Robust Grasping under Uncertainty Employing Tactile Sensors

Luca Lach[1,2], Robert Haschke[2], Francesco Ferro[1], Helge Ritter[2]

***Abstract*— We present a holistic grasping controller, combining free-space position control and in-contact force-control for reliable grasping given uncertain object pose estimates. Employing tactile fingertip sensors, undesired object displacement during grasping is minimized or completely avoided, by stopping the finger closing motion for individual joints as long as force-closure cannot yet be guaranteed. Experiments conducted on the parallel-jaw gripper of the PAL TIAGo robot confirm the efficiency of the proposed approach.**

I. Introduction

In recent years, service robotics has increased in popularity in both, commerce and research. The most important skills to master for robots that are employed in a household setting are object grasping and manipulation. Unfortunately, today's robot performance in this domain is mostly underwhelming due to a large variety of environments, resulting in a high degree of perceptual uncertainty, as well as the lack of (force) feedback during physical interaction with objects.

As a majority of available service robots only have non-sensorized end-effectors, they rely on open-loop grasping approaches [1]. Due to high uncertainty in object pose estimation, these methods often have undesired side effects during manipulation. Without tactile sensors, they are unable to detect object contact and cannot adapt their pre-determined end-effector trajectory accordingly. Thus, they risk tipping over the to-be-grasped object or displacing it in some other unintended fashion, sometimes rendering a stable grasp infeasible. Having established a stable grasp with an object, it is important to control contact forces to avoid damage to the object. By equipping their end-effectors with tactile sensors, robots can synchronize their finger-closing motions, flexibly adapt to sensed object contacts, and control contact forces exerted onto the object. This allows the robot to detect and react to erroneous object pose estimates and to gently grasp also fragile objects.

Another challenge in this field of research is posed by the high diversity of both, robot end-effectors and tactile sensors. Kappassov *et al.* [2] reviewed over 28 different fingertip sensors for dexterous hands alone. Other types of end-effectors are usually combined with different sensor types, increasing the number of end-effector sensor combinations even more [3]. Most of the research on grasp force control was therefore targeting specific combinations of an end-effector and its tactile sensors, e.g. the Shadow Robot Hand grasping unknown objects in simulation [4] or

This work was supported by the European Union Horizon 2020 Marie Curie Actions under Grant 813713 NeuTouch.

[1]PAL Robotics, Barcelona, Spain
[2]Neuroinformatics Group, Bielefeld University, Bielefeld, Germany

Fig. 1: Grasping an object with TIAGo's sensorized parallel-jaw gripper.

with MID fingertip sensors [5], the PR2 with a pressure sensor array mounted on a parallel-jaw gripper [6], or a three-fingered robot hand in simulation [7]. Providing an end-effector-agnostic solution would greatly decrease the individual development effort for new robot platforms that employ tactile sensors on their end-effectors.

The main contribution of our work is a holistic grasp controller that makes as little assumptions about the robot platform as possible. It is well-suited for different types of end-effectors and tactile sensors and its core library is middleware agnostic. The controller aims at an increased robustness when establishing object contact during the finger-closing phase, compensating for perceptual uncertainties in object pose estimation. To this end, the controller notices unilateral object contact and pauses finger-closing until a force-closure grasp can be guaranteed. In this fashion, undesired object displacement during the grasping phase is minimized or completely avoided. Additionally, we propose a grasp force controller for the holding phase, which is applicable to a large variety of end-effectors and tactile sensors. We verify these claims in a set of real-world experiments using the service robot TIAGo with a sensorized parallel-jaw gripper as shown in Fig. 1.

II. Related Work

In a review of human grasping, Johansson and Flanagan [8] highlight the importance of tactile sensations for this skill. They divide the manipulation task into different phases and argue that during each phase, a different controller is responsible for generating the hand movements. The first goal is thereby to acquire a stable object contact and then increasing the force that is exerted on the object. For the choice of the target force, humans rely on previous knowledge which they gather from past manipulation experiences. We follow this approach and divide the grasping task into three phases similar to their reach, load, and hold phases.

[6] developed a force controller for the PR2 robot that is based on findings of [8]. They divide the grasp into different phases where they switch from free position control to force control upon object contact. Their work also covers lifting the object and corrections afterwards, e.g. when object slippage occurs. One of their assumptions is that the robot is always able to position its gripper such that the fingers are equidistant to the object. In contrast, our controller accounts for non-optimal end-effector to object alignment by synchronizing finger movements upon object contact.

Other approaches mathematically motivate how force control can be realized on robot end-effectors [4], [7]. By constraining finger movements depending on their contribution to the force exerted on the object, they avoid undesired object movements thus improving grasp stability. Like in our approach, [4] uses a PI controller to realize force control. However, their method is designed specifically for multi-fingered hands and does not consider finger synchronization as our approach.

Machine learning techniques are also used in manipulation to improve grasping performance in uncertain environments. [9] use SVMs to detect unstable grasps and adjust them to improve grasp stability on a three-fingered gripper. In [10], the authors use contact information from tactile sensors in combination with reinforcement learning. Their results show that using tactile feedback for machine learning methods, grasp robustness is greatly increased in simulation. As with all machine learning methods, generalization is one of the largest challenges. Reproducing these results would therefore be a time-consuming task even for only slightly different robot setups.

III. Grasp Controller

We propose a holistic control approach for grasping, covering all phases from finger closing over establishing contacts to holding (cf. Fig. 2). Consequently, the controller state distinguishes these three phases and applies a different control strategy in each of them. During state I, i.e. when the fingers are approaching the object, the controller performs open-loop trajectory control, following a pre-planned joint trajectory targeting a hand posture that would eventually penetrate the object. However, as soon as a tactile sensor reliably detects contact, i.e. perceives a contact force $f(t)$ just above the noise threshold f_θ, all joints along the kinemetic chain from the sensor link up to the hand base switch into state II. Within this state, these joints stop moving and thus just maintain contact with the object. This avoids undesired object pushing during finger closing in situations where the object is not perfectly centered between the fingers. Alternatively we also considered actively maintaining a contact force at the noise threshold level. However, due to noisy

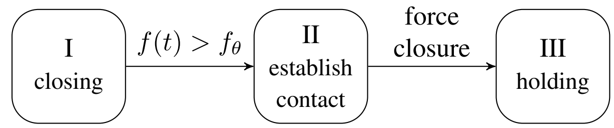

Fig. 2: State diagram illustrating the 3 phases of grasping.

force measurements this still resulted in a continued closing motion, displacing the object in an undesired manner.

Each time a new contact is established, the resulting grasp is evaluated for force-closure [11] as described in sec. III-A. If the grasp is considered force-closure, state III is entered, which finally performs grasp force control aiming to maintain specific contact forces as outlined in sec. III-B. Desired target forces can either be provided by a higher-level cognitive system component based on common-sense object knowledge; fragile objecs, for example, require a lower grasp force than rigid and heavy objects. Alternatively, the grasp forces could be reactively adapted due to recognized incipient slippage as suggested e.g. in [5], [12].

A. Evaluating Force Closure

Force-closure is a formally defined property of a grasp configuration stating that the grasp can balance any (bounded) external disturbance wrench with contact forces $\mathbf{f}_i$ that satisfy the friction cone constraints, i.e. don't induce local slippage. To determine the net wrench applied onto an object through contact points $\mathbf{p}_i$, independent contact forces $\mathbf{f}_i$ are mapped onto 6-dimensional contact wrenches $\mathbf{F}_{C_i} = (\mathbf{f}_i, \tau_i)$ expressed in local contact frames C_i, which are then transformed into the common object frame O using $Ad^\top_{T^{-1}_{oc_i}}$, where they are finally summed up to yield the net wrench F_O expressed in frame O [11]:

$$\mathbf{F} = \sum_{i=1}^{n} Ad^\top_{T^{-1}_{oc_i}} \cdot \mathbf{F}_i = \sum_{i=1}^{n} Ad^\top_{T^{-1}_{oc_i}} B_i \cdot \mathbf{f}_i \equiv G \cdot \mathbf{f} \tag{1}$$

$$B^\top = \begin{bmatrix} 1&0&0&0&0&0 \\ 0&1&0&0&0&0 \\ 0&0&1&0&0&0 \\ 0&0&0&0&0&1 \end{bmatrix} \qquad Ad^\top_{T^{-1}_{oc_i}} = \begin{pmatrix} R_{oc_i} & \\ \hat{\mathbf{p}}_{oc_i} \cdot R_{oc_i} & R_{oc_i} \end{pmatrix}$$

Here, $\mathbf{f} = [\mathbf{f}_1^\top, \ldots, \mathbf{f}_n^\top]^\top$ denotes the vector of concatenated contact forces $\mathbf{f}_i$, G denotes the grasp matrix, and $B \equiv B_i$ is the wrench basis modelling soft-finger contacts [11]. The latter is a point-contact model allowing for force transmission along normal ($f_{i,z}$) and tangential ($f_{i,x}, f_{i,y}$) axes as well as for torque transmission along the normal axis of the contact ($f_{i,\tau}$). These independent force components $\mathbf{f}_i \in \mathbb{R}^4$ need to satisfy friction cone constraints FC_i derived from Coulomb friction to avoid slippage:

$$0 \leq \sqrt{f_{i,x}^2 + f_{i,y}^2} \leq \mu_i f_{i,z} \quad \text{and} \quad |f_{i,\tau}| \leq \mu_{i,\tau} f_{i,z} \,.$$

A grasp defined by its grasp matrix G and the collection of all friction cone constraints $FC = FC_1 \times \ldots \times FC_n$ is force-closure, iff G is surjective and there exist strict internal forces, i.e. forces $\mathbf{f}_N$ that strictly satisfy the friction cone constraints and that don't have a net effect onto the object ($G \cdot \mathbf{f}_N = 0$). Both conditions can be easily verified given the geometry of the grasp, i.e. its contact locations $\mathbf{p}_i \equiv$

$\mathbf{p}_{oc_i}$ and contact normals $\mathbf{n}_i$ (determining R_{oc_i}), as well as a conservative estimation of the friction coefficients $\mu \equiv \mu_i$.

B. Grasp Force Control

We assume that tactile sensors only provide contact forces along the normal direction at the contact point. This rather weak assumption allows applying the grasping force controller to a large variety of tactile sensors [2], but restricts the precision of the achievable control result. Traditional grasp force controllers strive for a globally optimal contact force distribution ensuring grasp stability, i.e. ensuring that (i) all contact forces stay within friction cone bounds, (ii) applied forces exactly balance external forces (e.g. gravity), and (iii) local contact forces are minimized [11]. However, this approach is only meaningful if the full 3D contact force is controllable and thus measurable. Here, we assume we cannot measure shear forces nor determine the friction coefficients required to evaluate friction cone constraints. Further, we assume that force measurements are noisy and poorly calibrated – a reasonable assumption for many high-density tactile sensor arrays. This means in particular that antipodal forces being perfectly in balance in the physical world might result in unequal force sensor measurements or, phrased the other way around, that given balanced force measurements, the actual physical forces might not perfectly sum up to zero, thus resulting in a net force applied to the grasped object, eventually inducing drift.

Consequently, we are not primarily aiming at realizing a zero net wrench onto the object as proposed in traditional grasp force controllers like [7]. Rather, our starting point is the realization of desired contact (normal) forces f_i^{goal} for individual contact sensors i. Only as a subordinate control objective, we consider a zero net force, thus avoiding object drift during holding.

Given the desired normal force magnitudes f_i^{goal} and the normal directions $\mathbf{n}_i$ at the contact locations $\mathbf{p}_i$, we can determine the corresponding 3-dimensional force vector as follows:

$$\mathbf{f}_i^{\text{goal}} = f_i^{\text{goal}} \cdot \mathbf{n}_i \tag{2}$$

To simplify notation, we consider all vectors being represented with respect to a common coordinate frame, namely the end-effector base frame, if not otherwise stated. Within this work, we further assume that the end-effector is position-controlled. Consequently, we transform force deltas into Cartesian position deltas according to Hooke's law:

$$\Delta\mathbf{x}_i(t) = \frac{1}{k}\Delta\mathbf{f}_i(t) = \frac{1}{k}(f_i^{\text{goal}} - f_i(t)) \cdot \mathbf{n}_i\,, \tag{3}$$

where k characterizes the object's stiffness. Estimating k accurately is difficult in practice, thus we utilize PI control to account for uncertainties in estimating k:

$$\mathbf{u}_i(t) = K_{\text{P}} \cdot \Delta\mathbf{x}_i(t) + K_{\text{I}} \int_{\text{III}} \Delta\mathbf{x}_i(\tau)\, d\tau \tag{4}$$

Here, K_{P} and K_{I} denote the coefficients for the proportional and integral terms and the integral is calculated over all time steps since the transition into phase III. If f^{goal} is not reached, the increasing integral term will push the controller to do so regardless of erroneous estimates of k.

Finally, joint position deltas are computed via inverse velocity kinematics, employing the pseudo-inverse of the overall sensor Jacobian $J(\mathbf{q})$:

$$\Delta\mathbf{q}(t) = J^{\dagger} \cdot \mathbf{u}(t)\,, \tag{5}$$

where the Jacobian $J(\mathbf{q})$ maps joint velocities $\dot{\mathbf{q}}$ onto Cartesian velocities $\dot{\mathbf{x}} = [\dot{\mathbf{x}}_1^\top, \ldots, \dot{\mathbf{x}}_n^\top]^\top$ of all sensor frames $i = \{1, \ldots, n\}$. $\mathbf{u} = [\mathbf{u}_1^\top, \ldots, \mathbf{u}_n^\top]^\top$ denotes the concatenated control vector. The commanded joint positions eventually are determined by simple integration:

$$\mathbf{q}^{\text{cmd}}(t) = \mathbf{q}(t) + \Delta\mathbf{q}(t)\,. \tag{6}$$

C. Drift Compensation

To compensate for object drift caused by poor estimation of actual 3d force vectors as outlined above, we suggest to superimpose force control with object-pose control. This can be done based on visual feedback as suggested in [13] or based on a virtual object frame estimated from initial contact points as suggested in [7]. We didn't yet implement this drift compensation, but leave this for future work.

D. Implementation

In order to be compatible with most modern robots, we implemented the grasp controller using ROS [14], the most commonly used robotics middleware. The default controller library ros_control [15] offers a JointTrajectoryController (JTC), which performs free position control for a set of joints given a trajectory. Our controller inherits from the JTC class keeping all its interfaces. Hence, a grasp can be initiated by sending a JointTrajectoryActionGoal in the same fashion a JTC task would have been triggered. Therefore, our controller can easily be integrated into existing grasping pipelines without much effort. Parameters specific to force control can be adapted via designated ROS services advertized by our controller.

The controller offers two different modes for phase III: an open-ended force control mode and a mode finishing control upon reaching the desired force. For the former one, the controller attempts to hold the desired force forever. During this mode, the desired force can still be changed, e.g. to raise grip strength while moving the arm. The latter mode signals action success once all tactile sensors reach their desired force.

IV. Experiments

We evaluated the control approach on the TIAGo robot, a service robot developed to serve in household settings. Hence, manipulation is one of its most important capabilities. It has a 7 DoF arm with end-effectors that can be easily exchanged or modified, making it an extensible and adaptable manipulation platform. It is available either with a parallel-jaw gripper or a five-fingered hand. For our experiments, we decided for the parallel-jaw gripper as it (i) allows for a larger gripper opening, (ii) provides a higher grip force, and (iii) allowed for easy integration of simple tactile sensors.

TABLE I: Statistics of uncalibrated sensor readout.

Finger	μ	σ	max
Left	0.01577	0.00459	0.03322
Right	0.00273	0.00367	0.01777

By default, TIAGo's gripper has two plastic fingers that are *not* sensorised. In order to measure contact forces at the fingers, we designed a new set of fingers employing load cell/strain gauge sensors as the finger "bones" mounted to the gripper's base and augmented with a finger tip as shown in Fig. 1. These sensors measure the force applied to their tips using a Wheatstone Bridge. A LabJack U6 is used to convert the analog sensor signals to digital ones. TIAGo's internal computer handles the sensor acquisition employing LabJack's Python API using the parameters $resolutionIndex = 0$ and $gainIndex = 3$. We found this combination to yield the best results in terms of sampling frequency and noise. Finally, the sensor values are bias-compensated and published on a ROS topic.

To this end, we determined the sensor bias from 10.000 raw sensor readings, when no load was applied to the sensors. The bias statistics is summarized in table I. Based on the highest measured values, we also set a conservative noise threshold $f_\theta = 0.21N$ to avoid false-positive contact detections. Choosing a well-suited value is crucial for the controller's performance. A too tight threshold can lead to premature stopping of the finger-closing phase, while a too conservative threshold greatly affects the controller's sensitivity. In practice, the thresholds have to be changed very rarely as they have shown to be rather stable across robot restarts.

To assess the sensor's first-touch sensitivity, we aligned them parallel to the ground and placed small weights (different Euro coins) on their tips until the measured sensor value was continuously above the threshold f_θ for a period of 30 seconds. In this fashion, we determined first-touch sensitivities of $0.35N$ and $0.38N$ for the left and right sensors respectively, which is roughly twice as much as f_θ. As the difference in sensitivity is rather small, we can safely assume that the measurements from both sensors are comparable, thus avoiding the need for explicit drift compensation.

As the gripper's fingers are directly opposing each other, the control algorithm is significantly simplified. Firstly, force-closure is automatically guaranteed as soon as both fingers establish contact. Secondly, the sensor Jacobian is the identity matrix, representing a one-to-one relation between Cartesian sensor displacements and joint motions, thus avoiding the need to compute the inverse kinematics.

To assess the capability of the grasp controller to perform robust object grasps given uncertain object pose estimations, we performed a series of experiments, comparing the proposed controller to an open-loop baseline, which is what most robot platforms nowadays use by default. Our main focus is whether the robot can manipulate the object robustly, i.e. neither damaging nor tipping over the object. To this end, we present three different experiments which evaluate different aspects of the controller.

In experiment **1**, we show that the controller is able to handle objects more carefully than a controller without force feedback. TIAGo starts with its arm in a pre-grasp pose, its end-effector is positioned a few centimeters above the surface of a table with its fingers in the open position. Then, a soft object is placed in the center between the fingers and the gripper is closed using either the JTC or the proposed controller. The experiment is repeated three times with each controller and the object diameter is measured after closing the gripper. We expect that the JTC will exert more force on the object than the force controller and therefore has a higher chance of damaging the object.

Experiment **2** investigates the controller's robustness with respect to object position uncertainty. In real-world scenarios, the end-effector is rarely centered perfectly around the object, especially with mobile robots estimating the object pose by vision. Thus, with experiment **2**, we analyze the behavior of the controller in situations where the initial gripper placement is non-optimal. The experiment setup is similar to **1**, with the difference that the objects are not any longer centered between the fingers, but placed closer to one of the fingers. The object displacement during grasping is measured using millimeter paper that is fixed to the table surface. When detecting the first touch with the closer finger, the controller should stop the motion of this finger until the second finger acquired object contact as well. Hence, we assume that the new controller will cause much less undesired object displacement in contrast to JTC. The most important part of this experiment is to detect the first touch as soon as possible and to do so reliably. This depends on a variety of factors: quality of the calibration, choice of the noise threshold, and sensor sensitivity in general. We repeat this experiment with three objects that have different weights in order to assess the sensitivity of our force sensors during grasping. Each controller grasps each object three times with a constant offset to one of the fingers, resulting in $3 \cdot 3 \cdot 2 = 18$ trials.

With experiment **3** we demonstrate the controller's robustness with respect to external forces applied to the object while trying to maintain a target force. The setup is the same as in **1**, however the controller is instructed to maintain the target force instead of finishing the control task. Once the robot grasped the object, the target force is changed to show how the controller adapts to target changes. Afterwards, random external forces are exerted onto the object and the robot's fingers by a human.

During preliminary controller tests, we have determined the PI coefficients $K_\mathrm{P} = 1.9$ and $K_\mathrm{I} = 3.2$. The coefficients were tuned in such a way that overshooting the target force is prevented. This can lead to a slightly slower convergence to the target force but ensures that the object is not damaged. As humans heavily rely on accurate previous knowledge about the object and its stiffness [8], we also estimated k for each object in preliminary trials. To this end, k was chosen rather high initially, resulting in small position deltas

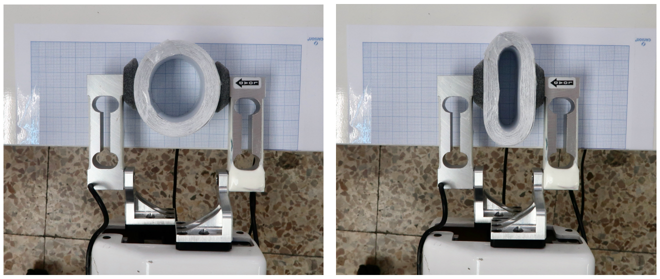

Fig. 3: Comparison of the final gripper postures of the proposed grasp controller (left) and the JTC (right).

TABLE II: Object displacements for different object weights. Displacements are measured in millimeters.

Object	**Weight**	k	**JTC**	**our controller**
Styrofoam	$2g$	800	15.8 ± 0.8	15.6 ± 0.8
Tape roll	$49g$	1000	12.3 ± 1.3	**4.9** ± 1.9
Glass Bottle	$265g$	2500	9.3 ± 0.7	**1.6** ± 0.5

to prevent object damage. If convergence was too slow it was decreased, if the controller exhibited overshooting, it was reduced. This process has proven to be rather quick, requiring only about five trials per object. The used stiffness values are summarized in Table II.

V. Experimental Results

In experiment **1**, we evaluated whether our controller is able to handle objects with more care than JTC. To this end, we used the tape roll shown in Fig. 3 with a nominal diameter of $51mm$. After grasping, the object diameter was reduced to $28 \pm 1.5mm$ and $47 \pm 1mm$ for the JTC resp. our controller, averaged over three consecutive trials. Thus, our controller reduced the object diameter by 10% while JTC reduced it by 46% on average. This confirms our assumption that the proposed grasp controller is able to regulate forces during grasping, thus preventing object damage.

For experiment **2**, we used three objects with different weights. Table II shows the average distance each controller moved the object in all six trials. For the first object, the styrofoam cylinder, the displacement for both controllers is very similar: this object was too lightweight to trigger the first-touch detection on the force sensors. For the other two objects, the displacement was significantly reduced. The experiments prove, that for fairly light objects like the tape roll ($49g$), TIAGo's force sensors can reliably detect first touch. In all cases the standard deviation is fairly small, indicating a high degree of repeatability of the grasping results.

Figure 4 compares forces and joint positions produced during grasping with both controllers. Important events are denoted by vertical dashed lines: establishing the first contact with a finger, establishing the second contact with the opposing finger, and the end of the control task. As an open-loop controller, JTC commands continuously decreasing joint positions (corresponding to fingers closing) regardless of measured forces. The first contact isn't noticed or considered. After the second contact is established, the contact forces quickly increase beyond the desired level f^{goal}. Interestingly, the force level decays in the following, which is presumably due to a protection mechanism implemented in the low-level gripper controller: For a short time the gripper can apply the large forces observed, but then reduces motor currents to self-protect from overheating.

With force control enabled, one can nicely observe that the first finger stops its motion once it notices object contact. Only if the opposing finger also acquires contact, the applied forces increase smoothly until f^{goal} is reached. Note that the commanded joint positions are slightly smaller than actual joint positions, which is primarily due to the integral term of the PI controller.

The results from experiment **2** are twofold: it shows that using force control, we can minimize undesired side effects of object manipulation during the grasping stage. As a consequence, our object manipulation becomes much safer and more predictable. Furthermore, it shows that the performance of any force controller highly depends on the capability to detect first touch. That capability is in turn dependent on the sensor sensitivity as well as its calibration. These requirements should therefore be considered carefully when choosing the sensor type and during its integration. As sensor development progresses and improves, we expect to see more sensitive sensors. Thus, with improved hardware, force controller performance will improve as well, because first touch can be detected earlier and more reliably.

In experiment **3**, TIAGo grasped an easily deformable object on a table in force maintainance mode. We illustrate the outcome of this experiment with a video where all the described behaviors can be seen. After grasping and convergence of contact forces, we command a higher target force f^{goal}. As can be seen from the plots embedded in the video, the controller quickly reacts to this change without overshooting the new target. Afterwards, the controller was disturbed by randomly applying external forces on the robot's fingers and onto the object. Here, the controller's response strongly depends on the choice of K_{I}. For moderate $K_{\text{I}} = 3.1$, the controller smoothly moves both joints to give way and match the desired force again. Increasing K_{I} too much (in our case 6.2) caused the object to drift in the robot's gripper even after the external disturbance force vanished. Only if a joint hits its joint-limit the drifting motion stops. Although f^{goal} is maintained all the time, this overreaction to external forces leads to undesired object movements. In future work, we will investigate drift compensation as outlined in sec. III-C to improve on this situation as well.

VI. Conclusion And Future Work

In this work, we have presented a holistic grasp controller that combines free position control during finger closing with force control during holding. Based on the mathematical foundations of force-closure grasps and inspired by human grasping behavior, we implemented a controller that minimizes undesired object displacement by stopping the finger closing motion as soon as object contact is detected. In the

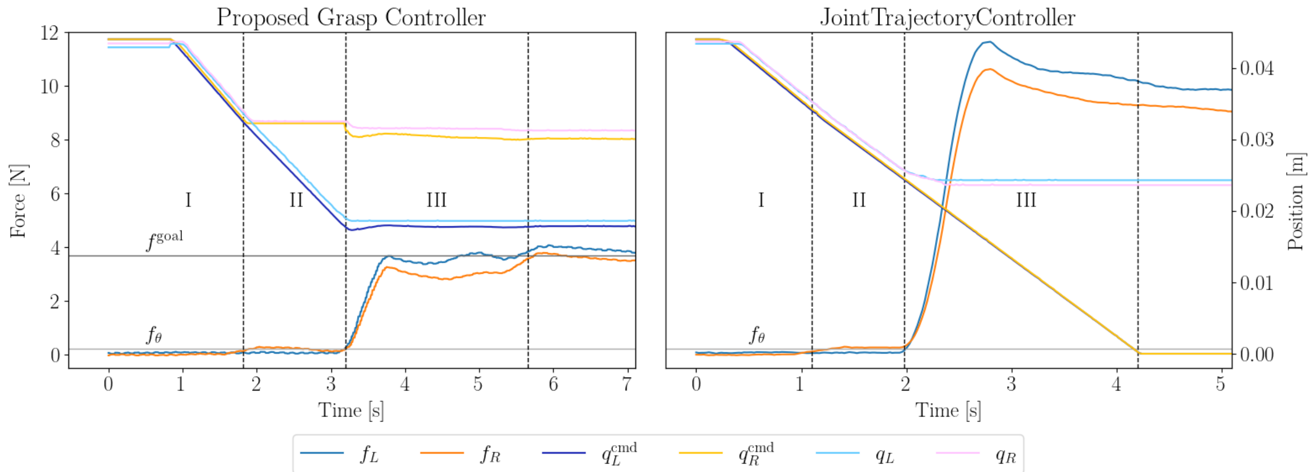

Fig. 4: Commanded joint positions and measured forces during a grasp.

object holding phase, desired contact forces are maintained with finger-local controllers, which turned out to be sufficient to avoid object drift due to unbalanced net forces in most grasping situations.

Our experiments have shown that our controller is able to handle objects much more gently and that it minimizes undesired object movements, given that is able to sense the object contact. Additionally, if tuned correctly, it is also able to withstand external forces exerted on the object or joints.

Although demonstrated for a simple parallel-jaw gripper only, the approach generalizes to multi-fingered hands, an aspect to be investigated in future work in more detail. Another interesting aspect to investigate in future, is how to avoid undesired object drift after applying external disturbance forces as observed in experiment **3**. Furthermore, we plan to estimate the object's stiffness k during grasping, avoiding the need for prior estimation of this parameter.

References

[1] J. Stückler, R. Steffens, D. Holz, and S. Behnke, "Efficient 3d object perception and grasp planning for mobile manipulation in domestic environments," *Robotics and Autonomous Systems*, vol. 61, no. 10, pp. 1106–1115, 2013.

[2] Z. Kappassov, J.-A. Corrales, and V. Perdereau, "Tactile sensing in dexterous robot hands," *Robotics and Autonomous Systems*, vol. 74, pp. 195–220, 2015.

[3] P. S. Girão, P. M. P. Ramos, O. Postolache, and J. M. D. Pereira, "Tactile sensors for robotic applications," *Measurement*, vol. 46, no. 3, p. 1257, 2013.

[4] Q. Li, R. Haschke, H. Ritter, and B. Bolder, "Towards unknown objects manipulation," *IFAC Proceedings Volumes*, vol. 45, no. 22, pp. 289–294, 2012.

[5] G. Walck, R. Haschke, M. Meier, and H. Ritter, "Robot Self-Protection by Virtual Actuator Fatigue: Application to Tendon-driven Dexterous Hands During Grasping," in *Proc. IROS*, Vancouver: IEEE, 2017.

[6] J. M. Romano, K. Hsiao, G. Niemeyer, S. Chitta, and K. J. Kuchenbecker, "Human-inspired robotic grasp control with tactile sensing," *IEEE Trans. on Robotics*, vol. 27, no. 6, pp. 1067–1079, 2011.

[7] K. Tahara, S. Arimoto, and M. Yoshida, "Dynamic object manipulation using a virtual frame by a triple soft-fingered robotic hand," in *Proc. ICRA*, 2010.

[8] R. S. Johansson and J. R. Flanagan, "Coding and use of tactile signals from the fingertips in object manipulation tasks," *Nature Reviews Neuroscience*, vol. 10, no. 5, pp. 345–359, 2009.

[9] H. Dang and P. K. Allen, "Stable grasping under pose uncertainty using tactile feedback," *Autonomous Robots*, vol. 36, no. 4, pp. 309–330, 2014.

[10] H. Merzić, M. Bogdanović, D. Kappler, L. Righetti, and J. Bohg, "Leveraging contact forces for learning to grasp," in *Proc. ICRA*, IEEE, 2019.

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

[12] R. Haschke, "Grasping and Manipulation of Unknown Objects Based on Visual and Tactile Feedback," in *Motion and Operation Planning of Robotic Systems*, G. Carbone and F. Gomez-Bravo, Eds. Springer, 2015, vol. 29, p. 522.

[13] Q. Li, R. Haschke, and H. Ritter, "A Visuo-Tactile Control Framework for Manipulation and Exploration of Unknown Objects," in *Proc. Humanoids*, 2015.

[14] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, A. Y. Ng, *et al.*, "ROS: An open-source robot operating system," in *ICRA workshop on open source software*, Kobe, 2009.

[15] S. Chitta, E. Marder-Eppstein, W. Meeussen, V. Pradeep, A. Tsouroukdissian, J. Bohren, D. Coleman, B. Magyar, G. Raiola, M. Lüdtke, and E. Perdomo, "ros_control: A generic and simple control framework for ROS," *Journal of Open Source Software*, vol. 2, p. 456, Dec. 2017.