Proceedings of The 2018 IEEE International
Conference on Real-time Computing and Robotics
August 1-5, 2018, Kandima, Maldives

Manipulation Task Simulation of a Soft Pneumatic Gripper Using ROS and Gazebo

Jun Chen[1,2,3], Hao Deng[1,2,3], Wenjie Chai[1,3], Jing Xiong[1] and Zeyang Xia[1,3,*]

***Abstract*— Bioinspired robotics have shown great advantages for manipulation tasks performing, compared to current rigid structure with limited degrees of freedom. And the research of pneumatic soft robotic grippers has flourished greatly in recent years. However, there is still no effective platform available for robotic manipulation task simulation when equipped with such a deformable gripper. This paper intends to demonstrate a simple but effective approach of how to create simulator of robotic system with soft gripper and implement robot control in a short time. Here we build this platform based on Robot Operation System (ROS) and Gazebo engine. First, we define an unified module description format for gripper configurations. Then, a dynamic model is formulated to control the action of each joints and gripper deformation after simulated pneumatic actuation. Finally, the entire simulation system including modeling, kinematics, control, and visualization is established. Experimental results from pick&place manipulation of several irregular objects have shown that the proposed simulation platform could achieve an easy configuration, good integration and functional visualization for task simulation of robotic system equipped with a soft pneumatic gripper, and provide a flexible technique for quick verification of prototype or algorithms in the research of soft robotics.**

***Index Terms*— Soft robotics, pneumatic gripper, manipulation, simulation.**

I. INTRODUCTION

In the past few decades, robots, based on articulated rigid structures and high stiffness, have been famous for blurring speed, unerring precision and achieved plentiful applications on almost all fields[1]. Moreover, this rigid structural design and prosperity lead to a natural modeling method that all robot can be defined used rigid link bodies and connected joints, for instance the fundamental D-H method [2], which has promoted the entail rigid-body kinematic and dynamic theories in classic robotics. Thus, there are varieties of simulators developed based on this modeling method, like the Microsoft Robotic Developer Studio (MRDS) [3], Open Robotics Automation Virtual Environment (OpenRAVE) [4], and Robot Operation System (ROS) [5]. These platforms have dramatically boosted the research and development of robotic systems.

In the latest ten years, some novel robotic designs have been proposed, which are typically the opposite of traditional philosophy. The robot body may have low stiffness on purpose and could perform large deformation, deviated the robot from genesally prescribed motions. However, those characteristics make the robot more adaptive to environments, and suitable for special applications, such as medical cases, or bioinspired manipulations. The research of this kind of novel robot, also called soft robotics, has flourished greatly in recent years [6] with the aspects from actuating mechanism to driving mode.

Contrary to the classic rigid robots, the soft robots have infinite degrees of freedom and most are underactuated. These make their kinematic and dynamic analysis difficult, which in turn make the development of a simulation system impossible. To our knowledge, there is still no effective platform available for robotic manipulation task simulation when equipped with a soft and deformable robot.

The key point to solve this problem, same as traditional robotics, is how to estimate the mechanical coupling between the actuator parameters and end effectors. Due to hyperplastic nature and non-linear behavior of materials, mainly silicone, of original soft robots, earlier researches have been focused on using the Finite Element Method (FEM) in order to accurately model the deformations that are the cause of this mechanical coupling. A outstanding toolkit may be a Simulation Open Framework Architecture (SOFA) [7, 8]. The analytic method tried to solve this from the constitutive law of the material which is measurable experimentally. Two issues have been found in our research when tried to integrate into a robotic simulator. The first one is it is quite difficult to set the material parameters, which will lead huge errors between the simulated and real results. The second is there would be huge efforts to integrate a rigid manipulator with a soft gripper into one system, because the different modeling method. So this might be an option for robot design and actuating mechanism validation, but impractical for a task simulation. So in our later research, we tried to go back to the beginning. It seems natural and straightforward to use large numbers of links and joints to model the deformation of soft robot, but try to optimize the expression of mechanical coupling between the actuators in each joints. The realistic meaning of this hypothesis is that elements in every soft robots are limited and the precision after simplification are

This work was supported by the National Science Foundation of China (61773365), National Key Project of Research and Development Plan of China Inter-Governmental Cooperative Science and Technology Project (2016YFE0128000), Major Project of Guangdong Province Science and Technology Department (2014B090919002), Shenzhen Research Project (GJHS20160331190459402), SIAT Innovation Program for Excellent Young Researchers (2016Y6G008) and Youth Innovation Promotion Association, Chinese Academy of Sciences (2015301).

[1]Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China

[2]University of Chinese Academy of Sciences

[3]CAS Key Laboratory of Human-Machine Intelligence-Synergy Systems, Chinese Academy of Sciences, Shenzhen, China

[*]Author to whom correspondence should be addressed: Shenzhen Institutes of Advanced Technology, 1068 Xueyuan Avenue, Shenzhen University Town, Shenzhen, China. Email: zy.xia@siat.ac.cn; Phone: +86-755-8639-2218

978-1-5386-6869-6/18/$31.00 © 2018 IEEE

acceptable for task simulation.

In this paper, we take a soft pneumatic gripper as an illustration to demonstrate a simple but effective approach to create simulator of robotic system with rigid manipulator and soft gripper, and also implement robot control of the whole robotic system. Over other approaches for modeling deformations, we propose to use the straightforward mechanism of articulated segmentations of links, and formulate the joint properties. We extract the soft constraint and constraint force mixing of each joints from Gazebo [9] engine. Then we introduce an unified description format for the soft gripper configurations, which allows to model all system hardware in one framework. By setting the joint configuration in a dynamic world, we can continue create the kinematics, control, and visualization modules to establish a manipulation task simulator. Through experiments we show that this simulator could achieve an easy configuration, good integration and visualization for task simulation with series of soft gripper.

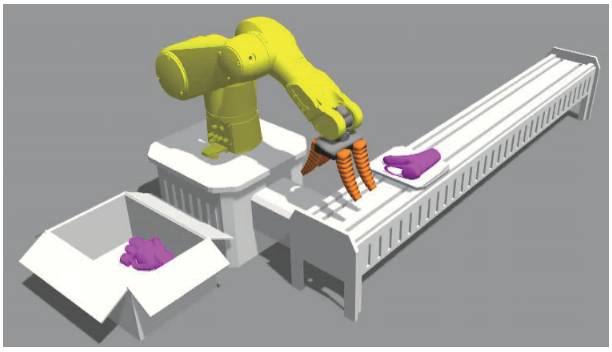
The following of the paper begins by an overview of the robotic system equipped with a soft pneumatic gripper and the manipulation task in the Section II. The proposed modeling method and formulation are presented in Section III. Section IV provides details about how to create a basic simulation platform. A pick&place experiments about the manipulation of several irregular objects are conducted in Section V. And conclusion and the perspectives of this work are discussed in the last Section.

II. OVERVIEW OF THE SYSTEM AND TASK

Grasping involves picking and placing actions and plays as an elemental role in almost all manipulation tasks. And the soft pneumatic grippers can use fewer actuators to drive more or even infinite DOFs, and provide an amazing feature of self-adaptation to uncertainty in grasping, which let grippers easy to control. Thus, we first give a brief overview of a robotic system equipped with this soft gripper, and then describe manipulation tasks of irregular shape object grasping.

Fig. 1a gives a configuration of a robotic grasping system. In which, a 6-DOF manipulator is mounted on a base. And a soft pneumatic 3-DOF fingers gripper is installed as the end of the flange as the end-effector. Each finger is designed based on the soft manipulators outlined in [10, 11]. But in our study, we modify the original design by optimizing the cross-sectional area. The compliance of this gripper allows it to pick up objects, especially irregular shaped ones as illustrated in Fig. 1b, that a traditional rigid gripper is not easily capable of without extensive manipulation planning.

The manipulation task of a soft pneumatic gripper is to pick and place, then box up the irregular objects coming from the horizontal conveyor belt. Additionally, there will be no visual servo for all the tasks, which means the robot will not have full knowledge of the targets to be grasped, but rather only the basic information from task initialization such as the picking and placing poses.



(a)

(b)

Fig. 1. Robotic system configuration and manipulation task. (a) Basic configuration of a robotic grasping system. (b) Irregular objects for grasping. From left to right are plastic models of a dwarf, tap, rabbit, stapler and lion.

III. GAZEBO-BASED MODELING AND FORMULATION

A. Mechanical Modeling

When a soft pneumatic gripper is activated, inner surface of the chamber is pressurized with distributed load and a complex deformation of two neighboring channels is caused. Based on this actuating mechanism, an equivalent mechanical model of a soft pneumatic actuator is built in Fig. 2a. Thus we propose the hypothesis for the theoretical foundations of our simulation framework for deformation object as:

(1) One continuum deformable object can be segmented into finite rigid bodies. And each neighboring bodies are connected to a link by hinge joints.

(2) Each link has one linear DOF in a plane perpendicular to the link axis, and two DOFs in rotation around the axis and has no upper and lower limits.

(3) Deformation and mechanical property of the body can be decomposed into link deformation, joint kindynamic model and their mechanical properties.

(4) Precision of the above simplification is acceptable for task simulation.

Let us start with the structural description that models the behaviors of two connecting bodies. In Fig. 2b, the pose of two neighboring bodies is donated by (p_1, r_1) and (p_2, r_2) at initial time t_0, (p_1', r_1') and (p_2', r_2') at any time t . All the variables are the function of time. Using $l_{t0} = p_2 - p_1$ and $l_t = p_2' - p_1'$, we obtain the linear extension: $\Delta l = |l_t| - |l_{t_0}|$. And from time t_0 to time t , the rotation of rigid body 1 and rigid body 2 can be computed as:

$$
\begin{aligned}
r_{1n} &= r_1' Q_{quaternion}(angle(t)/2))^{-1} r_1^{-1} \\
r_{2n} &= r_2' Q_{quaternion}(angle(t)/2))^{-1} r_2^{-1} \\
r_{1v} &= r_{1n} l_{t_0} \\
r_{2v} &= r_{2n} l_{t_0}
\end{aligned}
$$

$$r_{1d} = Q_d(r_{1v}, l_t)$$
$$r_{2d} = Q_d(r_{2v}, l_t)$$

where $Q_quaternion$ donates the conversion of an angle from Euler to Quaternion representation, and Q_d donates the angle between two vectors.

After establishing the above values, we can computer the deformational force based on the Hooke law.

The force from linear extension Δl and applied on rigid body 1 and rigid body 2 can be formulated as:

$$F_{l_1} = \widetilde{l_t}\Delta l K_s = -F_{l_2}$$

The moment from rotation r_{1d} and applied on rigid body 1 and rigid body 2 can be formulated as:

$$(v_{r_1d}, e_{r_1d}) = Q_{axis-angle}(r_{1d})$$
$$T_{r_1d} = v_{r_1d} e_{r_1d} K_{ts}$$

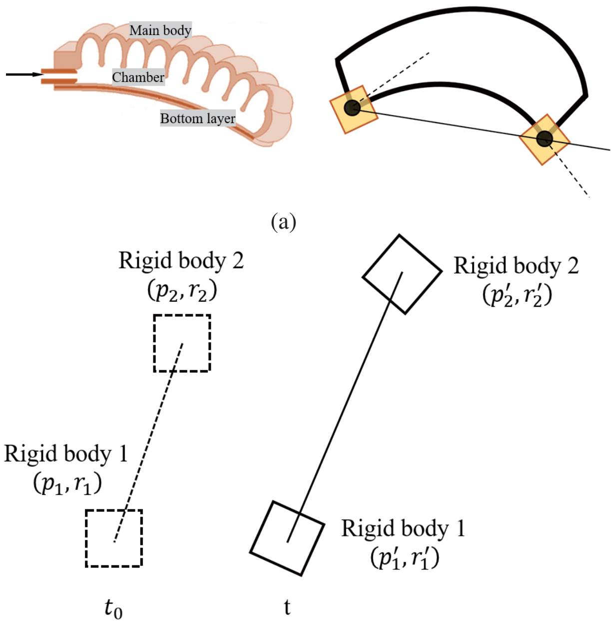
where $Q_{axis-angle}$ returns the axis and angle of a rotation in Quaternion representation.

Since the moment from rotation is not at rotation center, they would generate force and applied on rigid body 1 and rigid body 2, which can be formulated as:

$$F_{e_{r_1d_1}} = -e_{r_1d}\,|l_t|^{-1}\,(\widetilde{l_t} \times v_{r_1d})K_{rs} = -F_{e_{r_1d_2}}$$

And we can also formulate the moment and generated force from rotation r_{2d} as:

$$(v_{r_2d}, e_{r_2d}) = Q_{axis-angle}(r_{2d})$$
$$T_{r_2d} = v_{r_2d} e_{r_2d} K_{ts}$$
$$F_{e_{r_2d_1}} = -e_{r_2d}\,|l_t|^{-1}\,(\widetilde{l_t} \times v_{r_2d})K_{rs} = -F_{e_{r_2d_2}}$$



(b)

Fig. 2. Mechanical modeling of the deformable object. (a) Equivalent mechanical model of a soft pneumatic actuator. (b) Description that models the behaviors of two connecting bodies.

Therefore, the force and moment applied on rigid body 1 and rigid body 2 can finally be formulated as:

$$F_1 = F_{l_1} + F_{e_{r_1d_1}} + F_{e_{r_2d_1}}$$
$$T_1 = T_{r_1d}$$
$$F_2 = F_{l_2} + F_{e_{r_1d_2}} + F_{e_{r_2d_2}}$$
$$T_2 = T_{r_2d}$$

B. Deformation Formulation and Computation

As presented above, the proposed method relies on the computation of deformational force and moment. Normally a numerical computation method is based on discrete iteration, which means the desired force will be calculated in a certain frequency, and in the period between two cycles, the system is treated as constant system. We can prove that the error of this iteration is $O(h^2)$, where h is the iteration step. In a mechanical system, the constant applied forces will generate acceleration, and will finally lead to misconvergence. This dynamic mechanism also happened in Gazebo, a physical simulation engine.

In our research, we modified some lower interfaces and configurations of the Open Dynamic Engine (ODE) [12] in Gazebo. The main concept of our ideas is to take over the control of pose updating, instead of default policy, we used Runge-Kutta iteration to calculate the poses of each rigid body.

Since the force and moment applied on rigid bodies are formulated, if letting

$$x(t) = (p(t), r(t))$$
$$\dot{x}(t) = (\dot{p}(t), \dot{r}(t))$$
$$\ddot{x}(t) = (\ddot{p}(t), \ddot{r}(t)) = f(x(t))$$

we can obtain

$$K_1 = \dot{x}(t_0)$$
$$L_1 = \ddot{x}(x(t_0))$$
$$K_2 = \dot{x}(t_0) + 0.5 \cdot \Delta t \cdot L_1$$
$$L_2 = \ddot{x}(x(t_0) + 0.5 \cdot \Delta t \cdot K_1)$$
$$K_3 = \dot{x}(t_0) + 0.5 \cdot \Delta t \cdot L_2$$
$$L_3 = \ddot{x}(x(t_0) + 0.5 \cdot \Delta t \cdot K_2)$$
$$K_4 = \dot{x}(t_0) + \Delta t \cdot L_3$$
$$L_4 = \ddot{x}(x(t_0) + \Delta t \cdot K_3)$$

hence we have

$$x(t + \Delta t) = x(t_0) + (K_1 + K_2 + K_3 + K_4) \cdot \Delta t/6$$
$$\dot{x}(t + \Delta t) = \dot{x}(t_0) + (L_1 + L_2 + L_3 + L_4) \cdot \Delta t/6$$

We assume the policy of pose updating holds, integrating over Δt, we can conclude when the period is enough short, the updated pose will be given.

C. Realization in Gazebo

We implement the formulation in Gazebo ODE engine as a plugin, which is responsible for deformable objects and works as a patch to the default model manager. Pseudocode for the plugin is given in Algorithm 1.

When the plugin is initialized, it will find all joints labeled with soft in the description file, URDF, and put in an array. After wrapping each pair of connecting mechanism with their properties, plugin will take over the control of joint behaviors. In each updating cycle, Runge-Kutta iteration is

Algorithm 1 ModelPlugin($model$)

procedure INIT($model$)
  Find all $joint$ with $Flag$, and put in $Joints$
  for each $joint$ in $Joints$ **do**
    Get $links$ of $joint$, and put in $body1$ and $body2$
    Wrapping $body1$ and $body2$
    Detach Gazebo joint control
    Create Soft control, and put in $soft\_joints$
  end for
end procedure

procedure ONUPDATE($info$)
  CalDeltaTime($info$)
  for each $soft\_joint$ in $soft\_joints$ **do**
    CalAngle(cur_angle, tar_angle, $delta\_t$)
  end for
end procedure

procedure RIGIDDELEGATEDUPDATERK
  GetBodyInfo() and put in cur_pose
  CalJointForce(cur_pose)
  Calculate $K1$, $L1$, $K2$, $L2$, $K3$, $L3$, $K4$, $L4$
  Calculate $joint\_pose$, $force$ and $velocity$
  Set Body velocity with joint damp
end procedure

called to calculate the poses, force and joint velocity of each rigid bodies.

IV. ROS-BASED SIMULATOR IMPLEMENT

As all classical robotic manipulation task simulator, ROS works as a fundamental layer of architecture support for algorithm realization. In standard ROS packages, a simulator called Rviz could provide basic visualization functions, but lack of dynamic functional modular. In our simulator, Gazebo and Rviz are integrated and configured to display information by taking advantages of each other.

Compared to previous work for rigid robot simulation [13, 14], a four-layer structure is built for this manipulation task including the task manager layer, planner layer, controller layer and visualizer layer. And most of work in this paper has been done in the controller layer and visualizer layer, thus we will depict the details for manipulation simulation of a soft gripper in this section.

A. Control of a Soft Pneumatic Gripper

Normally rigid multi-DOF grippers are in a joint position or force control mode, which means actuators are commanded by equal number of driving variables. For each finger of a soft pneumatic gripper, it is the one-DOF of compressed air that enable the actions, in a basic case, featuring by pressurized or not command and the pressure. There are a lot of researches having been done to build the mapping from air pressure to body deformation, mostly of which are based on the FEM In our study, we use a large number of links and joints to model the deformation of gripper.

If we can build the relationship between air pressure and desired angle of end effector, then we will be more concretely to see how this translates in the control modular of soft gripper. As shown in Fig. 3, from FEM we can obtain a series of data about deformation angle of end effector under the driving of different air pressure. By using a modular of least square fitting method from TensorFlow [15], we could obtain predicted relationships between air pressure and desired angle of end effector. We have presented theoretical foundation of gripper deformation and computation method. Thus, we could finally build entire control information flow from a given air pressure to joint commands of each soft joints. And a soft pneumatic controller can be realized for each finger of the gripper.

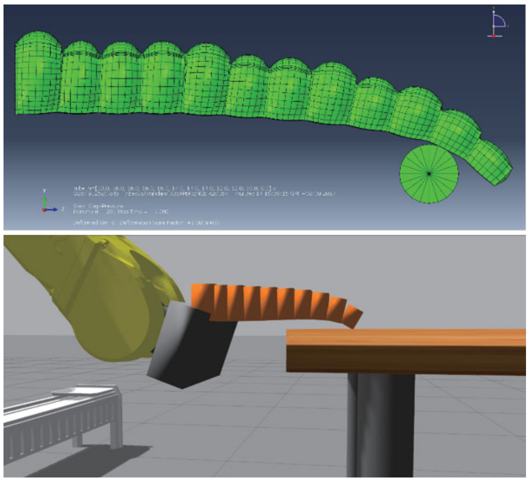


Fig. 3. Relationship between air pressure and deformation angle.

Fig. 4. Grasping actions from opening to closing simulated in the developed simulator.

After establishing the soft pneumatic controller, we bring up the whole system including a 6-DOF manipulator and the 3-DOF gripper. By controlling movement of each finger, we can achieve a series of grasping actions as showed in Fig. 4.

B. Pneumatic Force Computation and Visualization

From modeling and formulation of the inner mechanism, we have built the control equation of the deformation. And at the same time, we also got the inner force and displacement of the gripper. As shown in Fig. 5, in Rviz, we used markers to display the color bars and modified the visual property to show inner force on real time, which could also indicate states of element deformation.

V. MANIPULATION EXPERIMENTS AND RESULTS

We validated the simulator by performing manipulation task of grasping irregular objects. In preliminary experiment,

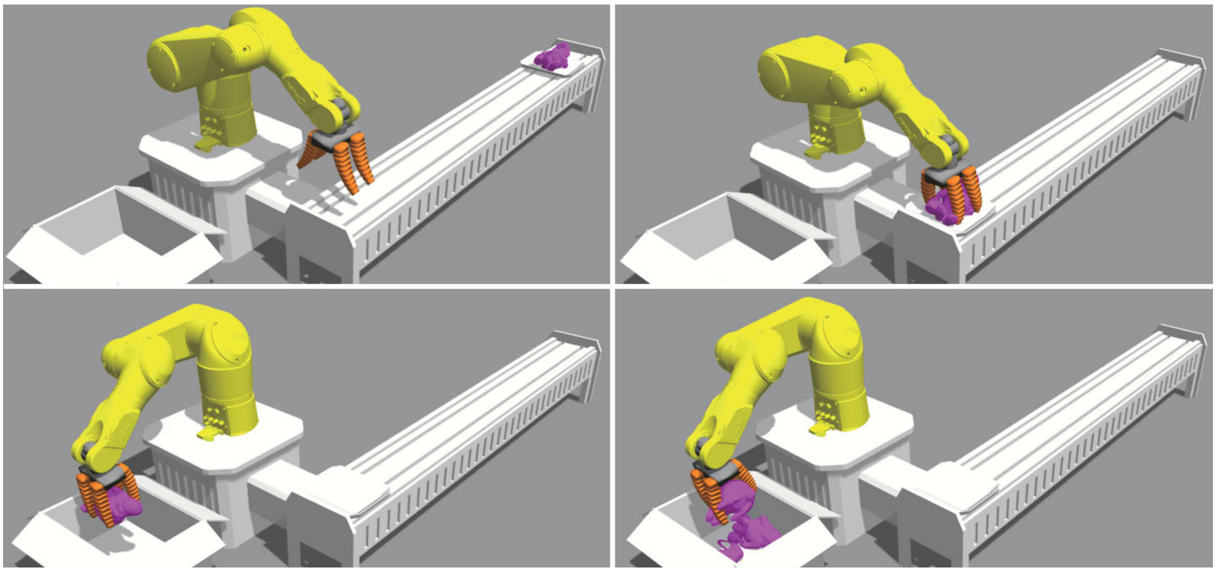


Fig. 6. Manipulation task simulation of a manipulator equipped with a soft pneumatic gripper.

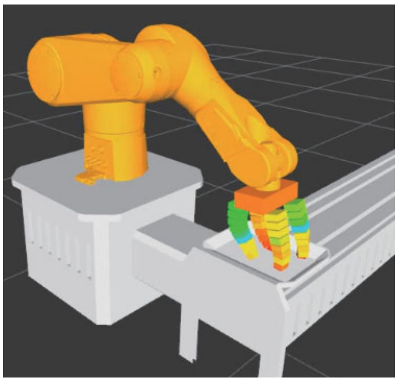


Fig. 5. Inner force and displacement display in Rviz using color bar.

the irregular objects are plastic models of a dwarf, tap, rabbit, stapler and lion. Fig. 6 gives the snapshots of the manipulation process. By setting a few trajectory waypoints which dividing the whole process into several separate parts. In which the soft pneumatic gripper picks up the irregular objects coming from horizontal conveyor belt in order, then places into a box.

Fig. 7. Simulation results of grasping a rabbit.

In Fig. 7, taking the grasping of a rabbit for example, we can see that soft pneumatic gripper can automatically adapt to different shape of objects, without visual serving. Which is a tough task for any traditional rigid gripper, from the aspect of robotic knowledge. The result of the grasping of all objects are given in Fig. 8. Experimental results have shown that the developed simulation system can provide a basic solution to kindynamic analysis of soft pneumatic gripper, and coordinate with current robotic simulation system. And video of this experiments can be download at: https://github.com/manipulation_simulator.

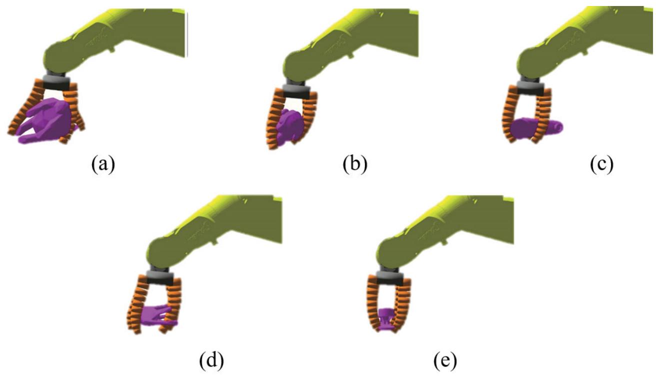


Fig. 8. Automatically adapting to different shape of objects, which are the a) rabbit, b) dwarf, c) tap, d) lion and e) stapler.

VI. CONCLUSION AND DISCUSSION

In this paper, an effective approach to creating simulator for a robotic system equipped with soft gripper was presented. The advantage of our method is that the entire platform was developed based on the existing ROS and Gazebo framework. In order to handle the deformable object, first, we define a unified description format for the soft gripper configurations. Then, a dynamic model is formulated to control the action of each joints and gripper deformation after simulated pneumatic actuation. Combining controllers of rigid manipulator and the proposed soft pneumatic object, the developed platform could coordinate with current robotic simulation system. Results from pick&place manipulation of several irregular objects have shown that the proposed simulation platform could achieve an easy configuration, good integration and functional visualization for task simulation of robotic system equipped with a soft pneumatic gripper, and provide a flexible technique for quick verification of prototype or algorithms in the research of soft robotics.

Our future interests include (1) optimizing the unified description format for more soft robots, (2) applying on physical system and evaluating the precision, (3) improving the dynamic model for high accuracy of deformation prediction.

REFERENCES

[1] LiveScience, Available at: https://www.livescience.com/topics/robots.

[2] Denhavit J. A kinematic notation for lower-pair mechanisms based on matrices[J]. ASME J. Appl. Mech., 1955.

[3] Jackson J. Microsoft robotics studio: A technical introduction[J]. IEEE Robotics & Automation Magazine, 2007, 14(4).

[4] Diankov R, Kuffner J. OpenRAVE: A planning architecture for autonomous robotics[J]. Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-08-34, 2008, 79.

[5] Quigley M, Conley K, Gerkey B, et al. ROS: an open-source Robot Operating System[C]//ICRA workshop on open source software. 2009, 3(3.2): 5.

[6] Rus D, Tolley M T. Design, fabrication and control of soft robots[J]. Nature, 2015, 521(7553): 467.

[7] Allard J, Cotin S, Faure F, et al. Sofa-an open source framework for medical simulation[C]//MMVR 15-Medicine Meets Virtual Reality. IOP Press, 2007, 125: 13-18.

[8] https://www.sofa-framework.org/

[9] http://gazebosim.org/

[10] Marchese A D, Katzschmann R K, Rus D. A recipe for soft fluidic elastomer robots[J]. Soft Robotics, 2015, 2(1): 7-25.

[11] Katzschmann R K, Marchese A D, Rus D. Autonomous object manipulation using a soft planar grasping manipulator[J]. Soft robotics, 2015, 2(4): 155-164.

[12] http://www.ode.org/

[13] Qian W, Xia Z, Xiong J, et al. Manipulation task simulation using ros and gazebo[C]//Robotics and Biomimetics (ROBIO), 2014 IEEE International Conference on. IEEE, 2014: 2594-2598.

[14] Deng H, Xiong J, and Xia Z. Mobile Manipulation Task Simulation using ROS with MoveIt [C]// Real-time Computing and Robotics (RCAR), 2017 IEEE International Conference on. IEEE, 2017. (to be appeared)

[15] Abadi M, Barham P, Chen J, et al. TensorFlow: A System for Large-Scale Machine Learning[C]//OSDI. 2016, 16: 265-283.