Nonprehensile Manipulation Control and Task Planning for Deformable Object Manipulation: Results from the RoDyMan Project

Fabio Ruggiero[1], Jung-Tae Kim[1], Alejandro Gutierrez-Giles[2], Aykut C. Satici[3], Alejandro Donaire[4], Jonathan Cacace[1], Luca Rosario Buonocore[5], Giuseppe Andrea Fontanelli[1], Vincenzo Lippiello[1], Bruno Siciliano[1]

[1] CREATE Consortium and Department of Electrical Engineering and Information Technology, University of Naples Federico II, Via Claudio 21, 80125, Naples, Italy
[2] Center for Research and Advanced Studies of the National Polytechnic Institute, Av. IPN 2508, San Pedro Zacatenco, 07300, Mexico City, Mexico
[3] Mechanical and Biomedical Engineering, Boise State University, 1910 University Dr., 83725, Boise, ID, USA
[4] School of Engineering, The University of Newcastle, University Drive, 2308, Callaghan, NSW, Australia.
[5] CERN, Router de Meyrin 385, Geneve 23, Switzerland.

Abstract. This chapter aims the broadcasting of the results achieved by the *RoDyMan* project about the task planning manipulation of deformable objects, and the nonprehensile manipulation control. The final demonstrator of the project is a pizza-making process. After an introduction to the general topic of nonprehensile manipulation, the mechatronic design and the high-level software architecture are described. Then, the smoothed particle hydrodynamic formulation is briefly introduced, along with the description of a detection method for a deformable object. The task planning for stretching a modelling clay, emulating the pizza dough, is sketched. After, the problematic control objective is split into several nonprehensile motion primitives: holonomic and nonholonomic rolling, friction-induced manipulation, and tossing are the described primitives. This chapter highlights the achievements reached so far by the project, and pave the way towards future research directions.

Keywords: nonprehensile dynamic manipulation, deformable object, perception, robot planning, robot control

1 Introduction

Last decades saw a dramatic improvement in powerful technology for sensing and actuation, leading to a significant advancement in robotic manipulation capabilities. However, the human skills in manipulating a massive range of objects in a dexterous and fast way are still far off from being replaced by robots.

Roughly speaking, manipulating an object means to change its configuration regarding position and orientation. Such a change can be performed in many ways on the basis of the performed grasp. In case the object is entirely caged between the fingertips or the

palm, supposing infinite power for the muscles, the hand can resist any external disturbance applied to the object. If even infinitesimal motions of the object are limited by the hand, the grasp is said to be in a *form closure* configuration, otherwise it is referred to as a *force closure* configuration [1]. Bilateral constraints appear in both closures. On the other hand, if only unilateral constraints are involved in the grasp, the hand can resist only external disturbances applied in specific directions. A simple example is given by an object pushed on a plane: the fingertip can resist only external disturbances applied on the object and counteracting the pushing direction; it is instead not able to resist to an external force lifting the object. In this case, *nonprehensile manipulation* occurs. Nonprehensile manipulation can be endorsed as *dynamic* when the dynamics of both the object and the manipulating hand are crucial to accomplishing the task. Some manipulation actions must be inevitably performed in a prehensile way, like screwing or unscrewing a bottle cap, while other manipulation actions must be performed in a nonprehensile way, like carrying a glass of liquid on a tray. Some other manipulation tasks can be instead carried out either in a prehensile or in a nonprehensile way, while other more complex tasks may need both ways to be finalised. An example may be given by a juggler that has to throw a ball with one hand (nonprehensile) while the other hand must grab another ball (nonprehensile for a dynamic catch, prehensile when the ball is firmly grasped between the fingertips and the palm).

Nonprehensile manipulation benefits of several advantages as highlighted in [2]. The workspace of the robot can be increased since unilateral constraints allow breaking the contact with the manipulated part (e.g., the robot can throw and dynamically catch the object). Since it is not anymore required having a standard gripper, the structure of the robot can be simplified because any available surface can be employed to manipulate the object through suitable contact forces. Finally, since form and force closure grasps are not involved, the dexterity of the manipulation task is augmented because it is possible to control more degrees of freedom of the object than the ones of the manipulator itself. Nevertheless, these advantages go to the detriment of an increase of the complexity of the planning and the control design.

1.1 The RoDyMan project

This chapter illustrates the results achieved by the *RoDyMan* project[1] in the development of a service robot that can manipulate rigid, elastic, and soft objects also in a nonprehensile way. The sought goals are split into three main research challenges, which outline the chapter too.

- *Robot design*. A mobile robotic platform is built from scratch to test the nonprehensile dynamic manipulation tasks addressed by the project.
- *Modelling, perception, and task planning for deformable objects*. Accurate modeling of deformable objects is essential to plan and simulate their motion and design the related planning and control tasks properly. Moreover, the real-time requirements posed by robot interaction need an accurate tracking of these objects.
- *Nonprehensile manipulation planning and control*. The project wants to advance the state of the art in controlling rigid objects in a nonprehensile way and starting

[1] http://www.rodyman.eu

the investigation of problems related to both prehensile and nonprehensile manipulation of deformable objects.

The *RoDyMan* final demonstrator is the realisation of an autonomous pizza maker. Preparing a pizza involves an exceptional level of manual dexterity. Besides, it is indeed a media attractor and a tribute to the hometown of the project. Videos of the related experiments can be found in the laboratory YouTube channel[2].

1.2 Literature review

Different projects tried to address nonprehensile manipulation using different approaches. The RIBA robot[3] can lift and set down patients from/to their beds and/or wheelchairs. The addressed transporting task is indeed nonprehensile, but it is not dynamic since the patient is often considered as a rigid object. Motion planning strategies could be enough to achieve the task. The robot is covered by a soft material to enhance safety during the human-robot interaction. The ERC SHRINE project[4] wanted instead to advance the robot manipulation skills to overcome the limitations preventing robots from safe and smooth operations in anthropic environments. Some manipulation actions were executed in a nonprehensile fashion.

Beyond the mentioned projects, advancements in the domain of robotic nonprehensile manipulation are very slow. Mainly because such a domain lacks a solid theoretical background, and this may be caused by the absence of a shared device where researchers can compare their works (i.e., multi-fingered hands for the robotic grasping domain; humanoids for the researchers studying walking; unmanned aerial vehicles for the aerial robotic community). Nonprehensile manipulation has instead very different tasks and situations that can be addressed with different devices. Therefore, applications are commonly limited to ad-hoc solutions to solve distinct problems.

As previously outlined, the benefits of nonprehensile manipulation are compensated by some drawbacks related to the difficulty in finding the suitable control design. For instance, the possible change of the contact status during a nonprehensile manipulation task leads to non-smooth dynamics, complicating the controller design. Further control problems arising in nonprehensile manipulation tasks are highlighted in [3], while a recent survey regarding nonprehensile manipulation is available in [4]. Therefore, dynamic nonprehensile manipulation may be considered as the most complex manipulation action. A standard procedure to deal with nonprehensile manipulation tasks is to split them in simpler subtasks. Each single subtask is often referred to as a *nonprehensile manipulation primitive* [3]. Examples of such primitives are sliding (or friction induced manipulation) [5], holonomic rolling [6], nonholonomic rolling [7], throwing [8], dynamic catching [9], batting [10], juggling [11], dribbling [12], and so on. Each primitive is equipped with its motion planner and controller. Within a complex manipulation task, a primitive can be activated or deactivated by a high-level supervisor [13].

Besides, nonprehensile manipulation involves one or more objects that should be adequately tracked for the successful execution of the task. Recognition and tracking

[2] https://www.youtube.com/user/ThePRISMAlab
[3] http://rtc.nagoya.riken.jp/RIBA/index-e.html
[4] https://cordis.europa.eu/project/rcn/98813_en.html

of deformable objects are even more challenging. Typically, object tracking is based on local feature detection, (i.e., the traditional Harris' corner detector [14], SIFT [15], SURF [16] or FAST [17]). However, some deformable objects, like a bread dough, usually have few features. Hence, their boundaries are the unique information, and they can be extracted by edge detectors [18, 19], or with depth sensors like the Kinect device. A real-time tracking method for a deformable object with RGB-D data is proposed in [20] by using a finite element method (FEM) framework for the tracking of the object. On the other hand, recognising a deformable object is possible without having a model in advance. The model of the object is instead identified during its manipulation directly [21]. A neural network-based recognition method for a deformable object is proposed in [22], given the tactile and the camera sensor information without any model in advance. Manipulating and observing the object is useful in [23] and [24] to reconstruct its dynamic properties, which are then used to plan its deformations [25]. As an application for perception and manipulation of a deformable object, it is worth mentioning the procedure for the automatic stretching of a dough in [26].

Fig. 1. The *RoDyMan* robotic platform equipped with two Schunk SVH hands as end-effectors.

2 Robot design

2.1 Mechatronic design

The *RoDyMan* project also gives the name to a 21-degree-of-freedom (DoF) humanoid-like robotic platform composed by a mobile base and an upper-body with two anthropomorphic arms and a sensorized head (see Fig. 1). The mobile base is developed with omnidirectional mecanum wheels allowing the robot to move on the floor in any direction. The four wheels are linked to the structure through an extensible actuated mechanism to increase the contact area during the execution of rapid movements of the upper body. Two redundant 7-DoF arms constitute the upper-body limbs. Two SCHUNK LWA 4P 6-DoF arms were chosen thanks to the following properties: (i) the high payload; (ii) the human-like distance between the shoulder and the elbow, and between the elbow and the wrist; (iii) the possibility to control the arm without any external controller by using a controller area network (CAN) bus communication. In order to add a human-like redundancy, a seventh joint SCHUNK PRL-100 is directly integrated into the shoulder, and it is controlled via another CAN channel. The two arms are linked to a 2-DoF torso made up by two SCHUNK PRL-120 motors to rotate and bow the upper body. Besides, a pan-tilt neck links the sensorized head to the torso. The arms come out with a large variety of different end-effectors such as two SCHUNK Servo-electric 5-finger SVH underactuated hands, or suitable 3-D printed tools specifically designed for the different experiments carried out with the robot.

From the perception point of view, the *RoDyMan* robot is equipped with force sensors that can be mounted on the wrist to measure the interaction forces with the environment. Two laser scanners are placed in the mobile platform for odometry and simultaneous localisation and mapping. Different vision sensors are placed in the head to reconstruct the environment and implement advanced control strategies. In details, the head contains an RGB-D sensor that exhibits a reasonable resolution with medium refresh rate, a time-of-flight sensor that has less resolution with high refresh rate, and two high-quality cameras that can be used in both mono or stereo configuration to have the maximum visual resolution.

The *RoDyMan* robot is a stand-alone system with both the controller and power supply directly integrated into the body. In details, two high capacity batteries packs and an uninterruptible power supply are located in the mobile base to provide the *RoDyMan* robot with an 8-hour autonomy. Moreover, a QNX-based personal computer is used for the low-level motor control and safety. A standard Linux-based personal computer is instead employed for perception and high-level planning and control.

Finally, a complete dynamic model of the 21-DoF system was carried out to allow the development of advanced control strategies for human-robot interaction. The complete dynamic model was derived in a symbolic form using an ad-hoc developed Matlab toolbox. Moreover, the linear matrix inequality identification method described in [27] was employed to obtain the dynamic parameters by including the physical consistency within the optimisation procedure. The obtained results show a reasonable estimation of parameters like mass and inertia. Because of its high predominance, especially in the static part, friction was instead identified separately and included as a constraint in the subsequent optimisation.

2.2 Software architecture

A high-level control architecture handles the autonomous execution of actions on the *RoDyMan* robotic platform. Such architecture aims at planning and executing the robot activities needed to perform complex dynamic manipulation tasks. The control architecture is depicted in Figure 2, and it is explained in the following.

High-level or dynamic manipulation tasks (e.g., the pizza tossing) are specified as inputs for the system using the *Human-Robot Interaction* (HRI) module. The task is then decomposed by the *Supervisor* module, splitting high-level actions into different lower level tasks considering both the state of the robot and the information generated by the *Perception* module. After the decomposition process, each lower level action can be executed. Examples of high-level tasks are *Grasp(Object)*, *Search(Object)* or *Toss(Object)*, and sequences of non-prehensile manipulation primitives. The *Supervision* module performs the high-level task decomposition process. It is endowed with a hierarchical decomposition tasks library, similar to the hierarchical task networks, which can be composed by the system to achieve the desired goal [28]. The *Executors* module is instead responsible for handling the execution of the tasks decomposed by the Supervisor. In this context, when a non-prehensile manipulation action is required, the desired low-level controller is chosen and invoked from the dynamic manipulation task list. For all the other actions, the *Executors* is directly responsible for their execution. In particular, path and motion planning functionalities are implemented in such a module to find an obstacle-free way for the end-effectors of the robot and its base to achieve the requested actions. The *Executors* module relies on MoveIT! [29], a framework that integrates and a universal robot description file, the Open Motion Planning Library, and other toolkits. The *Controller* module is instead responsible for sending the generated motion trajectories to the joints of the robot and its base. Finally, the *Perception* module extracts information from the operative environment through image elaboration algorithms.

In conclusion, the described control architecture represents a suitable choice to support the *RoDyMan* robot. It statically allocates the best low-level controller to accomplish the desired dynamic manipulation task. The methodology represents an innovative way of decomposing high-level tasks for nonprehensile manipulation with respect to the available literature [13].

3 Manipulation planning for deformable objects

Unlike solid objects, it is difficult to predict the shape of a deformable object after applying external forces. Hence, in order to plan manipulation actions, it is favourable having at disposition a precise model of the object. The following subsection is based on the smoothed particle hydrodynamics (SPH) framework. The pizza dough is addressed as an illustrative example. Moreover, the manipulation of a deformable object also requires an accurate perception module that can extract proper abstract information from the sensor devices. An RGB image-bases perception method is then explained in the following. Based on these modelling and perception methods, the manipulation planning for a pizza dough is briefly sketched.

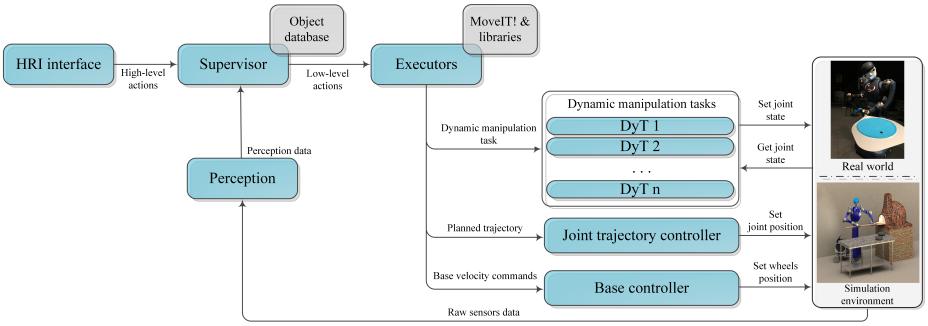


Fig. 2. *RoDyMan* high-level control architecture.

3.1 Modelling of deformable objects

The model of a deformable object is relevant for (i) compensating the lack of visual data from sensors; (ii) simplifying its representation; (iii) predicting its shape during the time.

A primary way to model a deformable object is to consider it as a sequence of solid objects linked by joints. An object modelled in this way is referred to as a foldable object. The deformability or flexibility of these objects is represented by how much the links can bend. The most typical method is based on a mass-spring-damper model [30, 31, 32, 33, 34], whose equations exploit Newton's second law and Hooke's law. More general models describing a deformable object regard it as a continuous object, which has a volume and some properties like elasticity or viscosity. According to the Navier-Stokes' theorem, that is a fundamental theorem representing continuous objects, the momentum conservation of an incompressible continuous object is formulated as

$$\rho \frac{\mathrm{D}\mathbf{v}}{\mathrm{D}t} = -\nabla \cdot p_h \mathbf{I}_r + \nabla \cdot \tau + \rho \mathbf{f}^{body}. \tag{1}$$

where ρ is density, $\mathbf{v} \in \mathbb{R}^r$ is velocity vector, $\mathbf{I}_r \in \mathbb{R}^{r \times r}$ is identity matrix, $\frac{\mathrm{D}}{\mathrm{D}t}$ is the material derivative operator, $\nabla \cdot$ is the divergence operator, $p_h \geq 0$ is the hydrostatic pressure, $\tau \in \mathbb{R}^{r \times r}$ is the viscous stress, $\mathbf{f}^{body} \in \mathbb{R}^r$ is the body force due to gravity, surface tension, or friction, and r is a proper dimension (usually $r = 3$). Among the various formulations for describing the Navier-Stokes' theorem (i.e., the finite difference method [35], the finite volume method [36], the finite element method [37, 38], and the gradient smoothing method [39]), the SPH method [40, 41] is one of the well-known formulations describing continuous objects. A physical quantity $\mathbf{A}$ (e.g., pressure, temperature, force) of any point $\mathbf{x} \in \mathbb{R}^r$ of a continuous object can be calculated by

$$\mathbf{A}(\mathbf{x}) = \int_{-\infty}^{\infty} \mathbf{A}(\mathbf{x}') W(\mathbf{x} - \mathbf{x}', h)\, d\mathbf{x}', \tag{2}$$

where $W : \mathbb{R}^r \to \mathbb{R}^{\geq 0}$ is an interpolating kernel with dimension $r = 3$, and h is the radius of the kernel domain. The value for the outer of the kernel domain is zero. The simplicity

of the formulation and the virtue of mesh-free Lagrangian models gave popularity to the SPH method within the domain of modelling continuous objects.

The pizza dough is here considered as illustrative example for a deformable object. A pizza dough is a rheological continuous material composed of various ingredients such as water, heavy water, glutamine residues, salts, agents affecting disulfide bonding, and the protein subunits [42]. The dough has high viscous properties because of gliadin, and elastic properties due to the presence of the glutenin [43, 44]. The viscosity is defined by the viscous stress $\tau = \mu \frac{\partial u}{\partial y}$ in (1), where μ is the dynamic viscosity coefficient, and $\frac{\partial u}{\partial y}$ is the local shear velocity. Unlike ideal Newtonian fluids having a constant dynamic viscosity coefficient, a general non-Newtonian fluid has a dynamic viscosity coefficient which depends on the local shear velocity. It is thus defined as a nonlinear equation. Several models exist for a non-Newtonian fluid like the Bingham plastic, the shear thinning, the shear thickening, or the Herschel-Bulkley-Papanastasiou model. The latter is suitable for rheological materials [45, 46, 47], whose equations are

$$\begin{cases} \tau = K|\dot{\gamma}|^{n-1}\dot{\gamma} \pm \tau_y, & \text{for } |\tau| > \tau_y, \\ \dot{\gamma} = 0, & \text{otherwise}, \end{cases} \tag{3}$$

where $\dot{\gamma} = \frac{\partial u}{\partial y}$. For a bread dough, it is experimentally known that $K = 5177$, $n = 0.417$, and $\tau_y = 298.76$ [47].

Based on the SPH formulation, the viscosity term in the momentum conservation (1) was designed and proposed in [40, 48, 49, 50, 51, 52, 53, 54, 55]. One of the representations for the viscosity force in a Newtonian fluid is

$$\mathbf{f}_i^{vis} = \frac{1}{\rho_i}\nabla \cdot \tau_i = \frac{1}{\rho_i}\mu\nabla^2\mathbf{v}_i \approx \frac{1}{\rho_i}\mu\sum_j \frac{m_j}{\rho_j}(\mathbf{v}_j - \mathbf{v}_i)\nabla^2 W_{ij}, \tag{4}$$

where m_j is the mass of a neighbour j-particle. The first equality of (4) holds because of volume conservation, while the second approximate equality holds because of the discretization of the second golden rule of the SPH formulation [48]. Four different viscous fluids are simulated in Fig. 3.

Volume conservation is difficult to guarantee for the SPH method. Hence, an additional procedure is needed, like the implicit incompressible SPH formulation [54], or the divergence-free SPH method [56].

3.2 Detection of deformable objects

Perception of a deformable object roughly consists of two parts. The former is the image processing of camera sensor data. The latter is the representation of the status of the object. Perception methods highly depend on the sought application. Considering as an illustrative example a modelling clay, emulating the pizza dough, a specific detector is explained in the following

Some restrictions are considered to decrease the complexity: (i) the workspace is reduced by defining a bounding area (i.e., a plain rectangular plate, see Fig. 4(a)); (ii)

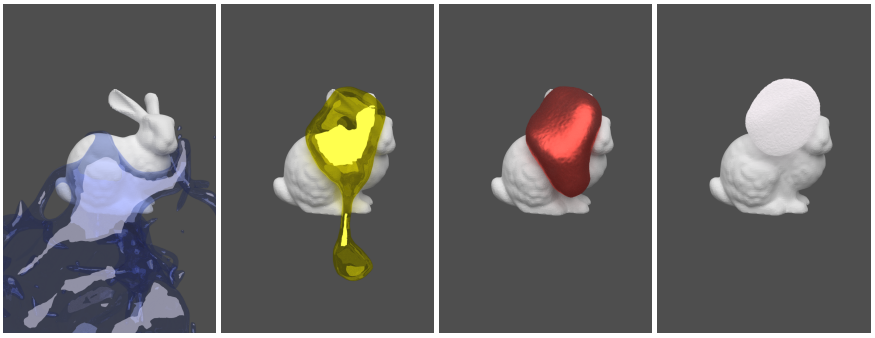


Fig. 3. Simulations of fluids with different dynamic viscosity coefficients. The fluids falls from above on the solid object representing a rabbit. From left to right the considered coefficients τ are 0 (water), 5 (honey), 70 (ketchup), and 250 (bread dough), respectively.

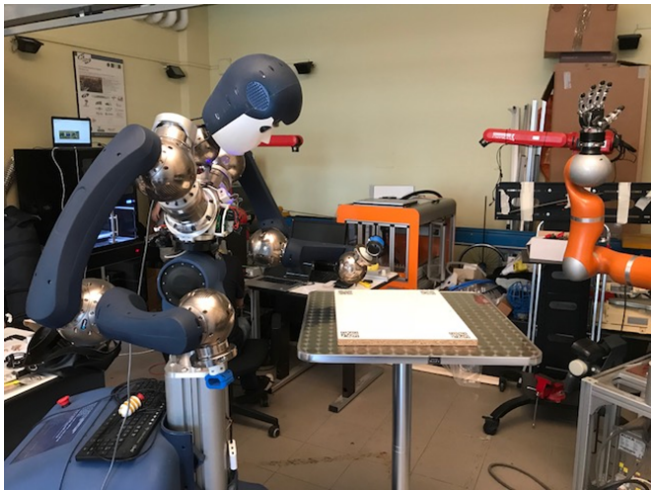
the contrast between the clay and the background is enhanced (e.g., in the considered case, a green clay is considered on a white background, see Fig. 4(b)).

Three corner marks on a rectangular plate, the absolute distance among them, and the kinematic information provided by the *RoDyMan* robot are enough to compute the frame transformation between a 2-D point in the camera image and the corresponding absolute position in the world frame [57], which is placed without loss o generality at the middle of the *RoDyMan* mobile base. Within the carried out experiments, QR codes are placed at each corner of the rectangular plate (see Fig. 4(a)). Such corner points are detected using an image matching with SIFT [15], SURF [16], or FAST [17]. In order to detect the corners quickly, AR codes and the corresponding AR code detector (i.e., ARToolkit [58]) are used. The detected corner points are also employed to remove outlines of the plate. Fig. 4(b) shows the detected green clay on a white plate which is identified by a blue contour. By supposing the object uniformly thick, except at the boundary margin, and using colour-based object and edge detectors [18], like the canny-edge detector [19], the modelling clay can be detected in real-time. Then, the corner points and the boundary of the deformable object in 2D camera image are transformed into the top-down orthogonal view of Fig. 4(c).

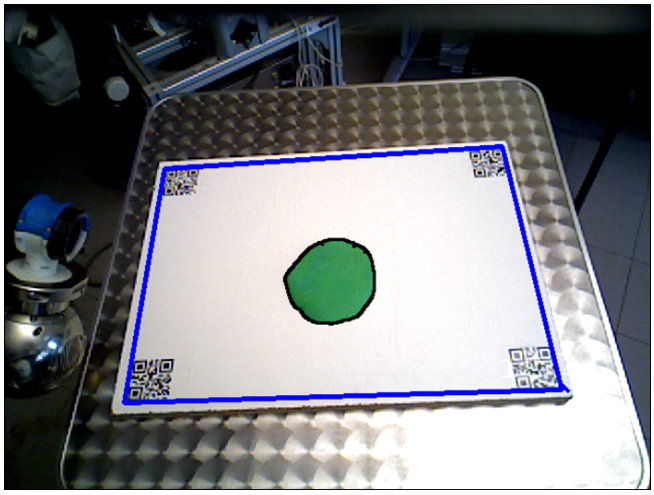
3.3 Task planning to stretch a deformable object by a tool

The illustrative example of stretching a pizza dough with a rolling-pin is addressed. In the experiments, a modelling clay is used in place of a real dough. The proposed procedure consists of four sequential steps: (i) object recognition; (ii) object deformation planning; (iii) tool action planning; (iv) robot manipulation planning. The flowchart is depicted in Fig. 5.

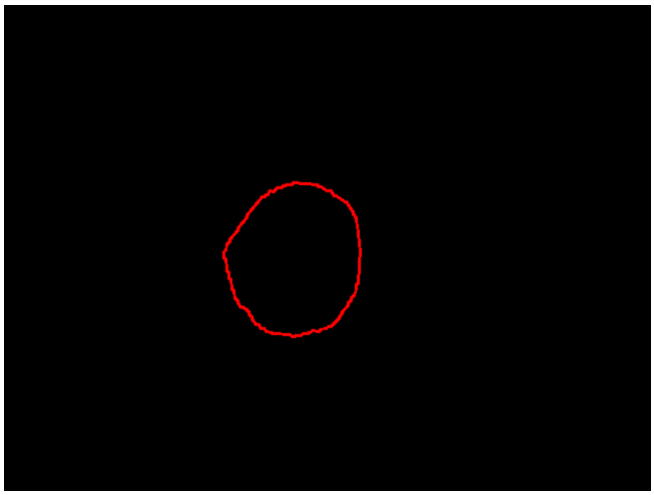
The recognition of the deformable object is explained in the previous subsection, and thus it is possible to reconstruct the current status of the modelling clay. By using the SPH model described in subsection 3.1, a transition look-up-table is built in simulation. The table contains a discretization of possible status and the transitions to other status given a rolling-pin's action. The object deformation planner generates a proper



(a) The *RoDyMan* robot and a white plate where the deformable object will be tracked.



(b) Detection of the background plate and the green modelling clay.



(c) Extracted modelling clay contour.

Fig. 4. Detection of a deformable object on a rectangular plate.

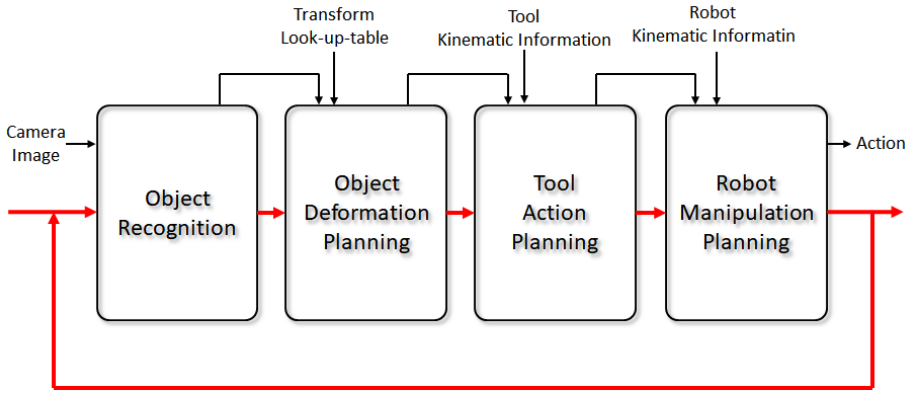


Fig. 5. Flowchart of the described procedure to stretch a modelling clay with a tool manipulated by a robot.

sequence of actions for the rolling-pin from the information of the current status of the object to bring it to the desired one. Based on such a sequence of actions, the movement of the rolling-pin and, in turn, the movements of the robot are planned. An example is shown in Fig. 6. The robot starts at an initial pose. The movement is planned to bring it towards a starting pose. From the current status of the object and its desired configuration, some stretching action is performed with the rolling-pin, and a final pose is reached. The sequence is repeated until the deformable object reaches the desired shape.

4 Nonprehensile manipulation: planning and control

Nonprehensile manipulation is maybe the most complex and dexterous manipulation task performed by a human being. In robotics, a literature analysis reveals that the conventional way to cope with a nonprehensile manipulation task is to split it into simpler subtasks [4]. Such subtasks are usually referred to as *nonprehensile manipulation primitives*. Three manipulation primitives are described in the following: (i) nonprehensile rolling, both holonomic and nonholonomic; (ii) friction-induced manipulation; (iii) throwing/tossing. The first primitive takes into account only rigid objects, while the second can instead consider both rigid and deformable objects. The latter addresses a deformable object explicitly. In the following, the controller and/or the motion planner designed for the primitives above are described.

4.1 Holonomic rolling

In this subsection, an actuated manipulator of a given shape that handles an object is considered. The manipulator is referred to as the hand and manipulates the object through rotations only, that is without grasping or caging it. In this class of manipulation, the object can only roll along the hand surface. Only planar holonomic rolling is thus presented.

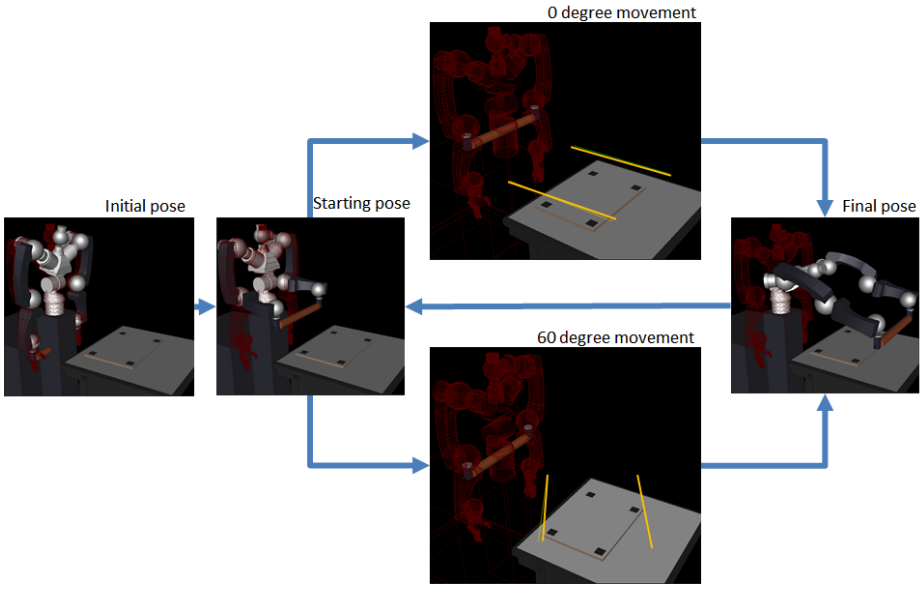


Fig. 6. Example of task planning actions to stretch a deformable object.

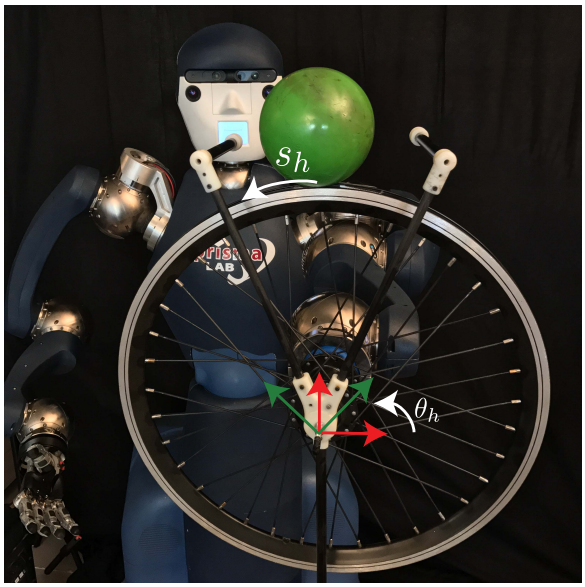


Fig. 7. The *RoDyMan* platform balancing a ball on a disk. The structure made by the three connected bars shown in the figure allows starting the experiments in a non-equilibrium position. The world frame and the body-fixed frame, attached to the rotating wheel, are displayed in red and green lines, respectively. The angle between these two frames is represented by θ_h.

The *RoDyMan* mechatronic platform uses highly-geared harmonic drives, which motivate the assumption that the acceleration of the hand $a_h = \ddot{\theta}_h \in \mathbb{R}$ con be considered as a control input for the system. Under the additional assumption that the hand can only rotate around its centre of mass, the dynamics of a nonprehensile planar rolling manipulation system can be described as follows

$$\ddot{\theta}_h = a_h, \tag{5a}$$

$$\ddot{s}_h = -b_{22}^{-1}\left(b_{12}a_h + c_{21}\dot{\theta}_h + c_{22}\dot{s}_h + g_2\right), \tag{5b}$$

where the parameters $b_{12} \in \mathbb{R}$ and $b_{22} \in \mathbb{R}$ are entries of the inertia matrix of the system $B \in \mathbb{R}^{2\times 2}$, $c_{21} \in \mathbb{R}$ and $c_{22} \in \mathbb{R}$ are entries of the (2×2) Coriolis matrix, and $g_2 \in \mathbb{R}$ describe the effect of the gravity. The contact position of the object on the hand, whose shape is parametrized through arclength, is represented by the variable $s_h \in \mathbb{R}$. A detailed description of the system can be found in [59], where it is also shown that a nonprehensile planar rolling manipulation system is differentially flat with the output $\frac{b_{12}}{b_{22}}\theta_h + s_h$ if the Coriolis forces are zero. This condition is verified for the ball-on-disk (BoD) system. Note that the dynamics of the BoD and the disk-on-disk (DoD) are similar in the transversal plane [6].

The BoD system is a mechanical arrangement of ball that can roll on top of a disk as shown in Fig 7. In this case, the disk is the actuated hand and the ball is the object. The control task for the BoD is to balance the object and simultaneously drive the hand to a desired angular position. This control problem is solved using passivity control (PBC) and port-Hamiltonian (pH) theory. In its classical formulation, the PBC applied to nonprehensile rolling system aims at finding a controller for system (5a)-(5b) such that the dynamics of the closed loop takes the form

$$\begin{bmatrix}\dot{\mathbf{q}}\\ \dot{\mathbf{p}}\end{bmatrix} = \begin{bmatrix}\mathbf{O} & \mathbf{B}^{-1}\mathbf{B}_d\\ -\mathbf{B}_d\mathbf{B}^{-1} & \mathbf{J}_2(\mathbf{q},\mathbf{p}) - \mathbf{R}_d(\mathbf{q},\mathbf{p})\end{bmatrix}\nabla H_d(\mathbf{q},\mathbf{p}), \tag{6}$$

where $\mathbf{q} \in \mathbb{R}^2$ and $\mathbf{p} \in \mathbb{R}^2$ are the vectors of generalized coordinates and momenta, respectively, and $\mathbf{O}$ the zero matrix of proper dimensions. The function $H_d = \frac{1}{2}\mathbf{p}^\top\mathbf{B}_d^{-1}(\mathbf{q})\mathbf{p} + V_d(\mathbf{q}) \in \mathbb{R}$ represents the desired energy in closed loop, where $\mathbf{B}_d \in \mathbb{R}^{2\times 2}$ is the desired mass matrix and $V_d \in \mathbb{R}$ is the desired potential energy. The function $\mathbf{J}_2 \in \mathbb{R}^{2\times 2}$ and $\mathbf{R}_d \in \mathbb{R}^{2\times 2}$ represent the gyroscopic forces and damping of the closed loop, respectively. In addition, it is required that the desired energy can be used as Lyapunov function for the closed loop, and thus stability is ensured. Besides, detectability of the passive output will ensure asymptotically stability of the desired equilibrium. The detailed design of the controller can be found in [6] for the DoD example.

As a milestone, it is possible to affirm that holonomic rolling nonprehensile manipulation primitive can be successfully modelled through the pH formalism and, consequently, controlled with PBC approaches [60]. This is relevant because it means that there exists a unified framework at least for such a class of nonprehensile manipulation primitive.

4.2 Nonholonomic rolling

The typical example of nonholonomic rolling is the so-called ball and plate benchmarking system. In such an example, a ball is steered towards the desired position with

a given orientation by a moving plate actuated by a robot. A planning algorithm plus the controller designed within the pH formalism are presented in [7].

In this subsection, instead, a further interesting example of the nonholonomic rolling primitive, studied in the framework of the *RoDyMan* project, is the hoop-and-pole system depicted in Figure 4.2. In this figure, the following coordinate frames are shown: an inertial frame $\mathbf{o}_\mathrm{w}$—$\mathbf{x}_\mathrm{w}\mathbf{y}_\mathrm{w}\mathbf{z}_\mathrm{w}$, a frame attached to the pole $\mathbf{o}_\mathrm{p}$—$\mathbf{x}_\mathrm{p}\mathbf{y}_\mathrm{p}\mathbf{z}_\mathrm{p}$, a frame attached to the hoop $\mathbf{o}_\mathrm{h}$—$\mathbf{x}_\mathrm{h}\mathbf{y}_\mathrm{h}\mathbf{z}_\mathrm{h}$, and a contact frame $\mathbf{o}_\mathrm{c}$—$\mathbf{x}_\mathrm{c}\mathbf{y}_\mathrm{c}\mathbf{z}_\mathrm{c}$. The contact frame is defined as follows: $\mathbf{o}_\mathrm{c} \in \mathbb{R}^3$ is the contact point, $\mathbf{x}_\mathrm{c} \in \mathbb{R}^3$ is a versor passing through the contact point and pointing outwards the pole surface, $\mathbf{y}_\mathrm{c} \in \mathbb{R}^3$ a versor in the intersection of the pole surface tangent plane at the contact point and the hoop equatorial plane, and $\mathbf{z}_\mathrm{c} \in \mathbb{R}^3$ defined to compose an orthonormal frame.

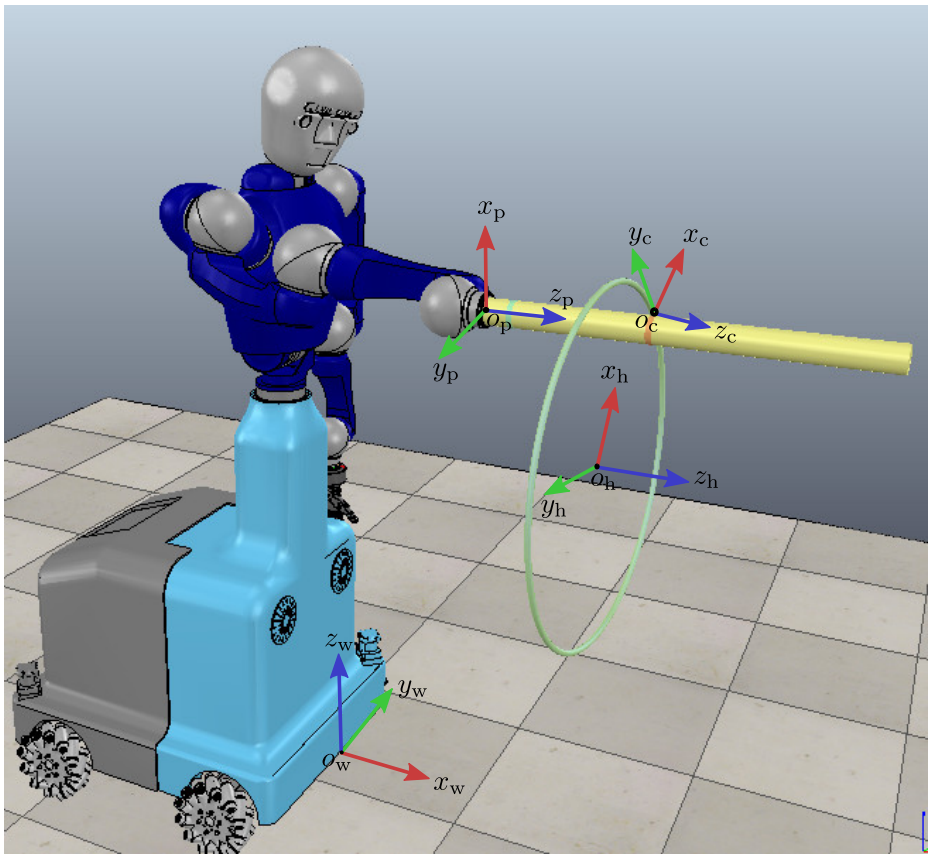


Fig. 8. Pole and hoop system driven by the *RoDyMan* robot in a simulation environment.

To obtain the contact kinematics [61, Section 6.2], the following contact coordinates are defined: $z_o \in \mathbb{R}$ is the distance to the contact point measured over the $\mathbf{z}_p$ axis, $\theta \in \mathbb{R}$ is the angle from one arbitrary point on the pole surface to the contact point measured over the $\mathbf{z}_p$ axis, $\gamma \in \mathbb{R}$ is the angle from one arbitrary point on the hoop surface to the contact point over the $\mathbf{z}_h$ axis, $\psi \in \mathbb{R}$ is the angle from an arbitrary point on the equator of the hoop to the contact point measured over the $\mathbf{y}_c$ axis, and $\phi \in \mathbb{R}$ is the angle between two tangent vectors, one of each surface, measured over the $\mathbf{x}_c$ axis (see [62] for details). These contact coordinates are grouped in $\mathbf{q}_c \triangleq \left[\gamma\ \psi\ z_o\ \theta\ \phi\right]^T \in \mathbb{R}^5$. Moreover, let $\mathbf{q}_p \in \mathbb{R}^m$ be the pole coordinates vector, where $m \leq 6$ is the number of the pole degrees of freedom. The complete generalised coordinates for the system under consideration are given by the vector $\mathbf{q} \triangleq \left[\mathbf{q}_c\,;\,\mathbf{q}_p\right] \in \mathbb{R}^{5+m}$, where $\left[\mathbf{x};\mathbf{y}\right]$ is a shorthand notation for $\left[\mathbf{x}^T\,\mathbf{y}^T\right]^T$. Finally, let $\mathbf{q}_h \triangleq \left[\gamma\ \psi\right]^T \in \mathbb{R}^2$ be the vector of hoop coordinates, and $\mathbf{q}_r \triangleq \left[\mathbf{q}_h\,;\,\mathbf{q}_p\right] \in \mathbb{R}^{2+m}$. From these definitions and by following the same modeling procedure as in [62], the following dynamic model is obtained

$$\mathbf{B}_h(\mathbf{q})\ddot{\mathbf{q}}_h + \mathbf{c}_h(\mathbf{q},\dot{\mathbf{q}}_r) + \mathbf{T}_h(\mathbf{q})\ddot{\mathbf{q}}_p = \mathbf{0} \tag{7}$$

$$\mathbf{B}_p(\mathbf{q})\ddot{\mathbf{q}}_p + \mathbf{c}_p(\mathbf{q},\dot{\mathbf{q}}_r) + \mathbf{T}_h^T(\mathbf{q})\ddot{\mathbf{q}}_h = \mathbf{u}\,, \tag{8}$$

subject to the constraints

$$\dot{\theta} = \left(l_h c_\phi / r_p\right)\dot{\gamma} \tag{9}$$

$$\dot{z}_o = -l_h s_\phi \dot{\gamma} \tag{10}$$

$$\dot{\phi} = -s_\psi \dot{\gamma}, \tag{11}$$

where $l_h > 0$ is the hoop radius, $r_p > 0$ is the pole radius, and s_x and c_x are shorthand notations for $\sin(x)$ and $\cos(x)$, respectively. In the above equations $\mathbf{B}_h(\mathbf{q}) \in \mathbb{R}^{2\times 2}$ and $\mathbf{B}_p(\mathbf{q}) \in \mathbb{R}^{m\times m}$ are symmetric positive definite matrices, $\mathbf{c}_h(\mathbf{q},\dot{\mathbf{q}}_r) \in \mathbb{R}^2$ and $\mathbf{c}_p(\mathbf{q},\dot{\mathbf{q}}_r) \in \mathbb{R}^m$ are vectors accounting for centripetal, Coriolis and gravitational forces, $\mathbf{u} \in \mathbb{R}^m$ is the vector of input forces acting on the pole, and $\mathbf{T}_h(\mathbf{q}) \in \mathbb{R}^{2\times m}$ is the inertia coupling matrix. If $\text{rank}(\mathbf{T}_h(\mathbf{q})) = 2, \forall \mathbf{q}$, the system is said to be *strong inertially coupled* [63], which is an assumption in this work.

The control objective is to rotate the hoop at a constant speed $\dot{\gamma}_d \in \mathbb{R}$ while simultaneously translating the hoop to an arbitrarily desired point over the pole surface $z_{od} \in \mathbb{R}$. Moreover, the pole coordinates must be stabilized. The controllability of the whole system depends on the coordinates chosen for the pole. After a controllability analysis [64], the task can be accomplished if the pole degrees of freedom consist of two translations along the $\mathbf{x}_p$ and $\mathbf{y}_p$ axes and two rotations around the same axes. By exploiting the strong inertially coupled property, the main idea is to construct two orthogonal projection matrices form the $\mathbf{T}_h(\mathbf{q})$ matrix in (7)-(8), that is, $\mathbf{P}_h \triangleq \mathbf{T}_h^+\mathbf{T}_h \in \mathbb{R}^{m\times m}$ and $\mathbf{Q}_h \triangleq \mathbf{I}_m - \mathbf{P}_h \in \mathbb{R}^{m\times m}$, where $\mathbf{T}_h^+ \triangleq \mathbf{T}_h^T\left(\mathbf{T}_h\mathbf{T}_h^T\right)^{-1} \in \mathbb{R}^{m\times 2}$ is the Penrose's right pseudoinverse. It is straightforward to verify that the following relations hold: $\mathbf{P}_h\mathbf{T}_h^T = \mathbf{T}_h^T$, $\mathbf{T}_h\mathbf{P}_h = \mathbf{T}_h$, $\mathbf{Q}_h\mathbf{T}_h^T = \mathbf{O}$, and $\mathbf{T}_h\mathbf{Q}_h = \mathbf{O}$. Based in these projectors, the control input in (8) is designed as

$$\mathbf{u} = \mathbf{B}_p\left(\mathbf{P}_h\mathbf{u}_P + \mathbf{Q}_h\mathbf{u}_Q\right). \tag{12}$$

A direct application of noncollocated partial feedback linearisation [63], results in the decoupled double integrator system $\ddot{\mathbf{q}}_{\mathrm{h}} = \mathbf{v}_{\mathrm{P}}$ for controlling the hoop coordinates, where $\mathbf{v}_{\mathrm{P}} \in \mathbb{R}^2$ is a new input to be designed. Notice, however, that for designing this control law the nonholonomic constraints (9)-(11) must be taken into account. Furthermore, not only the coordinates in $\mathbf{q}_{\mathrm{h}}$ must be stabilised, but ϕ, θ and z_{o} as well. This can be carried out by employing the backstepping methodology, as explained in [64].

As regards to the control of the pole coordinates, the linearisation along the trajectories of the steady state solutions, resulting from applying the noncollocated partial linearisation controller for the hoop coordinates, leads to a periodic linear time-varying system. For such particular kind of systems, a linear-quadratic regulator controller can be employed [65], and the resulting periodic Riccati equation can be solved by the *quasi-linearisation* method [66, p. 137]. Through a formal mathematical analysis, it is possible to guarantee the ultimate boundedness of the hoop coordinates, with an arbitrary small ultimate bound, and the local stabilisation of the pole coordinates [64].

The proposed scheme is validated through numerical simulations in [64]. The main challenges for carrying out experiments are the necessity of a speedy reconstruction of the hoop position and orientation and the required high speeds and accelerations for the actuators (robot joint motors).

4.3 Friction-induced manipulation

Positioning object by sliding is a problem studied in the literature over the last decade. For example, a mechanism called *universal planar manipulator* is designed in [67] for moving parts on the plane though friction induced movements. Later, the problem of positioning and orienting of a rigid body over a 2D plate is considered in [68]. More recently, a method for positioning and orienting a rigid body with a six-degrees-of-freedom rigid plate is proposed and successfully tested in [69].

In the context of the *RoDyMan* project, the illustrative example of turning a pizza over an actuated peel is tackled. To uniformly cook a pizza inside a wood-fired oven, in which the source of the heat is not distributed but localised in a specific position, it is necessary to turn around the pizza with a peel frequently. The movements inspired by Neapolitan pizza chefs are an example of friction-induced manipulation [70].

The problem of positioning and orientating a disk with a two-degrees-of-freedom manipulation system was first studied in [71]. In the cited article, the authors take advantage of the physical properties of the mechanical system to successfully drive the disk to an arbitrary desired position and orientation on the peel surface. The authors also show that a translation of the disk can be done without rotating it but not *vice versa*. One weak point of the cited work is that the control strategy is mainly carried out in an open loop fashion, that is, the visual feedback is only employed to indicate that the disk has arrived at a particular pose. In contrast, a model-based closed-loop controller for rotating the pizza at a desired constant speed over the peel surface is briefly proposed below.

In order to design the control strategy, a mathematical model of the system is firstly derived. Let $\mathbf{o}_{\mathrm{w}}$–$\mathbf{x}_{\mathrm{w}}\mathbf{y}_{\mathrm{w}}\mathbf{z}_{\mathrm{w}}$ be the inertial frame, $\mathbf{o}_{\mathrm{h}}$–$\mathbf{x}_{\mathrm{h}}\mathbf{y}_{\mathrm{h}}\mathbf{z}_{\mathrm{h}}$ a frame attached to the peel and $\mathbf{o}_{\mathrm{p}}$–$\mathbf{x}_{\mathrm{p}}\mathbf{y}_{\mathrm{p}}\mathbf{z}_{\mathrm{p}}$ a frame attached to the pizza, as shown in Figure 9. The generalised coordinates for the peel, driven by a robotic manipulator, are $x_{\mathrm{h}} \in \mathbb{R}$ and $\theta \in \mathbb{R}$, where

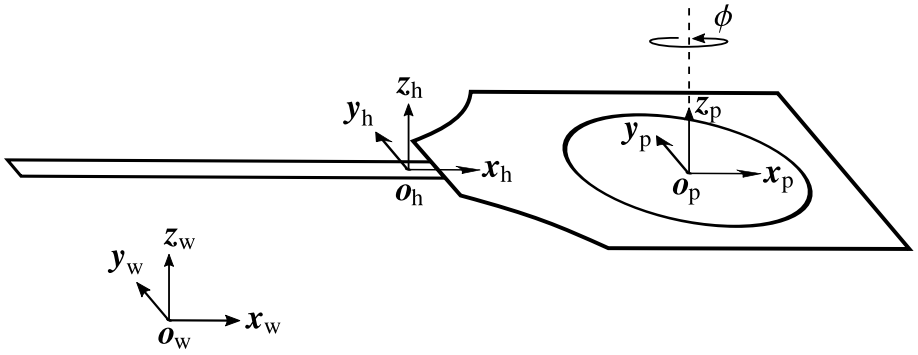


Fig. 9. Peel and pizza system.

x_h is the first component of $\mathbf{o}_h \in \mathbb{R}^3$ and θ is the rotation angle of the peel with respect to the inertial frame. The generalised coordinates for the pizza are chosen as $x_p, y_p, \phi \in \mathbb{R}$, where x_p and y_p are the first two components of $\mathbf{o}_p^h \in \mathbb{R}^3$ and ϕ is the angle of rotation of the pizza with respect to the $\mathbf{z}_p^h$ axis. Therefore, the configuration of the system is completely described by the vector $\mathbf{q} = \begin{bmatrix} x_h & \theta & x_p & y_p & \phi \end{bmatrix}^T \in \mathbb{R}^5$. Before applying the Euler–Lagrange methodology to obtain the equations of motion, the external and non conservative forces are described. Notice that the Coulomb friction terms play a critical role for this task. These terms are defined as functions of the relative velocities between the peel and the pizza $\dot{x}_p$ and $\dot{y}_p$, and are described by [71]

$$f_x = m_p g \mu_p \text{sign}(\dot{x}_p) \tag{13}$$

$$f_y = m_p g \mu_p \text{sign}(\dot{y}_p), \tag{14}$$

where $g \in \mathbb{R}$ is the gravity acceleration constant, $\mu_p \in \mathbb{R}$ is the Coulomb friction coefficient, and $\text{sign}(x)$ is the standard *sign* function. On the other hand, there is a torque over the $\mathbf{z}_p^h$ axis produced by the movement of the x_h coordinate and the change of the pressure distribution which in turn is induced by the acceleration on the θ coordinate [71], and is given by $\tau_\phi = \mu_\phi I_{px} \text{sign}(\dot{x}_p)\ddot{\theta}$, where $I_{px} \in \mathbb{R}$ is the $(1,1)$ component of the pizza inertia tensor $\mathbf{I}_p \in \mathbb{R}^{3\times 3}$. Overall, the non-conservative and external forces are represented by the vector $\xi = \begin{bmatrix} u_h & \tau_\theta & f_x & f_y & \tau_\phi \end{bmatrix}^T \in \mathbb{R}^5$, where $u_h \in \mathbb{R}$ is the external force applied over the peel in the $\mathbf{x}_h$ direction, and $\tau_\theta \in \mathbb{R}$ is the external torque over the same axis.

A further simplification can be made if it is considered the linear and angular accelerations of the peel as inputs, i.e., $\mathbf{u} \triangleq \begin{bmatrix} u_h & u_\theta \end{bmatrix}^T \triangleq \begin{bmatrix} \ddot{x}_h & \ddot{\theta} \end{bmatrix}^T \in \mathbb{R}^2$. In order to employ continuous tools to analyze the system dynamics, the following approximation of the sign function $\text{sign}(x_i) \approx \tanh(k_i x_i)$ is made, where k is a positive constant. Since the objective of the control is also the pizza rotation speed, the error following error is defined $\tilde{\dot{\phi}} = \dot{\phi} - \dot{\phi}_d$, where $\dot{\phi}_d \in \mathbb{R}$ is the desired pizza rotation speed. Now, let $\mathbf{x} \triangleq \begin{bmatrix} x_1 & x_2 & \cdots & x_9 \end{bmatrix}^T \triangleq \begin{bmatrix} x_h & \theta_h & x_p & y_p & \dot{x}_h & \dot{\theta}_h & \dot{x}_p & \dot{y}_p & \tilde{\dot{\phi}} \end{bmatrix}^T \in \mathbb{R}^9$ be the state space vector. Then,

the system dynamics can be put in the form

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}_1 u_1 + \mathbf{g}_2(\mathbf{x}) u_2 \,, \tag{15}$$

where $\mathbf{f}(\mathbf{x}) = \begin{bmatrix} x_5 & x_6 & x_7 & x_8 & \mathbf{f}_2(\mathbf{x}) \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^9$, $\mathbf{g}_1 = \begin{bmatrix} \mathbf{0} & \mathbf{g}_{12} \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^9$, and $\mathbf{g}_2(\mathbf{x}) = \begin{bmatrix} \mathbf{0} & \mathbf{g}_{22}(\mathbf{x}) \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^9$, with

$$\mathbf{f}_2(\mathbf{x}) = \begin{bmatrix} 0 \\ 0 \\ -\frac{b_{\mathrm{p}}}{m_{\mathrm{p}}} x_7 - g\mu_{\mathrm{p}} \tanh(k_7 x_7) \\ -\frac{b_{\mathrm{p}}}{m_{\mathrm{p}}} x_8 - g\mu_{\mathrm{p}} \tanh(k_8 x_8) - g\sin(x_2) + x_4 x_6^2 \\ 0 \end{bmatrix} \in \mathbb{R}^5 \tag{16}$$

$$\mathbf{g}_{12} = \begin{bmatrix} 1 & 0 & -1 & 0 & 0 \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^5 \tag{17}$$

$$\mathbf{g}_{22}(\mathbf{x}) = \begin{bmatrix} 0 & 1 & 0 & 0 & -\mu_\phi \left(I_{\mathrm{px}} / I_{\mathrm{pz}} \right) \tanh(k_7 x_7) \end{bmatrix}^{\mathrm{T}} \,, \in \mathbb{R}^5 \tag{18}$$

where $I_{\mathrm{pz}} \in \mathbb{R}$ is the $(3,3)$ component of $\mathbf{I}_{\mathrm{p}}$.

The control objective consists in inducing a rotating movement on the pizza dough at a desired angular speed $\dot{\phi}_{\mathrm{d}}$ while keeping the remaining coordinates as close to zero as possible. In order to fulfil this objective, the following control law is proposed

$$u_1 = -k_1 x_1 - k_5 x_5 + a_{\mathrm{h}} \sin(\omega_{\mathrm{h}} t) \tag{19}$$

$$u_2 = \frac{\tanh(k_7 x_7) I_{\mathrm{pz}}}{\mu_\phi I_{\mathrm{px}}} k_9 x_9 - k_2 x_2 - k_6 x_6 \,. \tag{20}$$

The first term of (19) is a PD control, employed to stabilise the peel linear movement plus a feedforward term $a_{\mathrm{h}} \sin(\omega_{\mathrm{h}} t)$, to ensure the condition $\dot{x}_{\mathrm{p}} \neq 0$. On the other hand, the control law (20) is a PD control to stabilise the peel orientation plus a nonlinear term to induce a rotation in the pizza by exploiting the torque τ_ϕ.

The centripetal forces term $x_4 x_6^2$ in (18) is neglected, as commonly assumed in the literature [69, 71]. Therefore, a closed-loop analysis can be carried out to guarantee the existence of periodic solutions of the peel coordinates around the equilibrium position. Besides, the stabilisation of the pizza centre of mass around zero is also ensured, as well as the practical stabilisation of the angular speed tracking error $\tilde{\phi}$, with arbitrary small tracking error.

4.4 Tossing of a deformable object

The nonprehensile throwing manipulation primitive aims at bringing an object to the desired location (with eventually a desired orientation) which is out of the workspace for the robot. The task becomes even more challenging if the thrown object is deformable. Again, the illustrative problem of tossing and catching a pizza dough is considered within the *RoDyMan* project. Such a procedure is frequently dexterously performed by human pizza-chefs. There are at least three reasons why tossing the dough during the preparation of the pizza is attractive: (i) the dough is stretched to the desired size; (ii) the dough naturally assumes a configuration that is thicker at the ends and thinner in the

middle; (iii) as the spinning dough freely falls, the outside of the dough dries, making it crunchy in the outside but light in the middle.

The combined model of a dough grasped with robotic fingers through unilateral constraints, and the kinematics and dynamics of the robot manipulator were derived in [8]. Upon that, a control law achieving the desired tossing motion can be designed. Besides, with a perfect knowledge of the motion of the dough, optimal trajectories can be generated in SE(3) for the catching phase. The optimal trajectory generation is repeated as new sensor information is available. The optimal trajectories are generated in such a way that the initial position, velocity, acceleration, and final velocity and accelerations are matched. Therefore, it is at least thrice continuously differentiable. An optimal trajectory whose initial and final accelerations are desired to satisfy a sixth-order boundary value problem (BVP). Such a BVP is generated by using the necessary conditions for a path to minimise a convex combination of the jerk and acceleration functionals. While minimising the jerk functional reduces the vibrations in the structure of the robotic manipulator, minimising the acceleration functional reduces the total amount of energy expended during the catching motion [72].

To derive the trajectories for tossing and catching, the following cost functional $L_c : TQ \times TQ \to \mathbb{R}$ is minimised

$$L_c(s) = \int_{t_0}^{t_f + s\delta t_f} \alpha \langle\!\langle \frac{D^2\mathbf{v}}{\mathrm{d}t}, \frac{D^2\mathbf{v}}{\mathrm{d}t} \rangle\!\rangle + \beta \langle\!\langle \frac{D\mathbf{v}}{\mathrm{d}t}, \frac{D\mathbf{v}}{\mathrm{d}t} \rangle\!\rangle \, \mathrm{d}t,$$

where the weights $\alpha \in \mathbb{R}$ and $\beta \in \mathbb{R}$ satisfy $\alpha + \beta = 1$, $t_0 \in \mathbb{R}$ and $t_f \in \mathbb{R}$ are the initial and final time instants, $\mathbf{v} \in \mathfrak{se}(3)$ is the end-effector twist, and $\langle\!\langle \cdot, \cdot \rangle\!\rangle$ is a metric on the Riemannian manifold. The symbol $\frac{D}{\mathrm{d}t}$ here denotes the covariant derivative of a tangent vector along a curve [73]. Only the case where the final position is left free and is part of the minimization problem is considered. The first variation of this functional is calculated, taking special care of the free endpoint conditions. More details can be found in [8].

Experimental validations are in progress. Nevertheless, preliminary results show that such kind of task requires high peak currents in the motors to toss the dough for more than 10 cm. The motors of the *RoDyMan* robot do not have such skills and must be improved as highlighted in the next section.

5 Discussion and future work

As sketched in the introduction, the illustrative example of manipulating a pizza dough is a media expedient. It is clear that if a robot can manipulate a pizza dough, it might be able to perform skilfull manipulation tasks.

Therefore, many achievements have been reached by the *RoDyMan* project. Nevertheless, many improvements are instead needed to be tackled. For instance, the *RoDyMan* robot must be revised to cope with the issues given by the high-velocity required by some nonprehensile manipulation task. Friction is, in general, the main problem affecting the actuators of the *RoDyMan* robot. A more accurate parametric estimation and robust controllers are essential. Besides, many assumptions are often made within a nonprehensile manipulation task to avoid the almost intrinsic non-smooth behaviour

of it. Such assumptions make a nonprehensile manipulations system like a prehensile one. The proof that the designed controller does not violate such assumptions is usually performed a-posteriori. The contact forces should be directly addressed within the model-based approach to avoid this aspect. This is indeed a future research direction leading to non-smooth and hybrid controllers.

Future development for the *RoDyMan* project also includes the application of the reached goals in other domains. For instance, the perception of elastic objects is currently being applied in the medical context to shape variations of muscles and organs. The manipulation performed while tossing the deformable dough is currently under investigation to improve the automation of gluing the shoes' lower surfaces.

From a theoretical point of view, the *RoDyMan* project may also be improved in the research of creating a general unified framework determining the design of ad-hoc controller to solve each nonprehensile manipulation primitive. The pH formalism and the PBC can be used to solve the holonomic nonprehensile rolling, but it is not enough. Moreover, a high-level supervisor enabling the correct switching between the different nonprehensile manipulation primitives must be addressed. A recent attempt is carried out in [13]. For instance, learning-based approaches may be helpful to design such a supervisor since task simplification, and human-inspired control strategies may be the key towards the fulfilling of the whole complex nonprehensile manipulation task.

Acknowledgements

The research leading to these results has been supported by the RoDyMan project, which has received funding from the European Research Council FP7 Ideas under Advanced Grant agreement number 320992.

Bibliography

[1] D. Prattichizzo and J.C. Trinkle. Grasping. In Bruno Siciliano and Oussama Khatib, editors, *Springer Handbook of Robotics*, pages 955–988. Springer International Publishing, 2016.

[2] K. M. Lynch and M. T. Mason. Dynamic nonprehensile manipulation: Controllability, planning, and experiments. *International Journal of Robotics Research*, 18(1):64–92, 1999.

[3] K. M. Lynch and T. D. Murphey. Control of nonprehensile manipulation. In Antonio Bicchi, Domenico Prattichizzo, and HenrikIskov Christensen, editors, *Control Problems in Robotics*, volume 4 of *Springer Tracts in Advanced Robotics*, pages 39–57. Springer Berlin Heidelberg, 2003.

[4] F. Ruggiero, V. Lippiello, and B. Siciliano. Nonprehensile dynamic manipulation: A survey. *IEEE Robotics and Automation Letters*, 3(3):1711–1718, 2018.

[5] T.H. Vose, P. Umbanhowar, and K.M. Lynch. Friction-induced velocity fields for point parts sliding on a rigid oscillated plate. *The International Journal of Robotics Research*, 28(8):1020–1039, 2009.

[6] A. Donaire, F. Ruggiero, L.R. Buonocore, V. Lippiello, and B. Siciliano. Passivity-based control for a rolling-balancing system: The nonprehensile disk-on-disk. *IEEE Transactions on Control Systems Technology*, 25(6):2135–2142, 2016.

[7] D. Serra, J. Ferguson, F. Ruggiero, A. Siniscalco, A. Petit, V. Lippiello, and B. Siciliano. On the experiments about the nonprehensile reconfiguration of a rolling sphere on a plate. In *26th Mediterranean Conference on Control and Automation*, Zadar, HR, 2018.

[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*, pages 3932–3939, Stockholm, SE, 2016.

[9] G. Bätz, A. Yaqub, H. Wu, K. Kuhnlenz, D. Wollherr, and M. Buss. Dynamic manipulation: Nonprehensile ball catching. In *18th Mediterranean Conference on Control and Automation*, pages 365–370, Marrakech, MA, 2010.

[10] D. Serra, A.C. Satici, F. Ruggiero, V. Lippiello, and B. Siciliano. An optimal trajectory planner for a robotic batting task: the table tennis example. In *13th International Conference on Informatics in Control, Automation and Robotics*, pages 90–101, Lisbon, PT, 2016.

[11] P. Reist and R. D'Andrea. Design and analysis of a blind juggling robot. *IEEE Transactions on Robotics*, 28(6):1228–1243, 2012.

[12] G. Bätz, U. Mettin, A. Schimdts, M. Scheint, D. Wollherr, and A. Shiriaev. Ball dribbling with an underactuated continuous-time control phase: Theory & experiments. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2890–2895, Taipei, TW, 2010.

[13] J.Z. Woodruff and K.M. Lynch. Planning and control for dynamic, nonprehensile, and hybrid manipulation tasks. In *2017 IEEE International Conference on Robotics and Automation*, pages 4066–4073, Singapore, 2017.

[14] C. Harris and M. Stephens. A combined corner and edge detector. In *4th Alvey vision conference*, pages 147–151, Manchester, UK, 1988.
[15] D. G. Lowe. Object recognition from local scale-invariant features. In *International Conference on Computer Vision*, 2, pages 1150–1157, Kerkyra, GR, 1999.
[16] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *European Conference on Computer Vision*, pages 404–417, Graz, AT, 2006.
[17] E. Rosten and T. Drummond. Machine learning for high-speed corner detection. In *European Conference on Computer Vision*, pages 430–443, Graz, AT, 2006.
[18] R. C. Gonzalez and R. E. Woods. *Digital Image Processing (3rd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2006.
[19] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, 1986.
[20] A. Petit, V. Lippiello, and B. Siciliano. Real-time tracking of 3D elastic objects with an RGB-D sensor. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3914–3921, Hamburg, GE, 2015.
[21] J. Lang. An acquisition method for interactive deformable models. In *Second International Conference on Creating, Connecting and Collaborating through Computing*, pages 165–170, Kyoto, JP, 2004.
[22] A.-M. Cretu, P. Payeur, and E.M. Petriu. Neural network mapping and clustering of elastic behavior from tactile and range imaging for virtualized reality applications. *IEEE Transactions on Instrumentation and Measurement*, 57(9):1918–1928, 2008.
[23] B. Frank, R. Schmedding, C. Stachniss, M. Teschner, and W. Burgard. Learning the elasticity parameters of deformable objects with a manipulation robot. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1877–1883, Taipei, TW, 2010.
[24] B. Frank, C. Stachniss, N. Abdo, and W. Burgard. Efficient motion planning for manipulation robots in environments with deformable objects. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2180–2185, San Francisco, CA, USA, 2011.
[25] C. Phillips-Grafflin and D. Berenson. A representation of deformable objects for motion planning with no physical simulation. In *2014 IEEE International Conference on Robotics and Automation*, pages 98–105, Hong Kong, CN, 2014.
[26] S. Tokumoto and S. Hirai. Deformation control of rheological food dough using a forming process model. In *2002 IEEE International Conference on Robotics and Automation*, pages 1457–1464, Washington, DC, USA, 2002.
[27] C. D. Sousa and R. Cortesão. Physically feasibility of robot base inertial parameters identification: A linear matrix inequality approach. *International Journal of Robotics Research*, 33(6):931–944, 2014.
[28] J. Cacace, A. Finzi, V. Lippiello, G. Loianno, and D. Sanzone. Aerial service vehicles for industrial inspection: Task decomposition and plan execution. *Applied Intelligence*, 42:49–62, 2015.
[29] S. Chitta, I.A. Sucan, and S. Cousins. Moveit! [ros topics]. *IEEE Robotics and Automation Magazine*, 19:18–19, 2012.
[30] M. Higashimori, K. Yoshimoto, and M. Kaneko. Active shaping of an unknown rheological object based on deformation decomposition into elasticity and plastic-

ity. In *2010 IEEE International Conference on Robotics and Automation*, pages 5120–5126, Anchorage, AK, USA, 2010.

[31] S. Frisken, F. Gibson, and B. Mirtich. A survey of deformable modeling in computer graphics. Technical report, Brigham and Women's Hospital, 1997.

[32] E. Anshelevich, S. Owens, F. Lamiraux, and L. E. Kavraki. Deformable volumes in path planning applications. In *2000 IEEE International Conference on Robotics and Automation*, volume 3, pages 2290–2295, San Francisco, CA, USA, 2000.

[33] T. Wada, S. Hirai, S. Kawamura, and N. Kamiji. Robust manipulation of deformable objects by a simple PID feedback. In *2001 IEEE International Conference on Robotics and Automation*, pages 85–90, Seoul, KR, 2001.

[34] A Nealen, M. Müller, R. Keiser, R. Boxerman, and M. Carlson. Physically based deformable models in computer graphics. *Eurographics: State of the Art Report*, pages 71–94, 2005.

[35] T. J. Chung. *Computational Fluid Dynamics*. Cambridge university press, Cambridge, MA, USA, 2nd edition, Sep. 2010.

[36] R. Eymard, T. R. Gallouët, and R. Herbin. The finite volume method. *Handbook of Numerical Analysis*, 7:713–1020, 2000.

[37] O.C. Zienkiewicz, R.L. Taylor, and J.Z. Zhu. *The Finite Element Method Set*. Butterworth-Heinemann, Oxford, 6 edition, 2005.

[38] G. Irving, J. Teran, and R. Fedkiw. Invertible finite elements for robust simulation of large deformation. In *2004 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 131–140, Grenoble, FR, 2004.

[39] G. R. Liu, J. Zhang, K. Y. Lam, H. Li, G. Xu, Z. H. Zhong, G. Y. Li, and X. Han. A gradient smoothing method (GSM) with directional correction for solid mechanics problems. *Computational Mechanics*, 41(3):457–472, 2008.

[40] J. J. Monaghan. Smoothed particle hydrodynamics. *Reports on Progress in Physics*, 68:1703–1759, August 2005.

[41] M. B. Liu and G. R. Liu. Smoothed particle hydrodynamics (SPH): an overview and recent developments. *Archives of Computational Methods in Engineering*, 17(1):25–76, 2010.

[42] M. R. Amjid, A. Shehzad, S. Hussain, M. A. Shabbir, M. R. Khan, and M. Shoaib. A comprehensive review on wheat flour dough rheology. *Pakistan Journal of Food Sciences*, 23:105–123, 2013.

[43] H. Wieser. Chemistry of gluten proteins. *Food Microbiology*, 24(2):115–119, 2007.

[44] F. Van Bockstaele, I. De Leyn, M. Eeckhout, and K. Dewettinck. Rheological properties of wheat flour dough and the relationship with bread volume. i. creep-recovery measurements. *Cereal Chemistry Journal*, 85(6):753, 2008.

[45] Souzanna Sofou, Edward B. Muliawan, Savvas G. Hatzikiriakos, and Evan Mitsoulis. Rheological characterization and constitutive modeling of bread dough. *Rheologica Acta*, 47(4):369–381, 2008.

[46] E. Mitsoulis. Numerical simulation of calendering viscoplastic fluids. *Journal of Non-Newtonian Fluid Mechanics*, 154:77–88, 2008.

[47] E. Mitsoulis and S. G. Hatzikiriakos. Rolling of bread dough: Experiments and simulations. *Food and Bioproducts Processing*, 87(2):124–138, 2009.

[48] J. J. Monaghan. Smoothed particle hydrodynamics. *Annu. Rev. Astron. Astrophys.*, 30:543–574, 1992.

[49] J. J. Monaghan. Simulating free surface flows with SPH. *J. Comput. Phys.*, 110(2):399–406, February 1994.

[50] S. Shao and E.Y.M. Lo. Incompressible SPH method for simulating newtonian and non-newtonian flows with a free surface. *Advances in Water Resources*, 26(7):787–800, 2003.

[51] L. F. de Souza Andrade, M. Sandim, F. Petronetto, P. Pagliosa, and A. Paiva. SPH fluids for viscous jet buckling. In *27th Conference on Graphics, Patterns and Images*, pages 65–72, Rio de Janeiro, BR, 2014.

[52] L. F. de Souza Andrade, M. Sandim, F. Petronetto, P. Pagliosa, and A. Paiva. Particle-based fluids for viscous jet buckling. *Computers & Graphics*, 52:106–115, 2015.

[53] T. Takahashi, Y. Dobashi, I. Fujishiro, T. Nishita, and M. C. Lin. Implicit formulation for SPH-based viscous fluids. *Computer Graphics Forum*, 34(2):493–502, 2015.

[54] A. Peer, M. Ihmsen, J. Cornelis, and M. Teschner. An implicit viscosity formulation for SPH fluids. *ACM Transactions on Graphics*, 34(4):1–10, 2015.

[55] J. Bender and D. Koschier. Divergence-free SPH for incompressible and viscous fluids. *IEEE Transactions on Visualization and Computer Graphics*, 23:1193–1206, 2017.

[56] J. Bender and D. Koschier. Divergence-free smoothed particle hydrodynamics. In *14th ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, pages 147–155, Los Angeles, CA, USA, 2015.

[57] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2 edition, 2004.

[58] H. Kato and M. Billinghurst. Marker tracking and HMD calibration for a video-based augmented reality conferencing system. In *2nd IEEE and ACM International Workshop on Augmented Reality*, pages 85–94, San Francisco, CA, USA, 1999.

[59] V. Lippiello, F. Ruggiero, and B. Siciliano. The effects of shapes in input-state linearization for stabilization of nonprehensile planar rolling dynamic manipulation. *Robotics and Automation Letters*, 1(1):492–499, 2016.

[60] D. Serra, F. Ruggiero, A. Donaire, L.R. Buonocore, V. Lippiello, and B. Siciliano. Control of nonprehensile planar rolling manipulation: A passivity based approach. *IEEE Transactions on Robotics*, 2019, DOI: 10.1109/TRO.2018.2887356.

[61] R. M. Murray, Z. Li, and S.S. Sastry. *A mathematical introduction to robotic manipulation*. CRC press, 1994.

[62] A. Gutiérrez-Giles, F. Ruggiero, V. Lippiello, and B. Siciliano. Modelling and control of a robotic hula-hoop system without velocity measurements. In *20th World Congress of the International Federation of Automatic Control*, pages 9808–9814, Toulouse, FR, 2017.

[63] M. W. Spong. Partial feedback linearization of underactuated mechanical systems. In *IEEE/RSJ/GI International Conference on Intelligent Robots and Systems*, pages 314–321, Munich, DE, 1994.

[64] A. Gutiérrez-Giles, F. Ruggiero, V. Lippiello, and B. Siciliano. Nonprehensile manipulation of an underactuated mechanical system with second-order nonholonomic constraints: The robotic hula-hoop. *IEEE Robotics and Automation Letters*, 3(2):1136–1143, 2018.

[65] A. Shiriaev, J.W. Perram, and C. Canudas-de Wit. Constructive tool for orbital stabilization of underactuated nonlinear systems: Virtual constraints approach. *IEEE Transactions on Automatic Control*, 50(8):1164–1176, 2005.

[66] S. Bittanti, A.J. Laub, and J.C. Willems. *The Riccati equation*. Springer-Verlag, New York, NY, 1991.

[67] D. S. Reznik and J. F. Canny. C'mon part, do the local motion! In *2001 IEEE International Conference on Robotics and Automation*, volume 3, pages 2235–2242, Seul, KR, 2001.

[68] T. H. Vose, P. Umbanhowar, and K. M. Lynch. Friction-induced lines of attraction and repulsion for parts sliding on an oscillated plate. *IEEE Transactions on Automation Science and Engineering*, 6(4):685–699, 2009.

[69] T. H. Vose, P. Umbanhowar, and K. M. Lynch. Sliding manipulation of rigid bodies on a controlled 6-dof plate. *The International Journal of Robotics Research*, 31(7):819–838, 2012.

[70] M. Higashimori, K. Utsumi, and M. Kaneko. Dexterous hyper plate inspired by pizza manipulation. In *2008 IEEE International Conference on Robotics and Automation*, pages 399–406, Pasadena, CA, USA, 2008.

[71] M. Higashimori, K. Utsumi, Y. Omoto, and M. Kaneko. Dynamic manipulation inspired by the handling of a pizza peel. *IEEE Transactions on Robotics*, 25(4):829–838, 2009.

[72] A. M. Bloch and P. E. Crouch. Nonholonomic control systems on riemannian manifolds. *SIAM Journal on Control and Optimization*, 33(1):126–148, 1995.

[73] W. M. Boothby. *An introduction to differentiable manifolds and Riemannian geometry*, volume 120. Academic press, 1986.