

# A Learning-based Controller for Multi-Contact Grasps on Unknown Objects with a Dexterous Hand

Dominik Winkelbauer<sup>1,2</sup>, Rudolph Triebel<sup>1,3</sup>, Berthold Bäuml<sup>1,2</sup>

**Abstract**—Existing grasp controllers usually either only support finger-tip grasps or need explicit configuration of the inner forces. We propose a novel grasp controller that supports arbitrary grasp types, including power grasps with multi-contacts, while operating self-contained on before unseen objects. No detailed contact information is needed, but only a rough 3D model, e.g., reconstructed from a single depth image. First, the external wrench being applied to the object is estimated by using the measured torques at the joints. Then, the torques necessary to counteract the estimated wrench while keeping the object at its initial pose are predicted. The torques are commanded via desired joint angles to an underlying joint-level impedance controller. To reach real-time performance, we propose a learning-based approach that is based on a wrench estimator- and a torque predictor neural network. Both networks are trained in a supervised fashion using data generated via the analytical formulation of the controller. In an extensive simulation-based evaluation, we show that our controller is able to keep 83.1 % of the tested grasps stable when applying external wrenches with up to 10 N. At the same time, we outperform the two tested baselines by being more efficient and inducing less involuntary object movement. Finally, we show that the controller also works on the real DLR-Hand II, reaching a cycle time of 6 ms. Website: [aidx-lab.org/grasping](http://aidx-lab.org/grasping)

## I. INTRODUCTION

Robust grasping is one of the most important skills a robot needs to be able to interact with the world. Planning the grasp, meaning positioning the hand and the fingers such that a stable grasp originates, has been the focus of extensive research [1, 2, 3, 4, 5]. However, especially when using multi-fingered hands, setting the correct torques to apply at each joint is also crucial for the stability of the grasp. These torques need to be adapted to changing external wrenches and, at the same time, the torques should only be as large as necessary, to be energy-efficient and to keep the load on object and hand as small as possible. Existing approaches that control the torques depending on a given grasping situation usually only support fingertip grasps where the fingers are in a non-singular position and the contact positions and normals are known [6, 7, 8, 9]. However, in reality, the opposite is usually the case: Many grasps, especially power grasps, lead to multi-contacts across multiple links. The object might be in contact with the fixed hand base or fingers might be in singular configurations. Also, detailed contact information is not

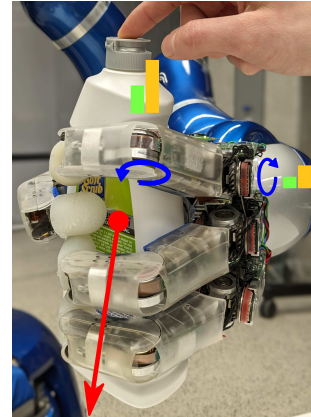


Fig. 1. The controller applied to a grasp on the YCB bleach bottle. The applied external wrench (red arrow) gets estimated and then the commanded torques are adjusted. Here, two joints are visualized with their applied torque before (green) and after (orange) the external wrench is applied.

available. In this work, we are approaching the grasp control problem without any constraints on the grasp or object. We propose a two-step approach, shown in Fig. 1: First, based on the measured torques, the external wrench acting on the object gets estimated. Then, the desired joint angles given to the underlying joint-level impedance controller are adjusted to counter the estimated wrench. Our controller can handle unknown objects without any human supervision and only uses the available sensory information, which includes the measured torques at each joint, the measured joint angles and the object observed via a depth camera. By using learning-based methods, we are able to run the controller in real-time and do not need any explicit detailed contact information. In summary, we make the following novel contributions:

- We propose a novel grasp controller that is able to estimate external wrenches using the measured joint torques and at the same time able to adapt the torques such that the wrench is countered
- We provide an analytical formulation of the controller based on an elastic model that is able to handle all kinds of grasps including power grasps with multi-contacts and grasps in singular positions
- We further propose a learning-based formulation that allows running the controller in real-time and without detailed contact information
- To be able to handle ambiguous situations, we propose to train the wrench estimation network using a novel loss which makes the network predict always the smallest wrench explaining a given situation

<sup>1</sup>DLR Institute of Robotics & Mechatronics, Germany

<sup>2</sup>Learning AI for Dexterous Robots Lab ([aidx-lab.org](http://aidx-lab.org)), Technical University of Munich, Germany

<sup>3</sup>Karlsruhe Institute of Technology (KIT), Germany

Contact: [Dominik.Winkelbauer@dlr.de](mailto:Dominik.Winkelbauer@dlr.de)

\*This work is supported by the Helmholtz Association under the joint research school “Munich School for Data Science - MUDS”.

## II. RELATED WORK

Grasping unknown objects using multi-fingered hands is becoming more and more popular, especially using data-driven methods. While also the correct configuration of the grasp forces is a crucial part of the grasping process [3], most methods rely on a simple heuristic, e.g. apply the same amount of force in each finger [1, 2, 3, 4]. We clearly outperform such heuristics by being more efficient and inducing less undesired object movement.

There exist methods that provide a more sophisticated way of controlling grasps. One of the most prominent approaches is the *object-level impedance controller*, introduced first by Schneider and Cannon [6]. Usually, such controllers are combined with the virtual linkage model [10], to be able to define the internal forces without having explicit knowledge of the contacts. The concept of the object-level impedance controller has been later extended with a novel damping design and to not require a simulation of a virtual object [8]. However, the methods mentioned above require the user to configure the virtual linkage based on the grasped object. Instead, our proposed method sets internal forces autonomously for unknown objects.

Pfanne et al. [9] proposed to use vision-based contact detection which allows determining the correct internal forces. However, their method only supports fingertip grasps, which is a general issue with object-level impedance controllers: They require the object to be able to move in each direction, however, in power grasps not much object movement is possible. In our approach, we only rely on the measured joint torques, which requires only little object movement.

Another way of object-stability control in grasping is by using a tactile sensor. Here, the main idea is usually to increase the inner forces as soon as a slip is detected [11, 12, 13, 14]. Again, these methods only support fingertip grasps, because the tactile sensor is usually only available at the fingertip and the proposed adaptation of the inner forces does not support multi-contacts.

Bicchi [15] recognized that in power grasps the methods used for fingertip grasps do not apply anymore and instead proposed an elastic model to solve the indeterminacy of contact forces. However, their approach uses a linearized formulation [16], while we propose a non-linear formulation which is harder to optimize but physically valid. Abdeetetal et al. [17] extended the approach of Bicchi [15] to support kinematic uncertainties. Both approaches need tactile sensors on all links to be able to estimate the applied external wrench via the measured contact forces. Instead, our approach only relies on the torques measured in each joint to do so.

## III. GRASP CONTROLLER

We look at the problem of controlling a robotic hand, that is grasping an object  $o$ , in a way that the object robustly stays in its initial pose, while changing external wrenches  $w \in \mathbb{R}^6$  act upon it. Fig. 3 shows the torque-controlled *DLR-Hand II* [18], that is used. The hand has  $L = 12$  actuated joints, three per finger. The grasp is defined by a 6D hand pose  $H$  and a joint configuration  $q_{\text{grasp}} \in \mathbb{R}^L$ , predicted by our

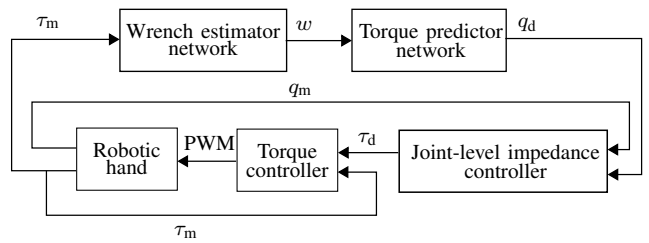


Fig. 2. Block diagram of our proposed controller. The controller is built on a joint-level impedance controller which commands the underlying torque controller based on the given desired joint angles. The core of our controller consists of the estimation of the external wrench based on the measured joint torques and the subsequent prediction of the desired joint angles which are necessary to counter the external wrench.

grasping network [5]. After applying the pre-grasp and going into contact with the object, the grasp controller is activated starting from the resulting joint configuration  $q_{\text{init}}$ . Afterward, the grasp is assumed to be quasi-static, which allows us to ignore inertial effects. The object frame is attached to its center of mass, however this is an arbitrary choice, as our approach is irrespective of the definition of the object frame.

### A. Overview

In the DLR-Hand II, each joint has a torque sensor, allowing the hand to be controlled via a set of torques  $\tau \in \mathbb{R}^L$ . This is done using a torque controller that adapts the PWM signal of each motor based on the measured torques  $\tau_m$  and a set of desired torques  $\tau_d$ . We base our controller (see Fig. 2) further on a joint-level impedance controller, which is defined as

$$\tau_d = k_p(q_d - q_m) - k_d\dot{q}_m, \quad (1)$$

with  $\dot{q}_m$  being the measured joint velocities,  $q_d$  being the desired,  $q_m$  being the measured joint angles and  $k_p / k_d$  being the stiffness / damping coefficient. This allows us on the one side to command torques by adapting  $q_d$ , while still being able to measure changing torques  $\tau_m$ . Specifically, in our grasp controller, we first estimate  $w$  based on  $\tau_m$  and in a second step adjust  $q_d$  such that the external wrench is counteracted. We have these two explicit steps because it is easier to approach each problem isolated and it makes the controller more explainable, as one can inspect the estimated external wrench.

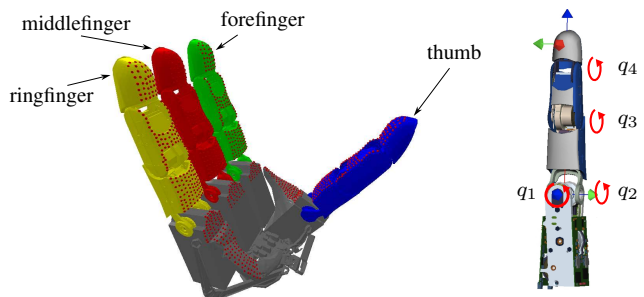


Fig. 3. The DLR-Hand II [18] which we use in our experiments. Each red dot represents a potential contact point. The finger on the right shows the four joints ( $q_3$  and  $q_4$  are coupled:  $q_3 = q_4$ )

## B. Elastic formulation

As we are targeting all kinds of grasps, including power grasps, we use an elastic formulation to handle the indeterminacy of contact forces [15, 19]: The given grasp results in  $N$  hand-object contacts, described by their position  $p_i$  and object normal  $n_i$ . In a stable grasp, the contact forces  $\{f_1, \dots, f_N\}$  counter the external wrench

$$w = -Gf, \quad (2)$$

and cause the torques  $\tau$  to be applied at the joints via

$$\tau = J^T f. \quad (3)$$

Here,  $G \in \mathbb{R}^{6 \times 3N}$  describes the grasp matrix,  $J \in \mathbb{R}^{L \times 3N}$  the jacobian and  $f \in \mathbb{R}^{3N}$  the stacked contact forces. We model the contact points to be slightly elastic, meaning that a spring is placed at each contact point, whose displacement  $d \in \mathbb{R}^{3N}$  is defined by small displacements to the object pose  $\Delta x \in \mathbb{R}^6$  and to the joint angles  $\Delta q \in \mathbb{R}^L$ :

$$d = G^T \Delta x - J \Delta q + d_{\text{init}}. \quad (4)$$

Here,  $d_{\text{init}}$  describes the displacement at the starting point of the controller. Based on the stiffnesses  $k_c \in \mathbb{R}^N$  of the contact springs, the contact displacements  $d$  relate to the contact forces  $f$ . The normal forces  $f_n$  are defined as

$$f_{n,i} = \begin{cases} k_{c,i} d_{n,i} & \text{if } n_i \cdot d_{n,i} < 0 \\ 0 & \text{if } n_i \cdot d_{n,i} \geq 0 \end{cases}, \quad (5)$$

with  $d_{n,i}$  being the the displacement along the contact normal  $n_i$ . The tangential forces  $f_t$  are defined as

$$f_{t,i} = \begin{cases} k_{c,i} d_{t,i} & \text{if } k_{c,i} \|d_{t,i}\| < \mu \|f_{n,i}\| \\ \mu \|f_{n,i}\| \frac{d_{t,i}}{\|d_{t,i}\|} & \text{else} \end{cases}, \quad (6)$$

with  $d_{t,i}$  being the displacement along the contact tangentials and  $\mu$  being the friction coefficient. For further explanations regarding the elastic formulation above, we refer to Winkelbauer et al. [19].

Due to the usage of the joint-level impedance controller (see (1)), the relation between torques and joint displacements is further fixed to

$$\tau = k_p(q_d - (q_{\text{init}} + \Delta q)). \quad (7)$$

## C. Wrench optimization

Given measured torques  $\tau$  and desired joint angles  $q_d$ , the external wrench  $w$  applied to the object should be determined. For grasps that contain singularities or where the object is in contact with the static hand base, there can be infinite possible  $w$  that fit the given measurements. To resolve this ambiguity, we desire to find the smallest external wrench  $w$ . We found that this improves the stability of the controller as no non-existent wrenches are hallucinated. In detail, the optimization problem is defined as

$$\begin{aligned} \min_{\Delta x, \Delta q} \quad & \|w_f\|^2 \\ \text{s.t.} \quad & (2) \text{ to } (7) \text{ are fulfilled,} \end{aligned} \quad (8)$$

with  $w_f$  being the 3D force part of the wrench.

## D. Torque optimization

Based on the estimated external wrench  $w$ , the desired joint angles  $q_d$  are adapted in the second step. Here, we add the additional constraint

$$\forall l \sum_{i \in \mathcal{F}_l} \mu \|f_{n,i}\| - \|f_{t,i}\| > v, \quad (9)$$

with  $v$  being the minimum friction margin,  $l$  being the finger index and  $\mathcal{F}_l$  being the set of all contacts placed on the tip of finger  $l$ . This makes sure, that each fingertip has contact forces with a minimal margin to the friction cone and therefore the fingers do not slip off the object in the case of a changing external wrench. Again, usually an infinite amount of possible  $q_d$  can be found. Identical to Winkelbauer et al. [19], we want to be as efficient as possible and therefore find the  $q_d$  which leads to the smallest torques: So, the optimization problem is defined as

$$\begin{aligned} \min_{\Delta x, \Delta q} \quad & \|\tau\|^2 \\ \text{s.t.} \quad & (2) \text{ to } (7) \text{ and } (9) \text{ are fulfilled} \end{aligned} \quad (10)$$

## IV. LEARNING-BASED FORMULATION

The analytical formulation of the controller, described in Section III, cannot be run in real-time on the real robot, as both steps require computationally expensive non-linear optimizations (the torque optimization takes ca. 200 s per iteration on a single CPU core). Further, exact contact points and normals are not known when grasping unknown objects. That is why we propose to learn these two steps, each using a neural network. While the offline training of the networks is still computationally expensive, the online inference is fast, which allows us to run the controller in real-time.

### A. Wrench estimation

The wrench estimation network  $\mathcal{W}$  learns the mapping

$$\mathcal{W} : \{o, q_{\text{grasp}}, q_{\text{init}}, q_d, \tau_m\} \rightarrow \{w\}. \quad (11)$$

1) *Architecture*: The input to the network consists of the joint angles predicted by the grasping network  $q_{\text{grasp}}$ , the measured joint angles  $q_{\text{init}}$  after applying the pre-grasp, the current desired joint angles  $q_d$ , the measured torques  $\tau_m$  and the object observation  $o$ . The object is transformed into the local coordinate frame of the hand and represented as a  $48^3$  voxelgrid, which is available based on the observed depth image through a preceding shape completion step [20]. Also, the wrench estimated by the network is represented relative to the coordinate frame of the hand, which makes it easier for the network to generalize across grasps. As visualized in Fig. 4a, the network consists of four 3D convolutional layers, which encode the object input  $o$ . Afterwards, the flattened

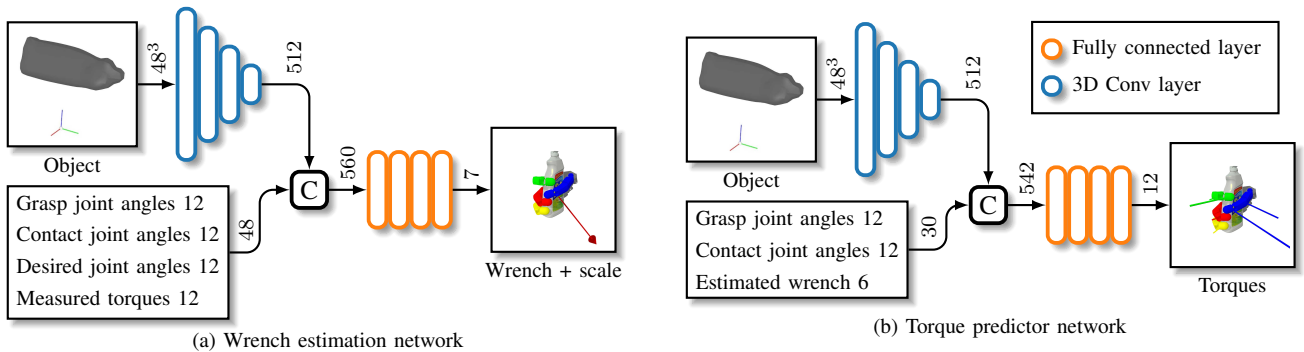


Fig. 4. Architectures of both used networks. The object, represented as a voxelgrid, is encoded using 3D convolutional layers and afterward concatenated with the additional one-dimensional input. A four-layer MLP estimates the external wrench or respectively the desired torques in the end.

object embedding is concatenated with the other inputs and given to four fully connected layers which estimate the wrench  $w$ .

2) *Training data*: The results of solving (8) depend on the ambiguity in the observed data. If perfect contact information and torque measurements are given, the wrench can often be estimated accurately. However, we are interested in the real-world case where neither is available. That is why we solve the objective (8) via the training loss, which considers the actual ambiguity in the data given to the network, and at the same time generate the training data in the following (inverse) way: First, we sample one external wrench  $w$  and one estimated external wrench  $w'$ . Using the estimated one, we use the torque prediction network to predict the corresponding desired joint angles  $q_d = \mathcal{T}(o, q_{\text{grasp}}, q_{\text{init}}, w')$ . Now, using the equation system (2) to (7), we solve for the measured torques  $\tau$  when applying the wrench  $w$  while using  $q_d$ . This results in a dataset of training samples  $\{(\hat{q}_{d,1}, \hat{w}_1), \dots, (\hat{q}_{d,T}, \hat{w}_T)\}$  of size  $T$ .

3) *Loss*: If one would train the network using a standard L2-loss, the network would learn the mean of all wrenches that correspond to a given ambiguous input. This can lead to severe problems at inference time, as the network could hallucinate wrenches. As already mentioned in Section III-C, we therefore always want to estimate the minimal wrench in ambiguous situations. Therefore, we propose a novel learning procedure: We extend the output of the network with an explicit estimation of the wrench scale  $s \in \mathbb{R}$ . To learn the scale of the smallest wrench corresponding to the given input, we use the loss function

$$\mathcal{L}_{\text{scale}} = \sum_{i=1}^T \mathbb{1}_{s_i > \hat{s}_i} |s_i - \hat{s}_i| + \sum_{i=1}^T -ts_i, \quad (12)$$

with  $\hat{s} = \|\hat{w}_i\|$  being the ground truth scale. By setting its derivative to zero, one finds its optimum to be at  $\frac{1}{T} \sum_{i=1}^T \mathbb{1}_{s_i > \hat{s}_i} = t$ . This means, the optimal  $s_i$  is bigger than  $t \in [0, 1]$  part of the ground-truth scales  $\hat{s}_i$  corresponding to a given ambiguous input. The loss function for the wrench estimation is defined as a standard L2-loss with the exception that only training samples are considered where the scale  $\hat{s}_i$  is smaller than the predicted one  $s_i$ :

$$\mathcal{L}_{\text{wrench}} = \sum_{i=1}^T \mathbb{1}_{s_i > \hat{s}_i} \|w_i - \hat{w}_i\|^2. \quad (13)$$

So, by setting  $t$ , one can adjust how precise the network should learn the minimal wrench. A smaller factor  $t$  makes the network go closer to the minimal wrench, however, it will also lead to making use of a smaller subset of the training data. The total loss is defined as  $\mathcal{L}_{\text{scale}} + \mathcal{L}_{\text{wrench}}$ .

### B. Torque prediction

The torque prediction network  $\mathcal{T}$  learns the mapping

$$\mathcal{T} : \{o, q_{\text{grasp}}, q_{\text{init}}, w\} \rightarrow \{q_d\}. \quad (14)$$

1) *Architecture*: So, based on the object  $o$ , the external wrench  $w$ , the initial joint angles  $q_{\text{init}}$  and the grasp joint angles  $q_{\text{grasp}}$ , the network should predict the smallest desired joint angles  $q_d$  which will lead to resisting the external wrench  $w$ . Instead of predicting  $q_d$  directly, we let the network predict the desired torques  $\tau$  and then set the desired joint angles via  $q_d = k_p \tau + (q_{\text{init}} + \Delta q)$ . This has the advantage, that the network is independent of  $k_p$ . As  $\Delta q$  is in our case relatively insignificant due to  $k_p \ll k_c$ , we assume it to be 0. However, if necessary one could also let the network additionally predict  $\Delta q$  or directly  $q_d$ . The architecture of the torque prediction network, visualized in Fig. 4b, is congruent to the wrench estimation network.

2) *Training*: To generate training data, we sample random external wrenches  $w$  and then solve the optimization problem (10). Similar to Winkelbauer et al. [19], we approach the highly non-linear problem using a local gradient-based root solver combined with a global evolutionary optimizer. This results in a training dataset  $\{(\hat{w}_1, \hat{\tau}_1), \dots, (\hat{w}_T, \hat{\tau}_T)\}$ . A standard L2 loss is used to train the network.

## V. EXPERIMENTS

### A. Training

For generating the training data, we use  $\mu = 0.4$ ,  $k_c = 10000 \text{ N m}^{-1}$  across all contacts and  $k_p = 5 \text{ N m rad}^{-1}$ . We generate training data using 15000 objects from the ShapeNet dataset [21]. For each object, we simulate a Kinect-like depth image and apply shape completion to reconstruct the original object [20]. Based on that, we use our grasping

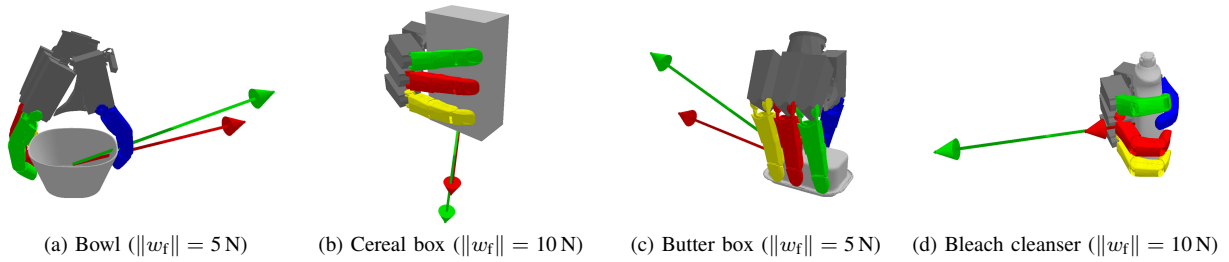


Fig. 5. Qualitative examples of the wrench estimation. Shows the ground truth external wrench in green and the wrench estimated by the network in red.



Fig. 6. The 30 objects used in the evaluation.

network [5] to generate 1000 grasps for each object, from which we select 24. This leads to a total dataset of 360 000 grasps. For each grasp, we simulate the pre-grasp and afterward gather dense contact points (see potential contact points in Fig. 3). For the wrench estimation network, we generate 40 training samples per grasp, as described in Section IV-A.2. The resulting dataset of  $1 \times 10^7$  samples is used to train the network using  $t = 0.05$  for  $2.5 \times 10^6$  iterations. For the torque prediction network, we generate one training sample per grasp, leading to a dataset size of 230 000. The network is trained for 500 000 iterations. Both networks are trained using the Adam optimizer and a learning rate of  $3 \times 10^{-4}$ .

When sampling external wrenches, we use the space of all possible 3D forces and additionally make use of the *object wrench space* (OWS) [22]. In detail, we restrict ourselves to all wrenches that can be caused by a single force being applied to the surface of the object. This has the advantage, that an intuitive scaling between forces and torques is provided. We uniformly sample  $\|w_f\|$  to be between 0 N and 10 N.

### B. Experimental setup

To verify our grasp controller, we evaluate it in simulation across 240 grasps on 30 objects (see Fig. 6). All grasps and objects are unknown to the networks. Each grasp is evaluated 100 times using different random external wrenches  $w$ . For sampling the wrench  $w$  we again take inspiration from the OWS, meaning we sample a point on the surface of the object and apply a force along the object normal. We use  $\|w_f\| = 10$  N, except for precision grasps, where we use  $\|w_f\| = 5$  N to stay inside the torque limits of the joints. The external wrench is applied to the object by linearly scaling it up from 0 to its full size in 4 s. Afterward, the full wrench is kept constant for 4 s. We mark a grasp as broken if the object deviates more than 3 cm or  $20^\circ$  from its initial pose at any point. Fig. 5 shows four exemplar evaluation runs together with the actual and estimated external wrench.

We compare our grasp controller with two baselines: One

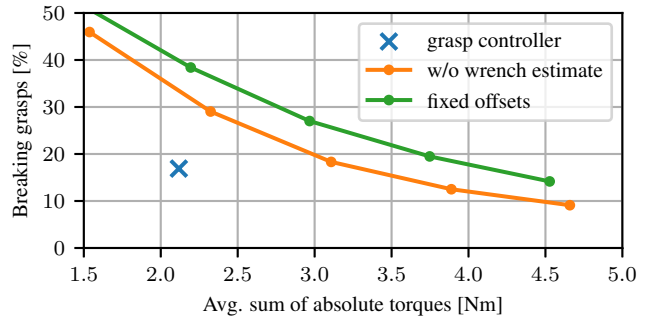


Fig. 7. Trade-off between efficiency and breaking grasps. Shows how the scaling factor in both baselines influences the ratio of breaking grasps compared to the required torques and compares it to our proposed controller. Along both axes, lower is better. Shows that our method is more efficient than both tested baselines.

called *fixed offsets*, which uses a fixed set of joint offsets  $\Delta q$  across all objects, specifically, 0.2 rad on the  $q_2$  and 0.1 rad on the  $q_3$  joints. This represents a common heuristic, where each finger is closed with the same force. In the other baseline, called *w/o wrench estimate*, we use the torque prediction network to predict for every given grasp a specific  $\Delta q = \mathcal{T}(o, q_{\text{init}}, q_{\text{grasp}}, w) - q_{\text{init}}$ , using  $w = 0$ . During the grasp,  $q_d$  is kept constant. For both heuristics, we compute  $q_d$  via  $q_d = \lambda \Delta q + q_{\text{init}}$  and use the same joint-level impedance controller as in the full grasp controller. To make a fair comparison, we try out different scaling factors  $\lambda \in \{1, 1.5, 2, 2.5, 3\}$  for both baselines.

### C. Efficiency

As shown in Fig. 7, when applying our controller on all grasps from the evaluation dataset, 16.9% of them break and on average, a torque sum of 2.1 N m is used across all objects. For comparison when using the *fixed offsets* baseline, the rate of broken grasps is ca. 40% when using a scaling factor that results in an equal torque consumption. If one would like to reach the same success rate, one would need to significantly increase the scale, however, this results in a higher torque usage of ca. 4.1 N m. The baseline *w/o wrench estimate* performs slightly better, as the joint offsets are specific to each object, however, it leads to the same conclusion: The baseline is way less efficient than our grasp controller when trying to reach the same success rate. At a scaling of 1, the *w/o wrench estimate* baseline equals our grasp controller, when one would keep  $q_d$  constant starting from the first timestep and ignore any changing external

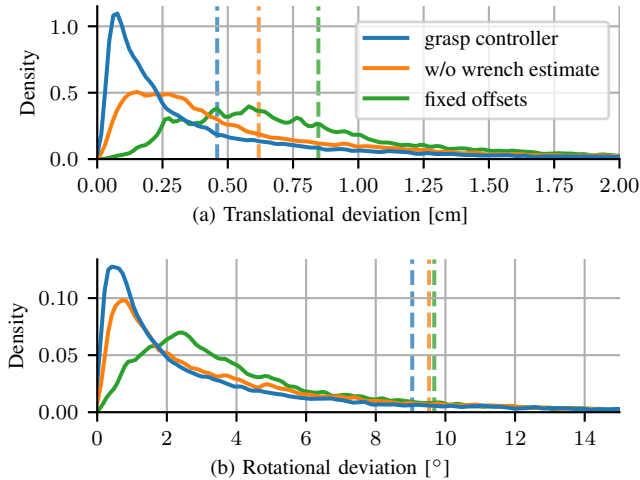


Fig. 8. Density diagrams showing the maximum object deviation of its initial pose when applying the respective method. Vertical dashed lines show the 90 % quantile of each method.

wrenches. In this case, this would lead to a failure rate of 46%. This shows that our controller is actively adapting to changing external wrenches, while still being efficient when no wrench is applied.

#### D. Object movement

Another advantage of our controller is that  $q_d$  is set in a way that the applied torques counter the external wrench when the object is at its initial pose. This means, that if the external wrench is estimated correctly, the object barely moves, even under high load. This is not the case when using one of the baselines. Here, the joint angles and therefore also the object pose need to deviate until the joint-level impedance controller applies the necessary torque. This deviation can be reduced by increasing the stiffness  $k_p$  of the controller, however, this also leads to more instability. Fig. 8 shows a density plot over the translational and rotational deviations of the evaluated objects. Our grasp controller keeps the deviation in 90 % of the cases below 4.6 mm and  $9.0^\circ$  which is in both cases smaller compared to the two baselines.

#### E. Minimal wrench estimation

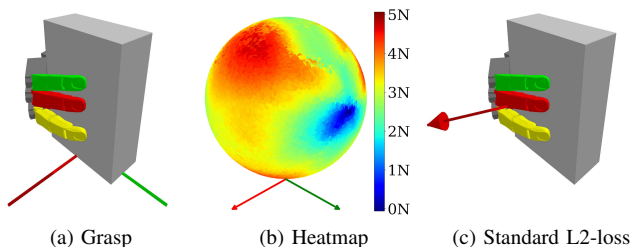


Fig. 9. Visualizes the effect of the minimal-wrench loss. For 3D external forces with  $\|w_f\| = 5\text{ N}$  applied to the object, we let the network estimate them based on the measured torques. The predicted wrench norm in Newton is visualized as a heatmap on a sphere. In Fig. 9c the hallucinated wrench predicted by a network trained with a standard L2 loss is shown.

In the first step of our controller, we estimate the external wrench being applied to the object. As explained, we decided

to always learn the minimal wrench explaining the given measurements. Figure Fig. 9 shows what the network trained using the minimal wrench loss predicts for a given grasp on a cereal box. Here, we sample random 3D forces with norm 5 N, we use  $q_d = \mathcal{T}(o, q_{\text{init}}, q_{\text{grasp}}, 0)$ , we solve for the resulting measured torques  $\tau$  using (2) to (7) and then let the network estimate the wrench  $w$  again based on  $\tau$ . In Fig. 9b we visualize the norm of the predicted forces  $\|w_f\|$  using a heatmap projected on a sphere. It is obvious that the predicted force gets toward zero when approaching forces parallel to the fingers along the y-axis. Due to the singularity, these wrenches do not result in any significant torques measured at the joints. Therefore, the network has learned to not predict any wrench at all along this axis. We further evaluate a network trained with a usual L2-loss: Fig. 9c shows the wrench that is estimated by such a network when no external wrench is actually applied. The network is hallucinating here an external force of size 2.8 N. In the same situation, our network trained with the minimal-wrench loss predicts an external force of norm 0.2 N.

#### F. Evaluation on the real robot

We evaluate the proposed grasp controller on our humanoid *Agile Justin* [23] using our grasping pipeline: First, the object is observed in the form of a single depth image using the robot’s Kinect camera. Based on the observed incomplete point cloud, we apply our shape completion and grasp prediction networks to find a stable grasp suitable for the given object [20]. After approaching the grasp pose using our learning-based motion planner [24], the fingers are closed until they get into contact with the object and then the hand lifts up the object. Now, the grasp controller is activated. Fig. 10 shows the estimated external wrench across various grasps. It can be seen that the estimated wrench fits the actual one in all cases. Also, Fig. 10a shows the predicted torques: For wrench A, the torques at  $q_1$  counter the external wrench, while the ones at  $q_2$  and  $q_3$  increase the normal forces such that the contacts do not break. For wrench B, the torques applied at the three fingers decrease and the ones applied at the thumb increase, while the torques at  $q_1$  stay constant. This shows that the predicted torques actively counter the estimated external wrench. In the experiments, the controller ran with a frequency of 40 Hz. However, also higher frequencies are possible, as one controller iteration takes 6 ms on a *NVIDIA V100*.

#### G. Discussion

Due to the physical limitations of the sensors used in the proposed controller, our approach has the following properties: Singularities in the finger kinematics can lead to the wrench not being measured at the torque sensors of the joint angles and therefore the wrench not being estimated correctly. Another reason for not estimating the wrench correctly are contact points with the fixed hand base, where contact forces are unknown. An example is shown in Fig. 5d. Furthermore, ambiguity also arises from the fact that we do not know the exact contact positions/normals and the

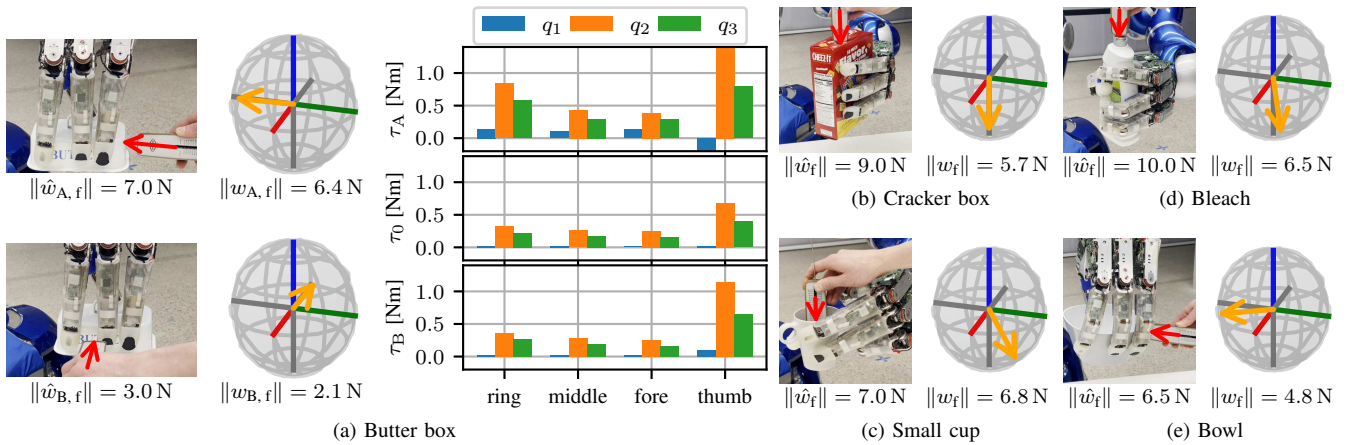


Fig. 10. Our grasp controller applied on the real robot across various objects. The applied forces are highlighted via red arrows and the normalized estimated wrenches are visualized as orange arrows next to each grasp. Fig. 10a shows additionally the predicted torques:  $\tau_0$  when no wrench is applied,  $\tau_A$  when the first shown wrench is applied and  $\tau_B$  when the second shown wrench is applied.

networks have to estimate them based on the observed and reconstructed object shape together with the joint positions.

## VI. CONCLUSIONS

In this work, we proposed a novel torque-based grasp controller that is able to robustly hold unknown objects. The controller only uses the measured joint torques and therefore does not require extensive object movement, which allows us to support power grasps with multi-contacts. We provided a full analytical formulation and proposed a learning-based adaptation, which decreases the cycle time to 6 ms and thus allows real-time performance. In a thorough evaluation in a physics-based simulator, we showed that the controller is more efficient and induces less involuntary object movement compared to the two tested baselines. We further showed the importance of the proposed minimal-wrench loss which makes sure the neural network does not make the controller unstable by hallucinating non-existent external wrenches. The controller has been shown to generalize to the real robot, where we showed that it is able to estimate applied wrenches and adapt the joint torques accordingly. Specifically, the controller was tested on our DLR-Hand II across various objects and grasp types. We leave it up for future work to reduce these physical limitations of our approach by incorporating more sensor modalities.

## REFERENCES

- [1] Q. Lu, M. van der Merwe, and T. Hermans, “Multi-Fingered Active Grasp Learning,” in *IEEE International Conference on Intelligent Robots and Systems*, 2020.
- [2] B. Wu *et al.*, “Generative Attention Learning: a “GenerAL” framework for high-performance multi-fingered grasping in clutter,” *Autonomous Robots*, vol. 44, no. 6, 2020.
- [3] U. R. Aktas *et al.*, “Deep dexterous grasping of novel objects from a single view,” *IEEE Transactions on Robotics*, 2019.
- [4] P. Mandikal and K. Grauman, “DexVIP: Learning Dexterous Grasping with Human Hand Pose Priors from Video,” in *International Conference on Robot Learning*, 2021.
- [5] D. Winkelbauer *et al.*, “A two-stage learning architecture that generates high-quality grasps for a multi-fingered hand,” in *IEEE International Conference on Intelligent Robots and Systems*, 2022.
- [6] S. A. Schneider and R. H. Cannon, “Object Impedance Control for Cooperative Manipulation: Theory and Experimental Results,” *IEEE Transactions on Robotics and Automation*, vol. 8, no. 3, 1992.
- [7] T. Wimboeck, C. Ott, and G. Hirzinger, “Passivity-based object-level impedance control for a multifingered hand,” in *IEEE International Conference on Intelligent Robots and Systems*, 2006.
- [8] T. Wimboeck, C. Ott, A. Albu-Schäffer, and G. Hirzinger, “Comparison of object-level grasp controllers for dynamic dexterous manipulation,” *International Journal of Robotics Research*, vol. 31, no. 1, 2012.
- [9] M. Pfanne *et al.*, “Object-Level Impedance Control for Dexterous In-Hand Manipulation,” *IEEE Robotics and Automation Letters*, vol. 5, no. 2, 2020.
- [10] D. Williams and O. Khatib, “Virtual linkage: A model for internal forces in multi-grasp manipulation,” in *IEEE International Conference on Robotics and Automation*, 1993.
- [11] Z. Su *et al.*, “Force estimation and slip detection/classification for grip control using a biomimetic tactile sensor,” in *IEEE International Conference on Humanoid Robots*, 2015.
- [12] N. Wettels *et al.*, “Grip control using biomimetic tactile sensing systems,” *IEEE Transactions on Mechatronics*, vol. 14, no. 6, 2009.
- [13] J. M. Romano *et al.*, “Human-inspired robotic grasp control with tactile sensing,” vol. 27, no. 6, 2011.
- [14] Z. Deng, Y. Jonetzko, L. Zhang, and J. Zhang, “Grasping force control of multi-fingered robotic hands through tactile sensing for object stabilization,” *Sensors (Switzerland)*, vol. 20, no. 4, 2020.
- [15] A. Bicchi, “Analysis and control of power grasping,” in *IEEE International Conference on Intelligent Robots and Systems*, 1991.
- [16] M. Haas-Heger and M. Ciocarlie, “Accurate Energetic Constraints for Passive Grasp Stability Analysis,” *IEEE Transactions on Robotics*, vol. 36, 2020.
- [17] M. Abdeetedal, H. Rezaee, H. A. Talebi, and F. Abdollahi, “Optimal adaptive Jacobian internal forces controller for multiple whole-limb manipulators in the presence of kinematic uncertainties,” *Mechatronics*, vol. 53, 2018.
- [18] J. Butterfaß, M. Grebenstein, H. Liu, and G. Hirzinger, “DLR-Hand II: Next generation of a dextrous robot hand,” in *IEEE International Conference on Robotics and Automation*, 2001.
- [19] D. Winkelbauer, B. Bäuml, and R. Triebel, “Learning-based real-time torque prediction for grasping unknown objects with a multi-fingered hand,” in *IEEE International Conference on Intelligent Robots and Systems*, 2023.
- [20] M. Humt, D. Winkelbauer, U. Hillenbrand, and B. Bäuml, “Combining shape completion and grasp prediction for fast and versatile grasping with a multi-fingered hand,” in *IEEE International Conference on Humanoid Robots*, 2023.
- [21] A. X. Chang *et al.*, “ShapeNet: An Information-Rich 3D Model Repository;” no. arXiv:1512.03012 [cs.GR], 2015.
- [22] N. S. Pollard, “Parallel methods for synthesizing whole-hand grasps from generalized prototypes,” Ph.D. dissertation, 1994.
- [23] B. Bäuml *et al.*, “Agile justin: An upgraded member of dlr’s family of lightweight and torque controlled humanoids,” in *IEEE International Conference on Robotics and Automation*, 2014.
- [24] J. Tenhumberg, D. Burschka, and B. Bäuml, “Speeding up optimization-based motion planning through deep learning,” in *IEEE International Conference on Intelligent Robots and Systems*, 2022.