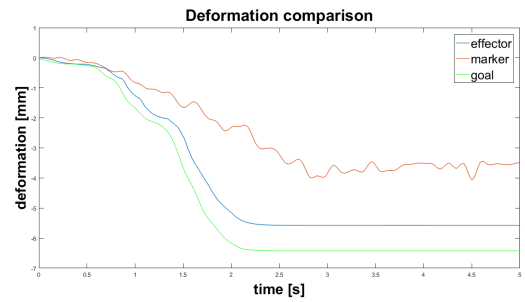


Fig. 12. A comparison between the deformations of the control point on the real object, on the simulated object in SOFA (end-effector) and the target defined by the operator is reported.

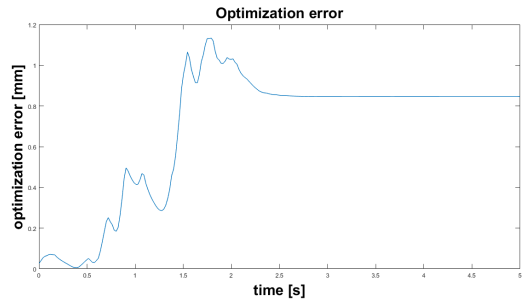


Fig. 13. The optimization error is represented.

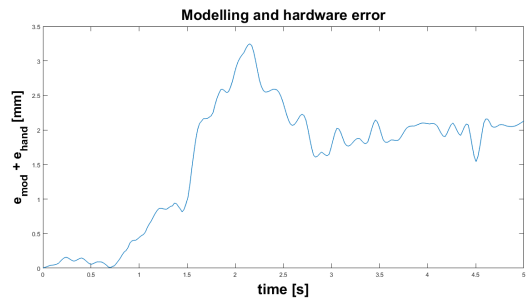


Fig. 14. The error due to underactuation, hand kinematic model uncertainties, error on sensor measurements and on the object model is reported.

## VI. CONCLUSION AND FUTURE WORKS

In this paper, a deformation control strategy for a soft object has been applied to an anthropomorphic robotic hand. We use an FEM method for object modeling and a unified representation based on Lagrange multipliers for contact force modeling. An online inverse simulation technique can then be employed to compute the proper contact forces at the hand fingertips to achieve a desired deformation, through the displacement of a set of selected end-effector points on the object. A preliminary step for the estimation of the elasticity parameters is carried out, based on the minimization of a fitting error between simulated and observed deformations, using an RGB-D camera. Some first results are presented using the SCHUNK 5-Finger Hand (S5FH) and a cylindrical object. The performance of this open-loop control framework is measured by means of the error between the displacements of the selected points on the real object and on the simulated object. Future works will improve the inverse simulation and the deformation control using vision feedback.

## REFERENCES

- [1] Z. Wang and S. Hirai, "Modeling and estimation of rheological properties of food products for manufacturing simulations," *Journal of food engineering*, vol. 102, no. 2, pp. 136–144, 2011.
- [2] T. Bretl and Z. McCarthy, "Quasi-static manipulation of a kirchhoff elastic rod based on a geometric analysis of equilibrium configurations," *The International Journal of Robotics Research*, vol. 33, no. 1, pp. 48–68, 2014.
- [3] S. Miller, J. Van Den Berg, M. Fritz, T. Darrell, K. Goldberg, and P. Abbeel, "A geometric approach to robotic laundry folding," *The International Journal of Robotics Research*, vol. 31, no. 2, pp. 249–267, 2012.
- [4] S. Patil and R. Alterovitz, "Toward automated tissue retraction in robot-assisted surgery," in *ICRA*, 2010, pp. 2088–2094.
- [5] D. Navarro-Alarcon and Y.-h. Liu, "Uncalibrated vision-based deformation control of compliant objects with online estimation of the jacobian matrix," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, 2013, pp. 4977–4982.
- [6] D. Navarro-Alarcon, Y.-h. Liu, J. G. Romero, and P. Li, "On the visual deformation serving of compliant objects: Uncalibrated control methods and experiments," *The International Journal of Robotics Research*, vol. 33, no. 11, pp. 1462–1480, 2014.
- [7] 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.
- [8] J. Smolen and A. Patriciu, "Deformation planning for robotic soft tissue manipulation," in *Advances in Computer-Human Interactions, 2009. ACHI'09. Second International Conferences on*. IEEE, 2009, pp. 199–204.
- [9] D. Berenson, "Manipulation of deformable objects without modeling and simulating deformation," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, 2013, pp. 4525–4532.
- [10] S. Patil *et al.*, "Motion planning under uncertainty in highly deformable environments," *Robotics science and systems: online proceedings*, 2011.
- [11] Y. Li, Y. Yue, D. Xu, E. Grinspun, and P. K. Allen, "Folding deformable objects using predictive simulation and trajectory optimization," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE, 2015, pp. 6000–6006.
- [12] J. Das and N. Sarkar, "Passivity-based target manipulation inside a deformable object by a robotic system with noncollocated feedback," *Advanced Robotics*, vol. 27, no. 11, pp. 861–875, 2013.
- [13] J. Schulman, J. Ho, C. Lee, and P. Abbeel, "Generalization in robotic manipulation through the use of non-rigid registration," in *Proceedings of the 16th International Symposium on Robotics Research (ISRR)*, 2013.
- [14] D. Hadfield-Menell, A. X. Lee, C. Finn, E. Tzeng, S. Huang, and P. Abbeel, "Beyond lowest-warping cost action selection in trajectory transfer," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 3231–3238.
- [15] T. Tang, C. Liu, W. Chen, and M. Tomizuka, "Robotic manipulation of deformable objects by tangent space mapping and non-rigid registration," in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*. IEEE, 2016, pp. 2689–2696.
- [16] P.-C. Yang, K. Sasaki, K. Suzuki, K. Kase, S. Sugano, and T. Ogata, "Repeatable folding task by humanoid robot worker using deep learning," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 397–403, 2017.
- [17] 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.
- [18] T. Yoshikawa, M. Koeda, and H. Fukuchi, "Grasping and manipulation by soft-fingered robot hands," in *Proc. of the 13th International Conference on Advanced Robotics*, 2007.
- [19] R. Deimel and O. Brock, "A novel type of compliant and underactuated robotic hand for dexterous grasping," *The International Journal of Robotics Research*, vol. 35, no. 1-3, pp. 161–185, 2016.
- [20] M. Tavakoli and A. T. de Almeida, "Adaptive under-actuated anthropomorphic hand: Isr-softhand," in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*. IEEE, 2014, pp. 1629–1634.
- [21] E. Coevoet, T. Morales-Bieze, F. Largilliere, Z. Zhang, M. Thieffry, M. Sanz-Lopez, B. Carrez, D. Marchal, O. Goury, J. Dequidt *et al.*, "Software toolkit for modeling, simulation, and control of soft robots," *Advanced Robotics*, vol. 31, no. 22, pp. 1208–1224, 2017.
- [22] C. Duriez, E. Coevoet, F. Largilliere, T. Morales-Bieze, Z. Zhang, M. Sanz-Lopez, B. Carrez, D. Marchal, O. Goury, and J. Dequidt, "Framework for online simulation of soft robots with optimization-based inverse model," in *Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR), IEEE International Conference on*. IEEE, 2016, pp. 111–118.
- [23] 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.
- [24] R. D. Cook, *Finite element modeling for stress analysis*. Wiley, 1994.
- [25] M. Müller and M. Gross, "Interactive virtual materials," in *Graphics Interface*, 2004, pp. 239–246.
- [26] A. Petit, V. Lippiello, and B. Siciliano, "Real-time tracking of 3D elastic objects with an RGB-D sensor," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2015, pp. 3914–3921.
- [27] F. Faure, C. Duriez, H. Delingette, J. Allard, B. Gilles, S. Marchesseau, H. Talbot, H. Courtecuisse, G. Bousquet, I. Peterlik *et al.*, "Sofa: A multi-model framework for interactive physical simulation," in *Soft Tissue Biomechanical Modeling for Computer Assisted Surgery*. Springer, 2012, pp. 283–321.
- [28] J. Nelder and R. Mead, "A simplex method for function minimization," *The Computer Journal*, vol. 7, no. 4, pp. 308–313, Jan. 1965.
- [29] F. Largilliere, V. Verona, E. Coevoet, M. Sanz-Lopez, J. Dequidt, and C. Duriez, "Real-time control of soft-robots using asynchronous finite element modeling," in *IEEE International Conference on Robotics and Automation*, 2015, pp. 2550–2555.
- [30] "Schunk hand webpage," <http://mobile.schunk-microsite.com/en/produkte/produkte/servoelektrische-5-finger-greifhand-svh.html>.
- [31] "Schunk SDK," [http://wiki.ros.org/schunk\\_svh\\_driver](http://wiki.ros.org/schunk_svh_driver).
- [32] F. Ficuciello, "Synergy-based control of underactuated anthropomorphic hands," *IEEE Transactions on Industrial Informatics*, 2018.
- [33] J. Allard, S. Cotin, F. Faure, P. Bensoussan, F. Poyer, C. Duriez, H. Delingette, and L. Grisoni, "Sofa-an open source framework for medical simulation," in *MMVR 15-Medicine Meets Virtual Reality*, vol. 125. IOP Press, 2007, pp. 13–18.