Adaptive Bayesian optimization for robotic pushing of thin fragile deformable objects

Rafael Herguedas[1,3], Ashok M. Sundaram[2], Gonzalo López-Nicolás[1], Máximo A. Roa[2], and Carlos Sagüés[1]

[1] Instituto de Investigación en Ingeniería de Aragón, Universidad de Zaragoza, Spain,
rherguedas@unizar.es, gonlopez@unizar.es, csagues@unizar.es,
[2] Institute of Robotics and Mechatronics, German Aerospace Center (DLR), 82234 Wessling, Germany,
Ashok.MeenakshiSundaram@dlr.de, Maximo.Roa@dlr.de,
[3] Instituto Tecnológico de Aragón, Zaragoza, Spain,
rherguedas@itainnova.es

Abstract. Robotic manipulation of deformable objects is challenging due to the great variety of materials and shapes. This task is even more complex when the object is also fragile, and the allowed amount of deformation needs to be constrained. For the goal of driving a thin fragile deformable object to a target 2D position and orientation, we propose a manipulation method based on executing planar pushing actions on the object edges with a robotic arm. Firstly, we obtain a probabilistic model through Gaussian process regression, which represents the time-varying deformation properties of the system. Then, we exploit the model in the framework of an Adaptive Bayesian Optimization (ABO) algorithm to compute the pushing action at each instant. We evaluate our proposal in simulation.

Keywords: Manipulation of deformable objects, robotic pushing, Bayesian optimization

1 Introduction

Deformable objects are very common in a number of tasks and processes. Ranging from clothes to food products, this kind of materials have a considerable impact in the global economy. Automatizing processes that require deformable object manipulation is interesting for different reasons (process efficiency, workers safety, assistance to people with disabilities, etc.), but it is also technically challenging [27]. For domestic and industrial manipulation tasks, perception, modeling and control of these objects are some of the main issues to tackle, and different solutions have been proposed and evaluated [23], [26].

Particularly, the topic of manipulation of fragile deformable objects is an open problem that has received increased attention in the last years, with solutions for grasping delicate objects by exploiting the environmental constraints [25], for cleaning deformable parts with fragile sections [15] and for vision-based object-compliant shape control [8], among others. For such fragile materials, the control actions must be limited and the manipulation strategy must be carefully designed, so that their integrity is not compromised in the process. Some examples of deformable fragile objects are fish and meat portions, food dough, unfired ceramic pieces and plastic parts at high temperatures.

Here, we propose a new method to manipulate thin fragile deformable objects based on a sequence of planar pushing actions. These actions aim at driving the object to a specific configuration (position in 2D and orientation around the vertical axis), while preserving its shape within an admissible range of deformation. In contrast to the standard prehensile manipulation of rigid or semi-rigid objects, pushing is less invasive and, therefore, more suitable for manipulating thin fragile objects that can be damaged when grasped with robotic tools. In this problem, the interaction forces between the supporting medium and the object must be taken into account. Besides, the complexity of the problem is increased by the fact that we consider the mechanical properties of the objects to be unknown.

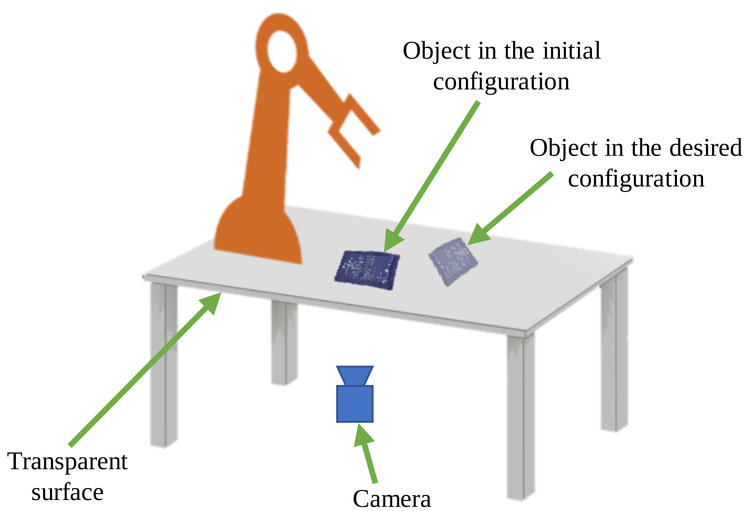

Fig. 1. System overview: A fragile thin deformable object is pushed by a robot manipulator to a desired configuration with compliant-object actions that respect its integrity. The input information (object contour) to the Adaptive Bayesian Optimization-based method is provided by a camera under the transparent surface that supports the object.

Manipulation of rigid objects by pushing is a mature topic in robotics [24]. A key work is based on Variational Heteroscedastic Gaussian processes to model planar pushing interaction [4]. An uncalibrated image-based controller is considered in another study for parking objects by pushing [16]. The problem of pushing objects with unknown center of mass is tackled in a recent work by finding a suitable two-edge-contact configuration [10]. However, the number of solutions for robotic non-prehensile manipulation of deformable objects is still limited. Some works in this context include the results of the RoDyMan project, focused on dynamic non-prehensile manipulation of rigid and deformable objects with robotic platforms [22]. Modeling the deformation of an object under pushing has been addressed by offline learning with integration of visual and force information [3]. Different studies tackle non-fixed contact manipulation configurations by means of optimization-based methods for contact adjustment [13] and contact point selection [11]. Recently, a strategy for object rearrangement based on planar pushing has been analyzed and validated [5]. In comparison with our method, these works only consider partially and not fully deformable objects.

Our proposed method is based on a Bayesian Optimization algorithm. This kind of methods are focused on finding the optimum value of an unknown function, in a small number of iterations. In each iteration of the algorithm, a probabilistic model (typically a Gaussian Process) is updated through Bayesian inference, with data selected by means of an acquisition function. In contrast to other alternative methods like Reinforcement Learning, the amount of data needed to create a behavioral model is much smaller. The pioneer work in [9] solves the problem of robotic motion planning in environments with deformable objects by means of Gaussian Process Regression. Safe and robust robotic grasping is tackled in another study by haptic exploration and unscented Bayesian optimization [19]. With respect to the manipulation of deformable objects, a solution has been proposed based on servo control and a deformation model learned with fast online Gaussian process regression [12]. Another study considers Bayesian inference and deep learning for solving real-to-sim problems of deformable objects manipulation [2]. Other applications of Bayesian optimization include non-rigid structure from motion [1] and soft landing control of reluctance actuators [17].

The main contribution of our study, in comparison with previous works, is applying the technique of sequential pushing to fully deformable objects, with pushing actions that contain the deformation of the object within an admissible range. In addition, our algorithm leverages Adaptive Bayesian Optimization and takes into account the temporal evolution of the object, thanks to a probabilistic model. Then, after every pushing action the object is deformed, and this is accounted by the learnt policy to drive the object to a planar goal configuration.

2 Manipulation task overview

The manipulation setup is equipped with a robotic arm and consists of a flat horizontal surface in which a thin fragile deformable object lies (see Fig. 1). A camera is placed in the environment so that the 2D contour of the object is detected without occlusions. In this case, the object contour corresponds to the projection in the horizontal surface along the vertical axis, and it is defined by the ordered sequence of M contour points $\mathbf{V}_c = [\mathbf{v}_{c1}, \mathbf{v}_{c2}, ..., \mathbf{v}_{cM}] \in \mathbb{R}^{2\times M}$. We consider that the state of the object is determined by the contour centroid

$\mathbf{c} \in \mathbb{R}^2$, the orientation of the contour around the vertical axis $\theta \in [0, 2\pi)$ and the shape of the contour represented as a binary mask $\boldsymbol{\gamma}$.

The goal is to perform object-compliant actions that transport and rotate the object to a goal position. Due to the thin and fragile structure of the object, it cannot be grasped and lifted, and its deformation must be bounded along the process. We define three different errors to assess the task completion:

$$e_c = \|(\mathbf{c} - \mathbf{c}_d)\| / \|\mathbf{c}_d\| \ , \tag{1}$$

$$e_\theta = (\theta - \theta_d)/\pi \ , \tag{2}$$

$$e_\gamma = \Sigma(\boldsymbol{\gamma} \oplus \boldsymbol{\gamma}_d)/\Sigma\, \boldsymbol{\gamma}_d \ , \tag{3}$$

where $\|\cdot\|$ is the 2-norm operator and the parameters with d subscript refer to the user-defined target values. The contour centroid is computed as $\mathbf{c} = \frac{1}{M}\mathbf{V}_c\mathbf{1}_M$, with $\mathbf{1}_M$ being a column vector of M ones. We apply Principal Component Analysis (PCA) to the set $\mathbf{V}_b = \mathbf{V}_c - \mathbf{c}\,\mathbf{1}_M^\intercal = \mathbf{U}\mathbf{S}\mathbf{V}^\intercal \in \mathbb{R}^{2\times M}$, and we obtain the orientation of the contour as $\theta = \mathrm{atan}(V_{2,1}, V_{1,1})$, where the subscripts refer to the corresponding elements of $\mathbf{V}$. Symmetry issues are avoided by checking the temporal consistency of θ.

Equation (3) represents a ratio, where the numerator corresponds to the difference in pixels between the current $\boldsymbol{\gamma}$ and the desired $\boldsymbol{\gamma}_d$ shapes of the object. $\boldsymbol{\gamma}$ is the binary mask of $\mathbf{R}(-\theta)\,\mathbf{V}_b$, where $\mathbf{R}(-\theta)$ is the 2D rotation matrix of $-\theta$ radians, and $\boldsymbol{\gamma}_d$ is the binary mask of the desired contour with centroid at the origin and whose principal direction is aligned with the x axis. The difference is obtained as the summation of the non-zero pixels resulting from the boolean XOR operation ($\oplus$ symbol) between $\boldsymbol{\gamma}$ and $\boldsymbol{\gamma}_d$. In turn, the denominator is the sum of non-zero pixels of the desired shape ($\boldsymbol{\gamma}_d$). Note that γ_d is not necessarily an exact shape to seek, but a mask in which the deformation of the object must be contained.

Then, the total task error is

$$e_T = k_c\, e_c + k_\theta\, |e_\theta| + k_\gamma\, e_\gamma \ , \tag{4}$$

where k_c, k_θ and k_γ are positive weights and $|\cdot|$ is the absolute value operator. Note that $e_T \geq 0$.

The method we propose for completing the manipulation task consists in executing a sequence of pushing actions $\mathbf{u} = (\kappa, d)$ via the end-effector of the arm. Parameter $d \in [d_{min}, d_{max}]$ stands for the pushing distance, which is always applied perpendicularly to the direction of the contour segment. $\kappa \in [0, 1]$ is defined as the object contour ratio. The contour ratio is a fraction of the effective contour $\mathbf{V}_e \subset \mathbf{V}_s$, where $\mathbf{V}_s$ is the augmented contour with

$$\mathbf{v}_{si} = \mathbf{v}_i + d_o\, \mathbf{v}_{bi}/\|\mathbf{v}_{bi}\| \ , \tag{5}$$

where $\mathbf{v}_{bi} = \mathbf{v}_i - \mathbf{c}$ and d_o is a safety offset to take into account the width of the pushing tool. Therefore, κ represents the position of the effective contour between the first effective point $\mathrm{v}_{e1}(\kappa = 0)$ and the last point $\mathrm{v}_{eM_e}(\kappa = 1)$, where M_e is the number of points of the effective contour. The first and the last effective points are selected manually for each object. The pushing actions configured with these parameters aim at reducing the task error e_T over time, while preserving the integrity of the object.

3 Optimization problem

We can express the manipulation task at the time instant k as the following optimization problem:

$$\begin{aligned} \text{Given} \quad & \mathbf{V}_c,\ \mathbf{c}_d,\ \theta_d,\ \boldsymbol{\gamma}_d,\ f(\mathbf{x}) \\ \underset{\mathbf{u}}{\text{minimize}} \quad & \xi = e_{T(k+1)}(f(\mathbf{x})) - e_{Tk} \\ \text{subject to:} \quad & \kappa_{min} \le \kappa \le \kappa_{max}\ , \\ & d_{min} \le d \le d_{max}\ , \end{aligned} \tag{6}$$

where $\mathbf{x} = (\mathbf{u}, t) = (\kappa, d, t) \in \mathbb{R}^3$ and $f(\mathbf{x})$ is an unknown function that maps κ and d with e_T, over time. Different physical effects are integrated in $f(\mathbf{x})$. The ones with greater impact on the development of the task are the deformation modes of the object and the friction between the object and the manipulation surface. In general, fragile deformable objects may show deformations that disappear when the acting force is released (elastic) or deformations that remain afterwards (plastic). Besides, the manipulation actions may create compression, traction, shear or torsion among other deformation modes. With respect to friction, we can differentiate between static friction, which happens when the object is not globally in motion, and dynamic friction, which acts during global motion.

The transition between these modes determines the difference between actions that induce local motions only (deformations) and the actions that produce both local and global motions (deformation and displacement). These effects and their interactions are complex to model, and their variability is high. This is the reason why obtaining $f(\mathbf{x})$ for each manipulation case seems appropriate. The method we propose is based on the Adaptive Bayesian Optimization (ABO) technique [20]. While a standard Bayesian Optimization (BO) approach allows obtaining the global optimum of an unknown time-invariant function, ABO allows to track the minimum of a time-varying function over time. This method can be divided in two main steps:

1. *Data acquisition.* At the beginning of the process, the parameter space (delimited by the constraints) is randomly sampled for obtaining an initial model. Then, the acquisition process modulates to sampling points that raise the highest probability of improvement.
2. *Model update.* The model is updated with new data after every sampling. This allows refining the knowledge about the underlying physics and the evolution of the system over time.

In the present case, sampling requires pushing and detecting the contour afterwards. Pushing modifies not only the state of the object, but also its deformation and friction properties, which may vary due to local stiffening, creasing and other effects. Then, solving (6) implies finding, at every instant k, the action that most decreases the error with respect to the previous instant, and satisfies the existing constraints. Therefore, the more negative ξ becomes, the more effective a pushing action will be.

4 Gaussian process regression

The method we propose requires a prior regression model. Under the assumption that the variability of the unknown model of our system, in space and time, can be predicted by a Gaussian probability function, we consider a spatiotemporal Gaussian Process (GP) as the required surrogate model. GPs are likely the most popular regression models for BO, and they represent the generalization of the Gaussian probability distribution [21]. This model is specified by a mean function $\mu_0(\mathbf{x})$ and a covariance function, or kernel, $k(\mathbf{x}, \mathbf{x}')$ as $f(\mathbf{x}) \sim \mathcal{GP}(\mu_0, k)$. We assume that $\mu_0 = 0$ for convenience, and we consider the kernel to be stationary, separable and of the form

$$k(\mathbf{x}, \mathbf{x}') = k_s(\mathbf{u}, \mathbf{u}') \cdot k_t(t, t') \ , \tag{7}$$

where k_s and k_t are the spatial and temporal kernel functions, respectively [20]. We consider kernel functions of the Matérn 52 form $k_{M52}(\mathbf{x}, \mathbf{x}')$:

$$k_{M52} = \sum_i \left(\sigma_f^2 + \frac{\sqrt{5}\|x_i - x_i'\|_2}{\rho_i} + \frac{5\|x_i - x_i'\|_2^2}{3\rho_i^2} \exp\left(-\frac{\sqrt{5}\|x_i - x_i'\|_2}{\rho_i} \right) \right) \ , \tag{8}$$

where σ_f^2 is the characteristic variance and ρ_i is the length scale of every dimension (with $i \equiv \kappa,\ d,\ t$).

For updating the model with the sets of S measurements $\mathbf{Y} = [\xi_1, \xi_2, ..., \xi_S] \in \mathbb{R}^{1\times S}$ and $\mathbf{X} = [\mathbf{x}_1^\intercal, \mathbf{x}_2^\intercal, ..., \mathbf{x}_S^\intercal] \in \mathbb{R}^{3\times S}$ we must compute the posterior Gaussian distribution

$$f|\mathbf{X}, \mathbf{Y}, \mathbf{x} = \mathcal{N}(\mu, \sigma) \ , \tag{9}$$

where μ and the variance σ^2 are computed with the Sherman-Morrison-Woodbury formula as

$$\mu = \mathbf{k}^\intercal \mathbf{K}^{-1} \mathbf{Y} \ , \tag{10}$$

$$\sigma^2 = k(\mathbf{x}, \mathbf{x}) - \mathbf{k}\mathbf{K}^{-1}\mathbf{k} \ . \tag{11}$$

$\mathbf{K} \in \mathbb{R}^{S\times S}$ is the covariance matrix with $\mathbf{K}_{ij} = k(\mathbf{X}_i, \mathbf{X}_j)$ and $\mathbf{k}$ is defined with $\mathbf{k}_i = k(\mathbf{X}_i, \mathbf{x})$, $\forall i, j \leq S$.

As mentioned before, we want to track the minimum of ξ over time. For efficiently getting a good estimate of this value, the variability of the solution search space must be represented well enough in the initial model. We apply the Latin Hypercube Sampling (LHS) technique [18], that allows covering the search domain in a near-random manner with S_0 initial samples. Algorithm 1 reports the main steps involved in getting the initial model.

Algorithm 1 Get initial model $f(\mathbf{x})$

Require: $S_0, \mathbf{V}_c, \mathbf{c}_d, \theta_d, \boldsymbol{\gamma}_d$

$\mathbf{U} \leftarrow \text{LHS}(S_0)$ # set of random pushing parameters
$S \leftarrow 0$
while $S < S_0$ **do**
  $\mathbf{u} \leftarrow \mathbf{U}(S)$
  $\mathbf{V}_c, \mathbf{X}, \mathbf{Y} \leftarrow applyPushingAction(\mathbf{u})$
  $S \leftarrow S + 1$
end while
return $f(\mathbf{x}) \leftarrow updateModel(\mathbf{X}, \mathbf{Y}), \mathbf{V}_c$

5 Acquisition function

One of the key aspects of ABO is that it allows tracking the minimum of $f(\mathbf{x})$ through successive trials or acquisitions where the value the search criterion yields is maximum. The acquisition step, in our case, should consider a balance between exploration and exploitation. Once the algorithm finds a suitable pushing action, this action should be exploited until its effectiveness decreases. In that moment, the algorithm should explore the solution search space to find a better pushing action candidate.

The dynamic acquisition function we consider is based on the Lower Confidence Bound (LCB) [7]:

$$\text{LCB}(\mathbf{x}) = \mu(\mathbf{x}) - w\sigma(\mathbf{x}) , \tag{12}$$

where $w \geq 0$ is a constant that regulates the confidence level. According to this function, the lower the LCB value the more promising a sample is. We look for a minimum LCB by optimizing the function under the problem constraints κ_{min}, κ_{max}, d_{min} and d_{max}. The Adam optimization algorithm [14] provides a suitable solution by means of exponential moving averages of the gradient and the squared gradient of the function. At every iteration of the method the search space boundaries are checked, and the evaluated point is saturated to the limits in case the constraints are violated. After a user-defined number of iterations, the best solution is selected as the candidate pushing action for the next time step.

It is worth mentioning that, in contrast to other applications of ABO, we do not optimize the time step in which we sample/push the system. This is due to the fact that the temporal evolution of the system is conditioned to the pushing step. Then, we are interested in applying the pushing actions as soon as possible, so that the operation time is minimized. We show the complete structure of our pushing method in Algorithm 2.

6 Simulation results

We evaluate our method by means of two different simulation scenarios in the Pybullet environment, a module in Python language focused on sim-to-real transfer

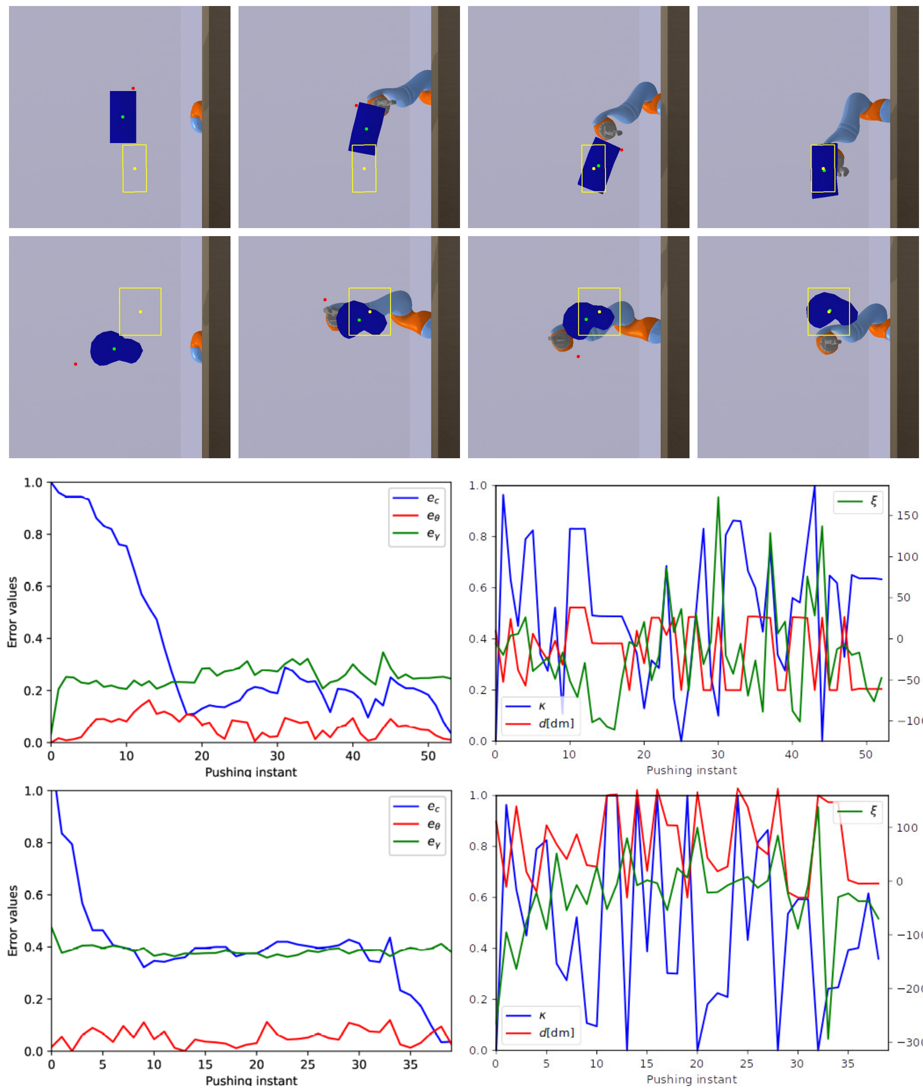

Fig. 2. The first two rows represent the sequence of four bottom-view snapshots, for two scenarios: a rectangular deformable object (first row) and a non-regular shaped deformable object (second row). The snapshots are acquired at $k = \{0, 10, 18, 53\}$ and $k = \{0, 10, 26, 39\}$, respectively. The deformable object, in blue, is pushed towards the target rectangle, in yellow. Note that the red dot, which indicates the next pushing point, is separated from the contour to compensate for the gripper width. The last two rows depict the errors and actions plots, for the first test scenario (top row) and second test scenario (bottom row).

Algorithm 2 Find optimal pushing action $\mathbf{u}$ at time t.

Require: $\mathbf{X}, \mathbf{Y}, \mathbf{V}_c, \mathbf{c}_d, \theta_d, \boldsymbol{\gamma}_d, \mu, \sigma$
$\mathbf{c} \leftarrow 1/M\,\mathbf{V}_c\mathbf{1}_M$
$\theta \leftarrow \mathrm{PCA}(\mathbf{V}_c)$
$\boldsymbol{\gamma} \leftarrow binaryMask(\mathbf{V}_c)$
$e_c \leftarrow \|(\mathbf{c} - \mathbf{c}_d)\| / \|\mathbf{c}_d\|$
$e_\theta \leftarrow (\theta - \theta_d)/\pi$
$e_\gamma \leftarrow \Sigma(\boldsymbol{\gamma} \oplus \boldsymbol{\gamma}_d)/\Sigma\,\boldsymbol{\gamma}_d$
while $(e_c \geq e_c^{stop})$ & $(e_\theta \geq e_\theta^{stop})$ & $(e_\gamma \geq e_\gamma^{stop})$ **do**
  $\kappa_{min}, \kappa_{max}, d_{min}, d_{max} \leftarrow constAdaption(e_T)$
  $\mathbf{u} \leftarrow acquisition(\mu, \sigma)$
  $\mathbf{V}_c, \mathbf{X}, \mathbf{Y} \leftarrow applyPushingAction(\mathbf{u})$
  $f(\mathbf{x}) \leftarrow updateModel(\mathbf{X}, \mathbf{Y})$
  $\mathbf{c} \leftarrow 1/M\,\mathbf{V}_c\mathbf{1}_M$
  $\theta \leftarrow \mathrm{PCA}(\mathbf{V}_c)$
  $\boldsymbol{\gamma} \leftarrow binaryMask(\mathbf{V}_c)$
  $e_T \leftarrow computeTotalError(\mathbf{c}, \theta, \boldsymbol{\gamma})$
end while
return $e_T, \mathbf{X}, \mathbf{Y}, \mathbf{V}_c$

[6]. In the first test, we consider a $0.2 \times 0.1 \times 0.02$ [m] (length $\times$ width $\times$ depth) flat rectangular deformable object, modelled with the finite element method (FEM). The object is placed over a transparent table, and it is detected by a virtual camera beneath the table. We extract the contour of the object by means of the OpenCV library, by color thresholding in the HSV space. The manipulator model is a *Kuka iiwa* arm attached to the table at a certain distance from the object. The goal of the task is to push the object to a target position while preserving its initial shape and orientation. For obtaining the initial model, we apply the space filling technique LHS with $S_0 = 10$ initial samples. In addition, the learning process is accelerated by delimiting the solution search space according to the most effective observed pushing actions.

The first row of Figure 2 shows four snapshots of the first experiment, at the initial time and after the 10^{th}, 18^{th} and 53^{th} pushing actions. In turn, the third row depicts the evolution of the task errors (left), and the evolution of the pushing parameters and the cost function along the task (right). In the latter, the left axis corresponds to the values of the pushing parameters κ and d, while the right axis indicates the values of ξ . Note that after the first 10 pushes, an initial model is learnt, and the position error decreases fast. Then, the rotation error decreases and the position and shape error start to increase. However, once the model learns to push under the new state of the object, the errors start to decrease until they reach the stop values. With respect to the pushing actions, we can see that the system exploits those pushing actions that reduce e_T with respect to the previous instant until they stop being effective. Then, the system iterates to find the pushing action that is most likely going to reduce the error at the current state of the system.

In the second scenario, we place a non-regular shaped object of 0.13 [m] width, 0.2 [m] length and 0.01 [m] thickness over the table, with different deformation properties than the previous one. These properties are set in order to simulate the behaviour of a piece of meat. We control the elasticity by adjusting the Lamé parameters of the model, and the friction between the table and the object by increasing the mass of the model. The task consists in pushing the object to a target place, by maintaining a compact configuration and preserving the initial orientation. For maintaining a compact configuration, we define a rectangular goal shape with aspect ratio close to 1. Note that γ_d can be seen as a deformation constraint for the algorithm. We set $S_0 = 10$, and due to the fact that longer pushes are required to move this object, we increase d_{min} and d_{max} with respect to the previous case.

The second row of Figure 2 shows the bottom view of the system at four different pushing instants, and the fourth row depicts the evolution of the task errors, the pushing actions and the cost function. In this example, the error decreases fast at the beginning when the initial model is learnt. After that, the errors continue to decrease with some small oscillations, and finally they reach the stop values. As for the pushing parameters and the cost function values, the same behavior than in the first scenario is shown: the system exploits the actions that decrease ξ until they stop being effective, and then it explores to find new parameters.

7 Conclusion

Manipulating deformable objects is a challenging task for robotic systems due to the large variety of deformations that can appear in the object, which demand specific techniques which may be dependent on the material properties. Unlike rigid objects, any force applied to a deformable object cannot only translate/rotate it, but also alters its shape. Despite the challenges, it is a crucial skill required for robots to manipulate a wide variety of household and industrial objects.

In this work, an algorithm based on Bayesian optimization is proposed to achieve robotic pushing of thin fragile deformable objects. The proposed approach has multiple advantages. First, the temporal evolution of the object is considered and a policy is learnt to perform a series of pushes on the object to move it from one location to another on a plane. By doing so, the shape of the deformable object is also actively controlled in order to maintain it within an admissible range. Second, in comparison to modeling techniques such as finite element methods, where a precise knowledge of the material property is required, our approach relies on a probabilistic model that is computed in few trial iterations and updated over time to make better predictions. Similarly, in comparison to approaches based on deep learning, the proposed probabilistic model is generic and works well for unknown deformable objects without the need of large new training data.

The approach is validated in simulation with objects of different shapes and properties. As future work, the pushing action would be further extended with a grasping action, where a thin deformable object is pushed to the edge of the table and a grasp utilizing the edge as environmental constraint can be executed [25].

Acknowledgments. This work has received funding from the European Union's Horizon Europe research and innovation program under grant agreement No. 101070600, project SoftEnable. This work was also supported via projects PID 2021-124137OB-I00 and TED2021-130224B-I00 (funded by MCIN/AEI/10.13039 /501100011033), projects T45_23R and T73_23R by Gobierno de Aragón, ERDF A way of making Europe and the European Union NextGenerationEU/PRTR.). The first author was partially supported by the EU through the European Social Fund (ESF) "Construyendo Europa desde Aragón".

References

1. Agudo, A., Moreno-Noguer, F., Calvo, B., Montiel, J.M.M.: Sequential non-rigid structure from motion using physical priors. IEEE Trans. on Pattern Analysis and Machine Intelligence 38(5), 979–994 (2016)
2. Antonova, R., Yang, J., Sundaresan, P., Fox, D., Ramos, F., Bohg, J.: A bayesian treatment of real-to-sim for deformable object manipulation. IEEE Robotics and Automation Letters 7(3), 5819–5826 (2022)
3. Arriola-Rios, V.E., Wyatt, J.L.: A multimodal model of object deformation under robotic pushing. IEEE Trans. on Cognitive and Developmental Systems 9(2), 153–169 (2017)
4. Bauza, M., Rodriguez, A.: A probabilistic data-driven model for planar pushing. In: IEEE Int. Conf. on Robotics and Automation (ICRA). pp. 3008–3015 (2017)
5. Chai, C.Y., Peng, W.H., Tsao, S.L.: Object rearrangement through planar pushing: A theoretical analysis and validation. IEEE Trans. on Robotics 38(5), 2703–2719 (2022)
6. Coumans, E.: Pybullet real-time physics simulation. https://pybullet.org (2010)
7. Cox, D., John, S.: A statistical method for global optimization. In: IEEE Int. Conf. on Systems, Man, and Cybernetics. pp. 1241–1246 (1992)
8. Cuiral-Zueco, I., Karayiannidis, Y., Lopez-Nicolas, G.: Contour based object-compliant shape control. IEEE Robotics and Automation Letters 8(8), 5164–5171 (2023)
9. Frank, B., Stachniss, C., Abdo, N., Burgard, W.: Using gaussian process regression for efficient motion planning in environments with deformable objects. In: AAAIWS'11-09: Proceedings of the 9th AAAI Conference on Automated Action Planning for Autonomous Mobile Robots. pp. 2–7 (2011)
10. Gao, Z., Elibol, A., Chong, N.Y.: Zero moment two edge pushing of novel objects with center of mass estimation. IEEE Trans. on Automation Science and Engineering 20(3), 1487–1499 (2023)
11. Han, L., Zhang, Y., Wang, H.: Vision-based contact point selection for the fully non-fixed contact manipulation of deformable objects. IEEE Robotics and Automation Letters 7(2), 4368–4375 (2022)

12. Hu, Z., Sun, P., Pan, J.: Three-dimensional deformable object manipulation using fast online gaussian process regression. IEEE Robotics and Automation Letters 3(2), 979–986 (2018)
13. Huang, J., Cai, Y., Chu, X., Taylor, R.H., Au, K.W.S.: Non-fixed contact manipulation control framework for deformable objects with active contact adjustment. IEEE Robotics and Automation Letters 6(2), 2878–2885 (2021)
14. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv:1412.6980 (2014)
15. Langsfeld, J.D., Kabir, A.M., Kaipa, K.N., Gupta, S.K.: Robotic bimanual cleaning of deformable objects with online learning of part and tool models. In: IEEE Int. Conf. on Automation Science and Engineering (CASE). pp. 626–632 (2016)
16. Lopez-Nicolas, G., Ozgur, E., Mezouar, Y.: Parking objects by pushing using uncalibrated visual servoing. Autonomous Robots 43, 1063–1078 (2019)
17. Moya-Lasheras, E., Sagues, C.: Run-to-run control with bayesian optimization for soft landing of short-stroke reluctance actuators. IEEE/ASME Trans. on Mechatronics 25(6), 2645–2656 (2020)
18. Moza, S.: sahilm89/lhsmdu: Latin hypercube sampling with multi-dimensional uniformity (lhsmdu): Speed boost minor compatibility fixes (2020)
19. Nogueira, J., Martinez-Cantin, R., Bernardino, A., Jamone, L.: Unscented bayesian optimization for safe robot grasping. In: IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS). pp. 1967–1972 (2016)
20. Nyikosa, F.M., Osborne, M.A., Roberts, S.J.: Bayesian optimization for dynamic problems. arXiv:1803.03432 (2018)
21. Rasmussen, C.E., Williams, C.K.I.: Gaussian processes for machine learning. Cambridge, MA, USA: MIT Press (2006)
22. Ruggiero, F., Petit, A., Serra, D., Satici, A.C., Cacace, J., Donaire, A., Ficuciello, F., Buonocore, L.R., Fontanelli, G.A., Lippiello, V., Villani, L., Siciliano, B.: Nonprehensile manipulation of deformable objects: Achievements and perspectives from the robotic dynamic manipulation project. IEEE Robotics & Automation Magazine 25(3), 83–92 (2018)
23. Sanchez, J., Corrales, J.A., Bouzgarrou, B.C., Mezouar, Y.: Robotic manipulation and sensing of deformable objects in domestic and industrial applications: a survey. Int. J. Robotics Research 37(7), 688–716 (2018)
24. Stüber, J., Zito, C., Stolkin, R.: Let's push things forward: A survey on robot pushing. Frontiers in Robotics and AI 7 (2020)
25. Sundaram, A.M., Friedl, W., Roa, M.A.: Environment-aware grasp strategy planning in clutter for a variable stiffness hand. In: IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS). pp. 9377–9384 (2020)
26. Yin, H., Varava, A., Kragic, D.: Modeling, learning, perception, and control methods for deformable object manipulation. Science Robotics 6(54), eabd8803 (2021)
27. Zhu, J., Cherubini, A., Dune, C., Navarro-Alarcon, D., Alambeigi, F., Berenson, D., Ficuciello, F., Harada, K., Kober, J., Li, X., Pan, J., Yuan, W., Gienger, M.: Challenges and outlook in robotic manipulation of deformable objects. IEEE Robotics & Automation Magazine 29(3), 67–77 (2022)