Deformation-aware Contact-rich Manipulation Skills Learning and Compliant Control

Weiyong Si[1], Cheng Guo[2], Jiale Dong[3], Zhenyu Lu[1], and Chenguang Yang*[1]

[1] University of the West of England, and Bristol Robotics Laboratory, Bristol, BS16 1QY, UK,
[2] University of Bristol, and Bristol Robotics Laboratory, Bristol, Bristol BS8 1TH, UK,
[3] South China University of Technology, Guangzhou, 510640, China

Abstract. In this paper, we study a vision-based reactive adaptation method for contact-rich manipulation tasks, based on the compliant control and learning from demonstration. For contact-rich tasks, the compliant control method is essential, especially when interacting with a deformable object with unknown properties, such as pizza dough. Learning from demonstration (LfD) provides a solution for this challenging task. However, the generalisation capabilities of LfD for deformable object manipulation tasks are still a challenging and opening issue, especially for unknown and dynamic tasks. Therefore, in this work, we investigate the vision and force-based perception feedback to enhance the generalisation of the LfD outcomes. The computer vision algorithm was developed to recognise the shape of the object and calculate the deviation between the desired shape and the current shape. The deviation of shape adjusts the parameters of learned primitive skills encoded by Dynamic Movement Primitives (DMPs). We adopt the pizza dough rolling task as the typical case to evaluate the performance of the proposed method. The shape and thickness of the dough can be made to the desired shape and thickness.

Keywords: Deformation-aware Pizza Dough Rolling (DaPADR), Learning from demonstration, Compliant control.

1 Introduction

Collaborative robots (cobots) have gained much attention in both academic and industrial communities. In the past, collaborative robots, such as Baxter robot and Franka Emika Panda etc., have been employed in a number of fields, such as manufacturing, medical examinations, and home service [1]. Although the cobots have been widely used in industry, it is still hard to interact with deformable objects, such as textile cloth, soft and deformable food in the kitchen and human tissue [2–4]. Manipulating soft and deformable objects is challenging due to the lack of an accurate model of the object and the unknown properties of the deformable objects. However, deformable objects, such as cloth, fruits and even our bodies, are very common in daily life, and interaction with deformable and soft objects is ubiquitous. Therefore, manipulating deformable objects by robots has gained much attention in both academic and industrial communities.

There is some literature on deformable object modelling and control, such as pizza dough manipulation [5–8]. Siciliano et al. has conducted comprehensive study on the pizza dough manipuation, including the perception, trajectory planning and controller design for pizza dough. For example, the modelling of the deformable object was studied, and optimization methods were investigated to generate trajectory for the robots in realtime [5]. The pizza dough manipulation by robots has attracted much attention. Stretching a dough with a rolling pin is a typical nonprehensile manipulation [4], which has been studied in [5].

Recently, as machine learning methods have been adopted in the robotic field, various robot learning methods have been proposed to tackle the autonomous execution of robots. A number of literature on robot skill learning has been published [1]. Learning from demonstration (LfD) is one effective robot skill learning technology lying on the human teachers' demonstrations [9]. Iterative learning and identification were investigated for the state monitoring [10]. In our previous work, LfD has been investigated for various manipulation tasks, such as writing characters, cutting vegetables/ fruit, assembly, and robot-assisted surgery [1]. In addition, the LfD based hierarchical framework was explored to learn the task of rolling pizza dough [11].

Vision-based trajectory planning and control methods, such as visual servoing, have been well studied in the industry [12]. However, pure vision-based planning and control method is hard to manipulate deformable objects autonomously. Due to the complex property of the deformable objects, it is impossible to plan the trajectories of robots in complex tasks manually in advance. Visual and haptic information provides real-time feedback for online planning and decision making of robots working on complex tasks under unknown conditions [13]. On the other hand, with developments in the field of machine learning, deep convolutional networks are outperforming traditional computer vision methods in many visual recognition tasks. Some new neural network architectures such as U-net [14, 15], are capable of accurate segmentation of complex medical image including cardiac MR segmentation [14] and lung cancer nodule detection [15], that can reach near radiologist level performance. Thus the architecture of deep convolutional networks can also be used to predict the boundaries of flexible matter and help robots determine their workspace. These applications are widely seen to replace traditional computer vision which has been used to obtain more accurate, noise-free data.

To tackle with the challenging issue of autonomous manipulation of deformable objects, we proposed a learning and adaptive framework for compliant manipulation of deformable objects, as shown in Fig. 1. The contributions of this work can be summarized as follows: 1) Vision-based deformation-aware method was investigated for deformable object manipulation to achieve the target shape. Comparing our previous work [4], we investigated subgoal decision strategy based

[4] Note nonprehensile manipulation means manipulation without grasping, such as pushing, flipping, throwing, and squeezing.

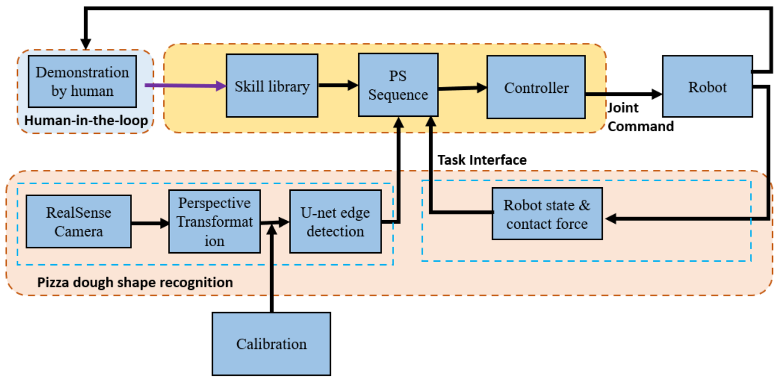

Fig. 1. The overall structure of the proposed framework.

on shape recognition and deviation from the target shape. 2) In addition, to manipulate the deformable dough, a compliant controller was designed to execute the trajectories generated from LfD. We integrated the learning and compliant control for deformable objects in an unified framework. 3) We adopted pizza dough stretching tasks as evaluation case on a real robot; and the pizza dough's shape and thickness are the evaluation criteria.

The rest of this paper is organised as follows: Section II presents the preliminary knowledge of robot manipulator dynamics and compliant control. The proposed deformation-aware manipulation skill learning and complaint control method is also presented in Section II. The experiment and results and performance analysis are presented in Section III. Finally, we conclude the paper with a discussion about future work in Section IV.

2 Problem formulation and methodology

2.1 Vision-based deformation recognition

The vision recognition is to detect the outline of the dough on the table and to attain the position of the dough edge in the robot's base coordinate system. The camera calibration is required to record the robot's working space before the experiment. As shown in Fig.3., four ArUco markers are placed on the table with a certian distance from each other, and the object needs to be placed within this area and the dough also needs to stay in this workspace. With these markers, the camera's view plane (the camera's frame) can be mapped onto the table using a perspective transformation method. Also, the camera frame can be used to obtain an image of the dough within the working area. Once the image is obtained, the dough can be distinguished from the background using U-net and the final positions of the dough edges can be obtained using canny edge detection.

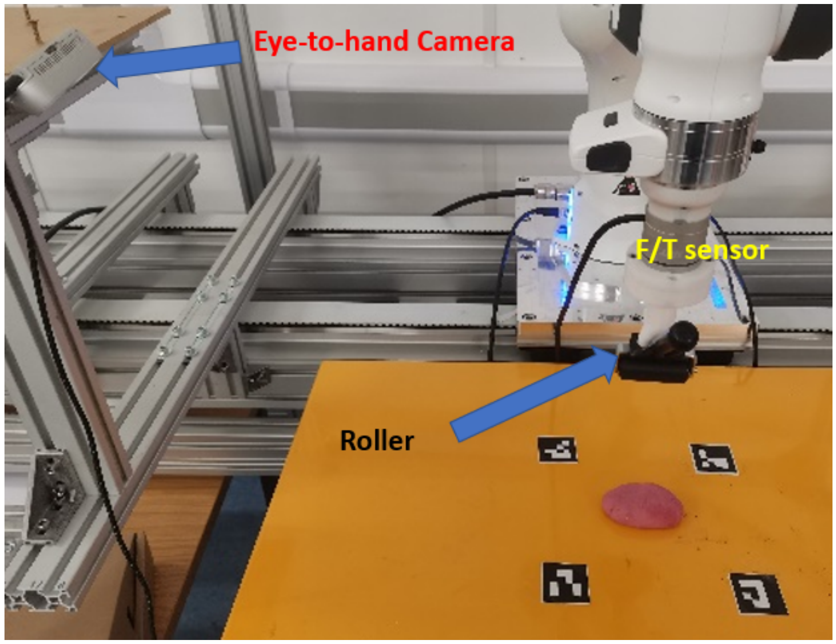

Fig. 2. The setup of deformation-aware stretching a pizza dough with a rolling pin. RealSense camera and F/T sensor were used to capture the deformation of shape and the contact force. A customised roller was used to stretch the pizza dough.

Perspective transformation Perspective transformation is the projection of an image onto a new viewing plane, also known as projective mapping [16]. The transformation matrix required in the affine transformation of an image is a 2×3 two-dimensional plane transformation matrix, whereas the perspective transformation is essentially a three-dimensional transformation of space, where the three dimensional coordinates are projected onto a different viewing plane according to their sub-coordinate variance. In this experiment we need the diagram of the view plane where the table is located on the image.

For a perspective transformation, it can be thought of a projection from a point on a quadrilateral onto the target quadrilateral after a linear transformation, where the shape of the target quadrilateral is a known condition. A general formula for a perspective transformation is as follows [17],

$$\begin{bmatrix} X' \\ Y' \\ \omega' \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \tag{1}$$

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \tag{2}$$

where u and v are the pixels of the original image. New coordinates x, y after transformation by perspective is $x = \frac{X'}{\omega'}$ and $y = \frac{Y'}{\omega'}$.

The above equation shows that the coordinates under the new view plane can be expressed as [17]:

$$x = \frac{X'}{\omega'} = \frac{a_{11}u + a_{12}v + a_{13}}{a_{31}u + a_{32}v + a_{33}} = \frac{k_1 u + k_2 v + k_3}{k_7 u + k_8 v + 1} \tag{3}$$

$$y = \frac{Y'}{\omega'} = \frac{a_{21}u + a_{22}v + a_{23}}{a_{31}u + a_{32}v + a_{33}} = \frac{k_4 u + k_5 v + k_6}{k_7 u + k_8 v + 1} \tag{4}$$

To solve for the eight unknowns factors in this equation, it requires eight sets of equations. Four known coordinate points are the pixel positions read from ArUco markers in the original image and four coordinate points of the target image are set to $(0,0), (0,500), (500,500), (500,0)$. This is due to the fact that the corresponding work area captured by camera is a $12.5 * 12.5cm$ square on the table plane, which corresponds to $1/500 * 12.5cm$ per pixel on the image. The transformation matrix A is calculated for each pixel point in the original image, the new image is projected to give a top view of the dough on the table.

Image segmentation Once the perspective-transformed image is obtained, the exact position of the dough needs to be obtained from this image. A deep learning approach with higher accuracy was chosen over traditional vision algorithms. The architecture of U-net was shown in Fig.3, which was used to segment the background from the dough in this work. It consists of a systolic path (left side) and an extended path (right side). The systolic path follows the typical structure of a convolutional network like VGG19. It consists of two 3×3 convolution (padded convolutions) layers, and each layer followed by a rectified linear unit (ReLU) and a 3x3 convolution layer with stride 2 for downsampling. At each downsampling step, we keep the number of feature channels. Each step in the expansion path includes an upsampling of the feature map, followed by a 2×2 convolution ("up-convolution") that halves the number of feature channels, concatenates with the correspondingly cropped feature map in the contraction path, and performs two 3×3 convolutions, each followed by a ReLU. In the final layer, a 1×1 convolution is used to map each 64-component feature vector to the desired number of categories. 1×1 convolution classifies each pixel in the image thereby distinguishing the dough from the background. The output of the network is shown in Fig.6.(c) and (f).

The network is trained using a stochastic gradient descent using the input photos and their accompanying segmentation maps. Before being fed into the network, the input pictures and segmentation maps are downsized to 256×256 to match the network's input. Unlike the network mentioned in the U-net paper, the output of this network has exactly the same size as the input image, which ensures that every pixel can be classified. The advantage is acquiring the image with a segmentation of the original input image, which better improves the accuracy of dough recognition.

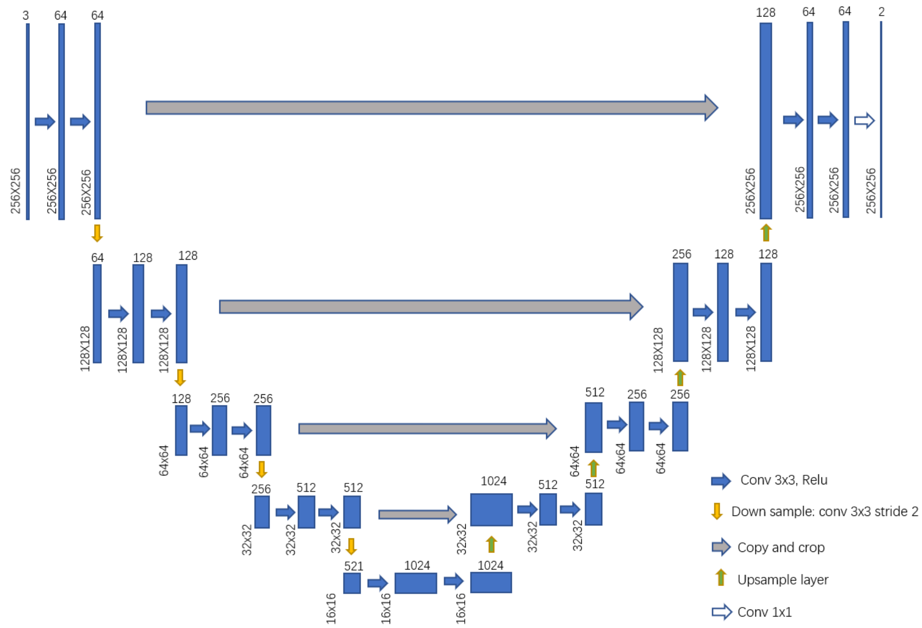

Fig. 3. U-net architecture. Each blue box corresponds to a multi-channel feature map. The number of channels is indicated at the top of the box. The dimensions of the feature map are in the lower left corner of the blue box. The arrows indicate the different operations.

Finally, by performing canny edge detection on the generated dough mask image, the positions corresponding to edge pixel are transformed into positions in the robot coordinate system, thus completing the recognition of dough contours.

2.2 Primitive skills modelling

DMPs was proposed to model the primitive behavior by roboticist [18]. In our previous work, we have proposed various unified and compact representation of the primitive motion skills by dynamic systems. For example, we studied the unified representation of position and orientation of primitive skills [19]. Dynamic system (DS)-based primitive skill modelling have been studied recently, as the convergence and stability can be guaranteed. Due to the reactive and stable characteristics of the dynamic system-based method, the DS-based method could be used to deal with the safety-critical scenarios [20].

Dynamic movement primitives (s) We model the primitive skills by position and orientation in a unified formulation in Cartesian space,,

$$\tau \dot{z} = \alpha_z(\beta_z(g^p - p) - z) + f_p(x) \tag{5}$$

$$\tau \dot{p} = z \tag{6}$$

$$\tau \dot{x} = -\alpha x \tag{7}$$

where g^p and p are target position and current position respectively, z is the velocity of dynamic system, β_z and α_z are designed parameters. $f_p(x)$ is the nonlinear term, which is used to modify the behaviour of the second-order system. The defination of $f_p(x)$ will be given later and the weight of this term can be learned through human demonstration data. The evolution of the dynamic system is determined by the canonical system, Eq.7, where the τ can tune the duration of the dynamic system, and the α is a positive parameter. The canonical system is shared by the position and orientation dynamic system to coordinate the translation and rotation simutaneously.

The orientation skill can also be encoded by the DMPs, however, the skill model is adopted by the quaternion-based form [21]. In our previous work, we encoded the translation and orientation simultaneously by DMPs [19]. The details of the model can refer [19], we define the model here to facilitate understanding,

$$\tau \dot{\eta} = \alpha_z \left(\beta_z 2 \log(g^o * \bar{q}) - \eta \right) + f_o(x) \tag{8}$$

$$\tau \dot{q} = \frac{1}{2} \eta * q \tag{9}$$

where η is the angular velocity, g^o and q are the target angle and current angle encoded by quaternion respectively. $\bar{q}$ represents the quaternion conjugation, and $*$ denotes the quaternino product, details can refer to [21]. β_z and α_z are designed parameters, which can be consistent with the position DMPs, $f_o(x)$ is the nonlinear term, which is used to modify the behaviour of the second-order system.

The nonlinear force term is used to define the transient behavior of the dynamic system. This term usually consists of a set of nonlinear basis function (RBFs), which can be given by,

$$f_p = \frac{\sum_{k=1}^{N} w_k^p \Phi_k(x)}{\sum_{k=1}^{N} \Phi_k(x)} x \tag{10}$$

$$f_o = \frac{\sum_{k=1}^{N} w_k^o \Phi_k(x)}{\sum_{k=1}^{N} \Phi_k(x)} x \tag{11}$$

$$\Phi_k(x) = \exp(-h_k (x - c_k)^2) \tag{12}$$

where N is the number of the RBFs, and the x is the phase variable determined by the canonical system. w_k^p and w_k^o are the weights of the i^{th} RBFs, $i \in [1, N]$ and h_k and c_k are the centres and widths of the RBFs, respectively.

We generate the complex trajectory by merging the individual primitive skill (PS), encoded by DMPs for the complex and contact-rich task with multiple-step, as shown in Algorithm 1. The multiple primitive skill are encoded by behavior tree [22], the DMP-based primitive skill was modelled as execution nodes. The target position and orientation of each primitive skills are calculated by shape recognition and desired shape and thickness. In this work, we only consider circular desired shape, four position and four orientation skills are needed. The four primitive skills are approaching, rolling forward, rolling back and back origin. For the circular shape, the primitive skill library with four primitive skills can generate trajectory for the roller. However, the vision-based trajectory generation algorithm can also be employed to different shapes. Different shape only needs to modify the control flow nodes in behavior tree.

Algorithm 1 Vision-based Trajectory Generation

while (!achieve desired shape and thickness) **do**
  starting point P_{st}, direction dir = vision feedback()
  while (current position != end point) **do**
    PS1.run(current position, P_{st}), PS2.run(current direction, dir)
  end while
  while (current position != end point) **do**
    PS3.run(current position, P_{st}), PS4.run(current direction, dir)
  end while
  while (current position != end point) **do**
    PS5.run(current position, P_{st}), PS6.run(current direction, dir)
  end while
  while (current position != end point) **do**
    PS7.run(current position, P_{st}), PS8.run(current direction, dir)
  end while
end while

2.3 Compliant control

The dynamic of the robot manipulator needs to be modelled for the compliant controller. For a serial robot manipulator, the dynamic equation in joint space can be given,

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + \tau_f(\dot{q}) + d_u = \tau_c + J^T(q)F_e \tag{13}$$

where $M(q)$ is the positive-definite and symmetric matrix; $C(q,\dot{q})$ represents the Coriolis and centrifugal term; $G(q)$ is the gravity torque; $\tau_f(\dot{q})$ is the friction torque and d_u represents the unmodelled terms. In this work, we ignored these terms. q is the joint angle, $\dot{q}$ and $\ddot{q}$ are the velocity and acceleration, respectively. F_e is the interaction force between end-effector and environment. τ_c is the control input, and $J^T(q)$ is the Jacobian transpose.

We model the manipulation skill in the task space to improve the generalization capability of the learned skills for different robots. In addition, encoding the manipulation skills in the task space is straightforward to learn versatile manipulation, including translation and orientation skills etc. Therefore we derive the controller in the task space to achieve the learned skills. The relationships between position, velocity and acceleration of the end-effector and joint variables are described as,

$$x(t) = f(q) \tag{14}$$

$$\dot{x}(t) = J(q)\dot{q} \tag{15}$$

$$\ddot{x} = \dot{J}(q) + J(q)\ddot{q} \tag{16}$$

where $x(t)$ is the position in the task space, $\dot{x}(t)$ and $\ddot{x}(t)$ represent the velocity and acceleration of the end-effector in task space. $f(q)$ is the feedforward kinematic relationship.

Considering the above equations and (1), we can derivate the dynamics of the robot manipulator in task space,

$$\Lambda_p(q)\ddot{x} + B(q,\dot{q})\dot{x} + G_p(q) + T_f(\dot{q}) + D = T_c + F_e \tag{17}$$

where $\Lambda_p(q)$ is the inertial matrix in task space, $B(q,\dot{q})$ account for Coriolis and centrifugal term, $G_p(q)$ is the gravity force, $T_f(\dot{q})$ is the friction force [5], D is the unmodelled force. T_c is the control input, and F_e is the interaction force, so we can get T_c,

$$T_c = A_d\ddot{x} + D_d(\dot{x} - \dot{x}_d) + K_d(x - x_d) \tag{18}$$

where A_d , D_d and K_d are the desired mass, damping and stiffness. The x_d and $\dot{x}_d$ are the desired position and velocity, respectively. The impedance is determined based on the desired interaction behaviour, and the properties of the environment. The stability of impedance controller can refer to [23].

3 Experiment

We carry out pizza dough stretching with a rolling pin to evaluate the performance of the proposed framework on a 7 DoFs cobot. A RGBD (RealSense) camera was used to capture the deformation of the object and a computer vision algorithm was used to calculate the difference between the real shape and the desired shape. A customised roller was used to stretch the pizza dough [6]. The setup of the experiment can be seen in Fig. 2. We defined evaluation metrics for dough rolling, as shown in Fig. 4. The red outline is the desired shape, and we defined two axes (the yellow lines) to judge whether the desired shape is achieved. During the dough rolling, we adjust the orientation of the roller based on the error along these two axes.

[5] Note we ignore the friction force in practice.

[6] Note the pizza dough is not as large as the real pizza dough, because the roller is not large enough. The proposed method can be employed for big dough if we adopt a large roller.

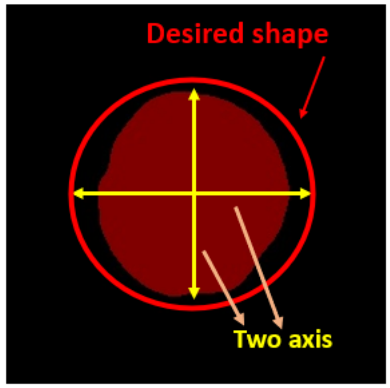

Fig. 4. The evaluated metric of dough rolling. The red outline is the desired shape, and we defined two axes (the yellow lines) to judge whether the desired shape is achieved. During the dough rolling, we adjust the orientation of the roller based on the error along these two axes.

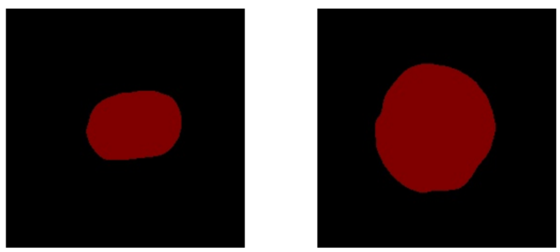

Fig. 5. The result of deformation recognisation and segmentation. The left one is the original shape of dough, and the right one is the final shape of dough.

3.1 The result of deformation segmentation

As shown in Figs. 5 and 6, the changing of the shape of the dough is shown during the rolling process. We test the image recognition and segmentation accuracy by U-net. We collected 20 real images during the dough rolling, and humans labelled the real outline of the dough. The segmentation error is less than 1% pixel, meaning that the dough is accurately segmented from the image by the model.

3.2 The position, orientation and contact force

Fig. 7 is the trajectories in 3D during the rolling. The robot approaches the dough and then rolls the dough based on the visual feedback to adjust the start point and rolling orientation. The rolling process is iteractively changing until the desired shape is achieved.

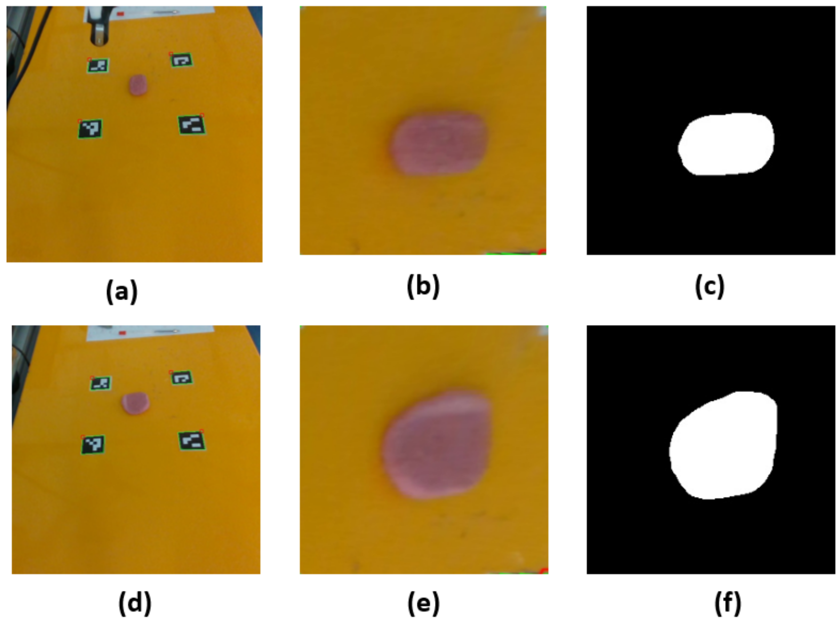

Fig. 6. Snapshot of dough stretching. (a)-(c) are the initial shape of the dough. (d)-(f) are the shape after stretching. (a) is the original image from camera. (b) is the middle image after neural networks processing. (c) is the segmentation of the dough. The machine vision algorithm can detect and calculate the deformation of dough in real time.

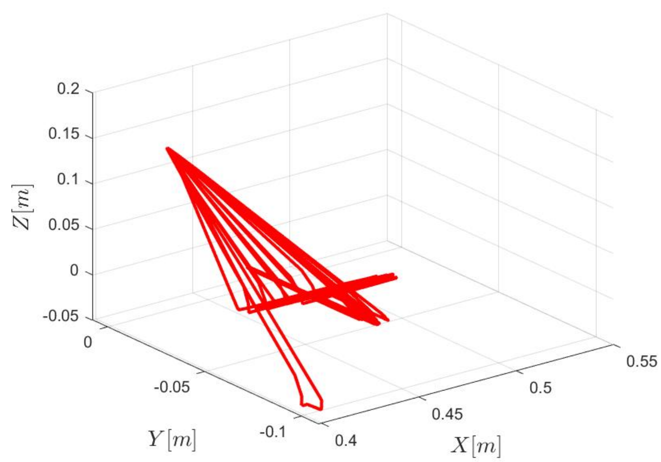

Fig. 7. The trajectory in 3D dimention.

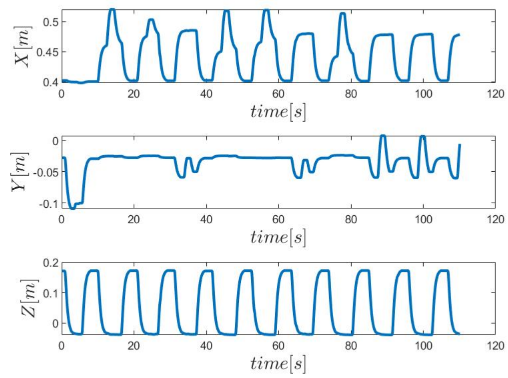

Fig. 8. The position trajectory along X,Y and Z.

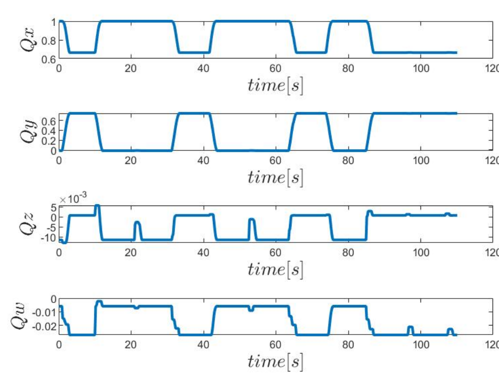

Fig. 9. The orientation trajectory represented by quaternion.

In Fig. 8, position trajectories along X,Y, and Z during the rolling is presented. The robot's range of motion in the X-Y direction changes as the dough is deforming and stretching.

Fig. 9 is the orientation trajectories along X,Y, and Z during the rolling. In this experiment, we can see the robot keep changing between two directions, which are based on the direction of movement of the dough stretched in different axes. Fig. 10 shows the contact force along X,Y, and Z during the rolling. Due to the complex and unknown properties of the dough, the change of contact force along X-Y-Z is complex in a period. Therefore, it is hard to control the contact force accurately. LfD through bilateral teleoperation provides a promising approach to transferring the contact-rich skill from humans to robots.

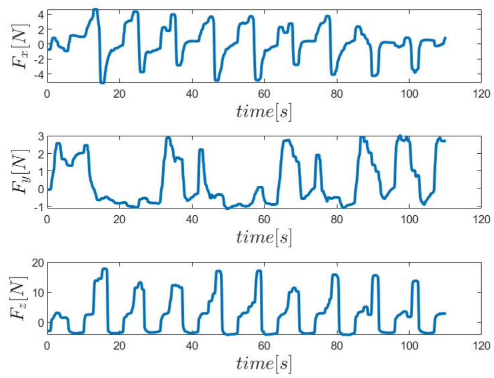

Fig. 10. The contact force along X,Y and Z..

3.3 The result of thickness

The thickness of the final dough is one metric for dough rolling by robots. We adopt the hybrid force/ position controller to detect the height of the final dough. The F_z along the Z-axis is controlled to 0, and the X-Y motion trajectory is controlled by teleoperation. The height along the Z-axis was recorded, and the variance of thickness can be measured.

We compare the performance of human rolling by teleoperation and autonomous rolling by the robot, and the results are shown in Table 1.

Table 1. Flatness quality performance.

Category	Variance of height
Human teleoperation	$3mm \pm 0.3mm$
Robot autonomous	$3.3mm \pm 0.2mm$

4 Discussion and conclusion

In this paper, we proposed a deformation-aware method for the manipulation of deformable objects. The perspective transformation and U-net method were used to recognise the object and segment the object from the background. Based on the target shape and size of the object, we calculate the desired task variables (desired position and orientation). The realtime trajectory was generated from the learned skill based on human demonstration. And compliant controller was designed to track the desired trajectory. Finally, we evaluate the shape of the final dough by calculating the error along the two axes of the dough. And the variance of the dough in the Z axis reflects the thickness quality. The result shows that the shape error is bellow 5% and the variance of the thickness is 0.3mm. However, the softness of the dough varies a lot, and we will investigate the variable impedance control and iterative learning method to overcome the challenge in the future.

5 Acknowledgements

This work was supported in part by the H2020 Marie Skłodowska-Curie Actions Individual Fellowship under Grant 101030691.

References

1. W. Si, N. Wang, and C. Yang, "A review on manipulation skill acquisition through teleoperation-based learning from demonstration," *Cognitive Computation and Systems*, vol. 3, no. 1, pp. 1–16, 2021.
2. J. Matas, S. James, and A. J. Davison, "Sim-to-real reinforcement learning for deformable object manipulation," in *Conference on Robot Learning.* PMLR, 2018, pp. 734–743.
3. 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.
4. W. Si, Y. Guan, and N. Wang, "Adaptive compliant skill learning for contact-rich manipulation with human in the loop," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 5834–5841, 2022.
5. J.-T. Kim, F. Ruggiero, V. Lippiello, and B. Siciliano, "Planning framework for robotic pizza dough stretching with a rolling pin," in *Robot Dynamic Manipulation.* Springer, 2022, pp. 229–253.
6. A. Gutiérrez-Giles, F. Ruggiero, V. Lippiello, and B. Siciliano, "Closed-loop control of a nonprehensile manipulation system inspired by the pizza-peel mechanism," in *2019 18th European Control Conference (ECC).* IEEE, 2019, pp. 1580–1585.
7. 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.
8. A. C. Satici, F. Ruggiero, V. Lippiello, and B. Siciliano, "A coordinate-free framework for robotic pizza tossing and catching," in *2016 IEEE International Conference on Robotics and Automation (ICRA).* IEEE, 2016, pp. 3932–3939.
9. W. Si, N. Wang, Q. Li, and C. Yang, "A framework for composite layup skill learning and generalizing through teleoperation," *Frontiers in Neurorobotics*, vol. 16, 2022.
10. Z. Lu, N. Wang, and C. Yang, "A novel iterative identification based on the optimised topology for common state monitoring in wireless sensor networks," *International Journal of Systems Science*, vol. 53, no. 1, pp. 25–39, 2022.
11. N. Figueroa, A. L. P. Ureche, and A. Billard, "Learning complex sequential tasks from demonstration: A pizza dough rolling case study," in *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI).* Ieee, 2016, pp. 611–612.

12. B. Espiau, F. Chaumette, and P. Rives, "A new approach to visual servoing in robotics," *ieee Transactions on Robotics and Automation*, vol. 8, no. 3, pp. 313–326, 1992.
13. È. Pairet, P. Ardón, M. Mistry, and Y. Petillot, "Learning generalizable coupling terms for obstacle avoidance via low-dimensional geometric descriptors," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3979–3986, 2019.
14. O. Oktay, J. Schlemper, L. L. Folgoc, M. Lee, M. Heinrich, K. Misawa, K. Mori, S. McDonagh, N. Y. Hammerla, B. Kainz *et al.*, "Attention u-net: Learning where to look for the pancreas," *arXiv preprint arXiv:1804.03999*, 2018.
15. O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention.* Springer, 2015, pp. 234–241.
16. J. Mezirow, "Perspective transformation," *Adult education*, vol. 28, no. 2, pp. 100–110, 1978.
17. R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. New York, NY, USA: Cambridge University Press, 2003.
18. A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal, "Dynamical movement primitives: learning attractor models for motor behaviors," *Neural computation*, vol. 25, no. 2, pp. 328–373, 2013.
19. W. Si, N. Wang, and C. Yang, "Composite dynamic movement primitives based on neural networks for human–robot skill transfer," *Neural Computing and Applications*, pp. 1–11, 2021.
20. A. Billard, S. Mirrazavi, and N. Figueroa, *Learning for Adaptive and Reactive Robot Control: A Dynamical Systems Approach.* MIT Press, 2022.
21. A. Ude, B. Nemec, T. Petrić, and J. Morimoto, "Orientation in cartesian space dynamic movement primitives," in *2014 IEEE International Conference on Robotics and Automation (ICRA).* IEEE, 2014, pp. 2997–3004.
22. M. Saveriano and J. Piater, "Combining decision making and dynamical systems for monitoring and executing manipulation tasks," *Elektrotech. Inftech.*, vol. 137, no. 6, pp. 309–315, 2020.
23. N. Hogan, "Impedance control: An approach to manipulation: Part i—theory," 1985.