Simulation of the underactuated Sake Robotics Gripper in V-REP

Simon-Konstantin Thiem[1], Svenja Stark[1], Daniel Tanneberg[1], Jan Peters[1,2] and Elmar Rueckert[1]

***Abstract*— The Sake Robotics Gripper is a cheap, robust and versatile underactuated gripper that has not been simulated yet. The simulated model has to be able to interpret the same ROS messages the real gripper receives. This paper proposes a reproduction of the Sake Robotics Gripper in V-REP. We analyze the tools provided by V-REP to develop an algorithm for simulating the underactuation of the real gripper. Our model can be used as a foundation for research in complex grasping and manipulation tasks with the Sake Robotics Gripper.**

I. INTRODUCTION

In the last years there has been a lot of progress in the creation of domestic robots. Even though they are able to perform certain tasks such as cleaning floors or mowing lawn, they still struggle with more complex manipulation and grasping tasks [1]. Various grippers are built based on the example of the human hand [2] [3]. For machine learning methods that try to learn such complex manipulation tasks this results in complicated models due to the hand's complexity, limiting the learning results. [4].

To achieve versatile grasping capabilities, a simple gripper is easier to control and more reliable for learning. We use the SAKE Robotics gripper (SRG) [5]. The gripper is attached to a Kuka LBR iiwa 14 R820 [6]. Compared to other state of the art grippers, the SRG is relatively cheap. This two-finger gripper is versatile since it encompasses objects and thus can grip thin as well as bulky items. Additionally, it does not have fragile finger tips that can break. It is an underactuated gripper and has only a simple one-dimensional input. This enables the application of control policies with only few parameters.

In this work we provide an open-source model of the SRG for the virtual robot experimentation platform V-REP (Coppelia Robotics GmbH, Zürich Switzerland) [7]. The model enables realistic simulations with physical contacts. Our simulation implements the same interface as the real gripper, enabling easy transfer from simulation to the real system. Even though the control signal of the actuator is one-dimensional, the gripper has four degrees of freedom. We propose an algorithm that mimics the behavior of this underactuation in V-REP. Additionally, we show an approach to integrate touch sensors in the V-REP simulation of the gripper.

The underactuation is implemented in a child script in V-REP. Therefore, we can maintain the advantage of having a low-dimensional input and ,thus, the only information that needs to be learned is how far the hand has to close. We compensate the lack of touch sensors with proximity sensors.

[1]Intelligent Autonomous Systems Lab, Technical University Darmstadt, Germany. {thiem, rueckert}@ias.tu-darmstadt.de

[2]Robot Learning Group, Max-Planck Institute for Intelligent Systems, Tuebingen, Germany mail@jan-peters.net

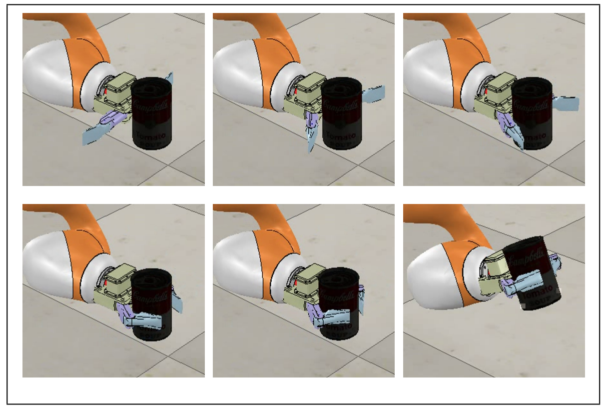

Fig. 1. Sequence of the simulated gripper grasping the tomato soup can from the ycb object model set [8] [9].

A. Related Work

The design of robotic grippers is an ongoing research field. Asfour et al. present a domestic robot for research purposes that uses a five finger gripper. They propose to do offline grasp analysis for each object that is supposed to be grasped [10]. This results in a huge database. By using online learning the system is able to manipulate unknown objects.

Prior to the study of Asfour et al., grippers have mainly been used and designed for industrial purposes. In most cases they were constructed for one specific task only. Lanni and Ceccarelli give a broad overview of two-finger grippers and propose an objective function for the constrained optimization problem to develop grippers for specific tasks [11]. There is also research on gripping materials which are difficult to clench. Drigalski et al. introduce a 3D printable gripper that is able to hold textiles [12], whereas Menciassi et al. focus on grasping objects of size of 1 mm and smaller [13].

In the late 1970s, Hirose and Umetani proposed a soft gripper which has an adjustable amount of finger links where each link results in a degree of freedom. Equally to the SRG, an underactuated pull on a wire causes the two fingers/chains to move inwards. As the chains can have arbitrary lengths, objects can be completely enveloped when grasped [14].

Being able to simulate a task ahead of its physical implementation, enables fast prototyping of control strategies and machine learning approaches. For creating the SGR simulation, we used the V-REP simulation software. It has a kinematic, a dynamics and a path planning module. While the implementation of the kinematic and the path planing module execute movement exactly as programmed, they are not able to correctly detect objects and forces. The dynamics module

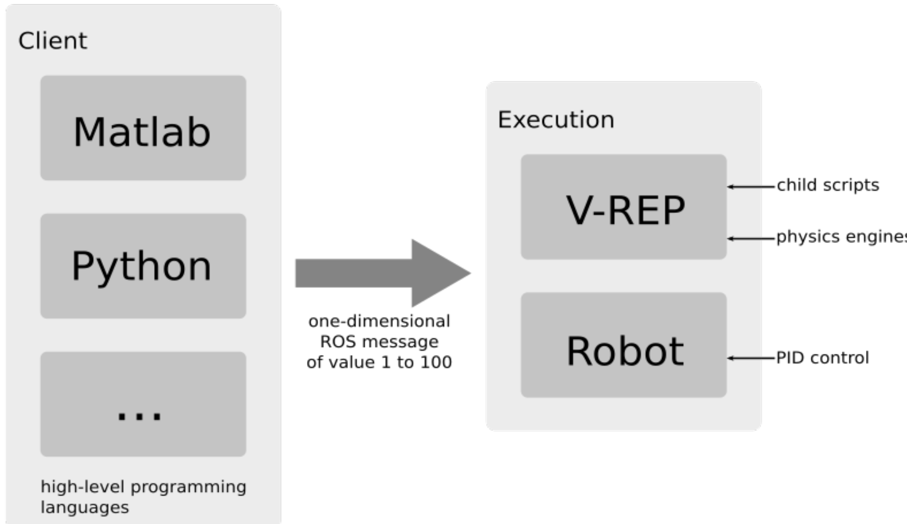

Fig. 2. System overview sketch. The client sends the same commands to the robot or to the V-REP simulation using ROS, making the high-level program invariant to where the execution actually takes place.

on the other hand, makes use of one of the included physics engines. Ivaldi, Peters, Padois and Nori did an analysis of the tools that are capable of dynamics simulations. While they point out the diversity of the tools, most of them rely on physics engines like Bullet or ODE that were originally designed for video games [15]. While games try to achieve a smooth user experience, their prior concern is on speed and less on accuracy. In simulated robotics however, the main goal is representing the reality as close as possible. This often results in a problem as modern physics engines can still be insufficient for manipulation tasks with physical contacts.
We chose to use V-REP as development environment because it is a user friendly, intuitive program, that has many features implemented, while being competitive with other state of the art robotics simulation tools like Gazebo [16].

II. Simulating an Underactuated Gripper

The SRG's size is appropriate for gripping objects like bottles, packages, and other everyday household items that are used in the kitchen. The gripper has a grasp width of $145\,\mathrm{mm}$ and can hold up to $5\,\mathrm{kg}$ when it can grasp while wrapping around an object or $2.5\,\mathrm{kg}$ when holding it with finger tips.

A. Data-flow

A setup for learning how to grasp an object can be separated into three parts (Figure 2). These parts are a learning algorithm, the Robot Operating System (ROS) and lastly the hardware or the simulation.
The desired learning algorithm can be implemented in any high-level programming language with a ROS interface. It has to send ROS messages containing information about the desired motion of the robot. Another series of messages delivers information for the motor which is in charge of moving the fingers of the SRG. These messages consist of a value between 1 and 100 that determines how far the SRG closes. The SRG is underactuated due to the signal for the grasp being only one-dimensional even though the gripper has 4 degrees of freedom. This will be discussed in Section II-C. ROS is in charge of communicating this information between the high-level program and V-REP or the robot.
For the execution we use the Kuka LBR iiwa 14 R820 [6] with the SRG [5] as hardware and alternatively we interpret these messages in the simulation software V-REP by Coppelia Robotics [7]. In V-REP, a child script, written in Lua, can be attached to each simulated object. These control their actions and can be used to receive and interpret ROS messages. We wrote them in a way, that we are able to use the exact same ROS messages for the robot and the SRG. This is of use because it allows the use of the same interface for simulation and hardware respectively. Hence it is possible to generate learning data by simulations which then can be used for hardware applications.

B. Problems of Current Physics Engines

Several physics engines (Bullet 2.78, Bullet 2.83, ODE, Vortex, Newton) can be chosen in V-REP to simulate the movement of the SRG. We used the Open Dynamics Engine (ODE) [17] since it performs best for our task. There are three issues that determine the behavior in the simulation which are outlined below.

1) Non-Convexity of Mesh Files: The mesh files of the SRG were taken from the official Github page of SakeRobotics [18]. Due to the SRG being a non-convex shape, the mesh files for it are non-convex as well, which is the reason for a low cohesion between separate parts of the gripper. This results in the possibility of joints being unable to apply force towards the right directions, items being pushed away instead of being grasped and fingers of the gripper jumping back

Algorithm 1: Simulated underactuation for one finger for a single time step.

```
Input: posOfBase, posOfTip, proxSensorBase,
  proxSensorTip, newPos, currentPos
opening = false
if currentPos < newPos then
    opening == true
else if newPos < currentPos then
    opening == false
if opening then
    Once tip is completly opened:
    if posOfTip < 0 then
        open base:
        posOfBase ← newPos
        keep tip at 0:
        posOfTip ← 0
    else open tip
        posOfTip ← (newPos − posOfBase)
closing gripper while the base of the finger touches an
  object:
else if proxSensorBase then
    posOfTip ← (newPos − posOfBase)
    if proxSensorTip then
        applying pressure:
        posOfBase ← 0.5(newPos − posOfBase −
          posOfTip) + posOfBase
        posOfTip ← 0.5(newPos − posOfBase −
          posOfTip) + posOfTip
else
    posOfBase ← newPos
```

and forth. We address the first two problems in Section II-C. Fortunately, the ODE physics engine is able to avoid fluctuation between non-convex shapes reasonably well.

2) Detecting Contacts: V-REP does not have sensors implemented that recognize touch or contacts. A potential solution is the use of a built-in force sensor in V-REP, which however requires a rigid link between the SRG and the grasped object, inhibiting slip completely. Alternatively proximity sensors can be used. These sensors are able to detect if an object is close to the fingers. The distance threshold at which they should signal an object contact can be tuned. However these values need to be greater than zero and therefore, the sensors fire slightly before an actual touch happens. This issue is discussed further in Section II-D.

3) Handling of Soft Items: Current physics engines are not able to handle soft items. These would require a different behavior [19] [20]. Thus we are not able to simulate the grasp of items that can be squeezed.

C. Control of the Real Grippers

The hardware link between the SRG and the robot has only a one-dimensional actuator. By default torsion springs

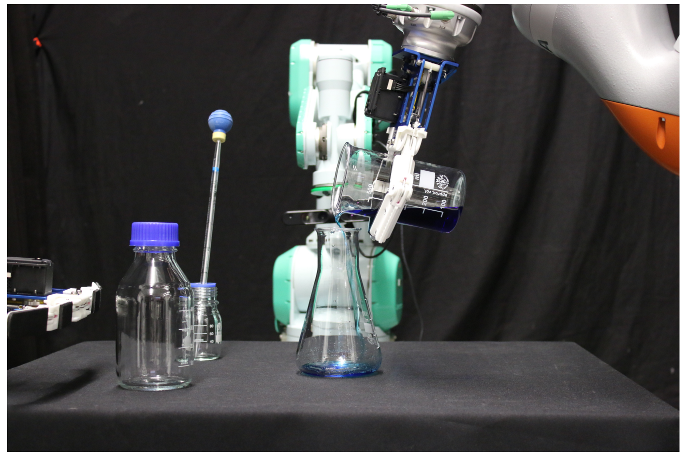

Fig. 3. Example application of the SRG holding a glass to pour a liquid.

hold the fingers in a 180 degrees straight position. To close the fingers the robot pulls on a wire exerting an opposing force against the springs, resulting in a contraction of the gripper. This command can be sent via a ROS message with a value between 0 and 100. The value determines how far the wire will be pulled. First the finger bases start moving until they hit a resistance. The wire is also connected to the finger tips. Thus, when the force cannot be emitted onto the base of the finger anymore, it is transfered to an area with lower resistance, the finger tip. Once the tips of the finger touch each other or the object, force is applied towards the object. This way a firm grip surrounding the grasped object can be established as demonstrated in Figure 3.

The SRG uses a PID controller to administer the pulse width modulation that controls the motor. This is denoted by τ and calculated by a feed forward term $u_{FF}(t)$ and a feedback term $u_{FB}(t)$ of the torque u at the time t. The κ is the pulse width modulation limit. The desired reference trajectory $q^*(t) \in [0, 100]$ is assumed to be given, e.g., it might be obtained from a minimum jerk reference trajectory generator [21] [22], stochastic optimal control [23] [24] or through kinesthetic teaching [25]. The current position is denoted by $q(t)$, velocity by $\dot{q}(t)$ and acceleration by $\ddot{q}(t)$.

$$\tau(t) = \lfloor u_{FF}(t) + u_{FB}(t) \rfloor_\kappa \tag{1}$$

$$u_{FF}(t) = K_{FF_1}\dot{q}(t) + K_{FF_2}\ddot{q}(t) \tag{2}$$

$$u_{FB}(t) = K_P(q^*(t) - q(t)) + + K_I(q^*(t) - q(t))\dot{q}(t) + \frac{K_D(q^*(t) - q(t))}{\dot{q}(t)} \tag{3}$$

Substituting Equation (2) and (3) into Equation (1) results in the final control signal $\tau(t)$ given by

$$\tau(t) = \lfloor K_{FF_1}\dot{q}(t) + K_{FF_2}\ddot{q}(t) + + K_P(q^*(t) - q(t)) + K_I(q^*(t) - q(t))\dot{q}(t) + + \frac{K_D(q^*(t) - q(t))}{\dot{q}(t)} \rfloor_\kappa \quad . \tag{4}$$

D. Modeling

We aim to simulate the underactuated closing behavior in V-REP (after the ROS messages have been received). The messages contain a value between 0 and 100 which is mapped to all possible radiants of the finger, where 0 corresponds to the open position and 100 to the closed one. Our Algorithm 1 closes the fingers until either the desired value is reached or one of the base proximity sensors fire. Such firing happens slightly before the fingers touch an object. Thus, no force is yet applied towards the item and therefore it can not be moved or lifted yet. Hereby we vastly reduce the possibility that the item is pushed away by the fingers. Once the proximity sensor of the finger tip fires, the finger bases start grasping again. This procedure happens when the fingers either reach the object that has to be grasped or each other. As a result, the fingers apply force to the object and thereby establish a secure grip. An example course of this sequence is pictured in Figure 1.

III. Conclusion and Future Work

The Sake Robotics Gripper (SRG) is a powerful gripper that can grasp numerous types of objects and loads up to 5 kg. Due to its underactuation, the SRG has the advantage of having a low dimensional grasp controller compared to other state of the art grippers. However, no simulation was provided. We contribute an open-source V-REP model of the SRG. We use ROS to send messages between a high level programming language like Matlab or Python to V-REP and interpret the motor command there to let the model perform grasps. For this purpose, we present an algorithm approach to mimic the underactuation of the SRG while it also handles the problem of not having any touch sensors in V-REP.[1]

Being able to simulate the SRG provides us with the possibility to apply reinforcement learning to grasps in simulation. This however requires a reward function of how well a grasp has been performed. A number of such reward functions have been proposed [26] [27] [28] [29]. In future work we want to evaluate different reward functions for grasping using our SRG model.

Acknowledgment

This project has received funding from the European Unions Horizon 2020 research and innovation program under grant agreement No 713010 (GOAL-Robots) and No 640554 (SKILLS4ROBOTS).

References

[1] H. Van Hoof, T. Hermans, G. Neumann, and J. Peters, "Learning robot in-hand manipulation with tactile features," in *Humanoid Robots (Humanoids), 2015 IEEE-RAS 15th International Conference on*, IEEE, 2015.

[2] H. Liu, K. Wu, P. Meusel, N. Seitz, G. Hirzinger, M. Jin, Y. Liu, S. Fan, T. Lan, and Z. Chen, "Multisensory five-finger dexterous hand: The dlr/hit hand ii," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, IEEE, 2008.

[3] S. Ekvall and D. Kragic, "Interactive grasp learning based on human demonstration," in *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, vol. 4, IEEE, 2004.

[4] T. Osa, J. Peters, and G. Neumann, "Experiments with hierarchical reinforcement learning of multiple grasping policies," in *International Symposium on Experimental Robotics*, Springer, 2016.

[5] "Sake robotics." Accessed: 2017-09-18.

[6] "Kuka lbr iiwa 14 r820." Accessed: 2017-09-19.

[7] "Coppelia robotics." Accessed: 2017-09-19.

[8] "Ycb benchmarks." Accessed: 2017-09-19.

[9] B. Calli, A. Singh, A. Walsman, S. Srinivasa, P. Abbeel, and A. M. Dollar, "The ycb object and model set: Towards common benchmarks for manipulation research," in *Advanced Robotics (ICAR), 2015 International Conference on*, IEEE, 2015.

[10] T. Asfour, P. Azad, N. Vahrenkamp, K. Regenstein, A. Bierbaum, K. Welke, J. Schroeder, and R. Dillmann, "Toward humanoid manipulation in human-centred environments," *Robotics and Autonomous Systems*, vol. 56, no. 1, 2008.

[11] C. Lanni and M. Ceccarelli, "An optimization problem algorithm for kinematic design of mechanisms for two-finger grippers," *Open Mechanical Engineering Journal*, vol. 3, 2009.

[12] F. von Drigalski, D. Yoshioka, W. Yamazaki, S.-G. Cho, M. Gall, P. M. U. Eljuri, V. Hoerig, J. Beltran, M. Ding, J. Takamatsu, *et al.*, "A versatile, open-source two-finger gripper for textile manipulation,"

[13] A. Menciassi, A. Eisinberg, I. Izzo, and P. Dario, "From" macro" to" micro" manipulation: models and experiments," *IEEE/ASME Transactions on mechatronics*, vol. 9, no. 2, 2004.

[14] S. Hirose and Y. Umetani, "The development of soft gripper for the versatile robot hand," *Mechanism and machine theory*, vol. 13, no. 3, 1978.

[15] S. Ivaldi, J. Peters, V. Padois, and F. Nori, "Tools for simulating humanoid robot dynamics: a survey based on user feedback," in *Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on*, IEEE, 2014.

[16] L. Nogueira, "Comparative analysis between gazebo and v-rep robotic simulators," *Seminario Interno de Cognicao Artificial-SICA 2014*, 2014.

[17] "Open dynamics engine." Accessed: 2017-09-19.

[18] "Github repository of sakerobotics containing the second generation of the ezgripper." Accessed: 2017-09-19.

[19] D. J. Montana, "The kinematics of contact with compliance," in *Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on*, IEEE, 1989.

[20] D. J. Montana, "Contact stability for two-fingered grasps," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 4, 1992.

[21] K. J. Kyriakopoulos and G. N. Saridis, "Minimum jerk path generation," in *Proceedings. 1988 IEEE International Conference on Robotics and Automation*, Apr 1988.

[22] T. Flash and N. Hogan, "The coordination of arm movements: an experimentally confirmed mathematical model," *Journal of neuroscience*, vol. 5, no. 7, 1985.

[23] H. J. Kappen, V. Gómez, and M. Opper, "Optimal control as a graphical model inference problem," *Machine learning*, vol. 87, no. 2, 2012.

[24] E. Rueckert, G. Neumann, M. Toussaint, and W. Maass, "Learned graphical models for probabilistic planning provide a new class of movement primitives," *Frontiers in Computational Neuroscience*, vol. 6, no. 97, 2013.

[25] A. Paraschos, C. Daniel, J. Peters, and G. Neumann, "Using probabilistic movement primitives in robotics," accepted.

[26] C. Ferrari and J. Canny, "Planning optimal grasps," in *Robotics and Automation, 1992. Proceedings., 1992 IEEE International Conference on*, IEEE, 1992.

[27] B. D. Ziebart, A. L. Maas, J. A. Bagnell, and A. K. Dey, "Maximum entropy inverse reinforcement learning.," in *AAAI*, vol. 8, Chicago, IL, USA, 2008.

[28] T. Baier-Lowenstein and J. Zhang, "Learning to grasp everyday objects using reinforcement-learning with automatic value cut-off," in *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, IEEE, 2007.

[29] T. Osa, A. M. G. Esfahani, R. Stolkin, R. Lioutikov, J. Peters, and G. Neumann, "Guiding trajectory optimization by demonstrated distributions," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, 2017.

[1]The V-REP implementation of the the gripper can be found at *https://git.ias.informatik.tu-darmstadt.de/thiem/EZGripper_vrep_model*