Multi-fingered Tactile Servoing for Grasping Adjustment under Partial Observation

Hanzhong Liu[1*], Bidan Huang[2†], Qiang Li[3], Yu Zheng[2], Yonggen Ling[2], Wangwei Lee[2], Yi Liu[4*], Ya-Yen Tsai[5*], and Chenguang Yang[6]

***Abstract*— Grasping of objects using multi-fingered robotic hands often fails due to small uncertainties in the hand motion control and the object's pose estimation. To tackle this problem, we propose a grasping adjustment strategy based on tactile seroving. Our technique employs feedback from a sensorized multi-fingered robotic hand to collaboratively servo the fingers and palm to achieve the desired grasp. We demonstrate the performance of our method through simulation and physical experiments by having a robot grasp different objects under conditions of variable uncertainty. The results show that our approach achieved a higher success rate and tolerated greater uncertainty than an open-looped grasp.**

I. Introduction

Over the past decades, researches on robotic grasping have increased rapidly [1], [2]. A variety of robotic hands have been developed for the purpose of object grasping. To achieve human-like dexterity, many of the recent developed hands have multiple fingers or high degrees of freedom and are equipped with tactile sensors [3]–[5]. Despite the extensive studies on robotic hand design and grasping algorithms, dexterous robotic grasping for everyday objects remains an open challenge due to the complexity of the contact dynamics and the large variety of grasping objects.

Typically, an autonomous grasp can be divided into two stages: grasp planning and grasp execution [1]. The former generates a stable grasp from a given scene, e.g. the pose and the model of the targeting object or an image of the environment, and the latter tackles how to control the robotic hand to achieve the planned grasp. Many of the grasp planning algorithms rely on accurate models of the objects to compute desired grasps, e.g. form-closure/force-closure grasps [6]–[10]. The generated desired grasps are stable in theory and require accurate object poses and precise control of the robotic hand. However, in reality, uncertainties and errors from pose estimation and control are inevitable. As a result, the executed grasp always deviates from the desired one, leading to grasping failure. To tackle these uncertainties and errors, we develop an algorithm to guide the robotic hand and fingers moving towards the desired grasp using tactile sensing.

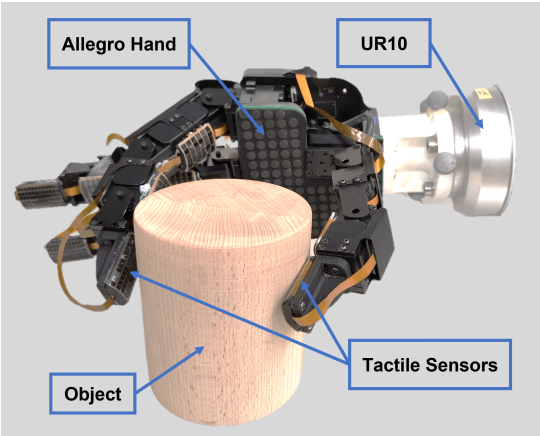

Fig. 1. Overview of the experiment platform.

Different from many of the previous robotic grasping studies that try to tackle the uncertainty at the planning stage, this work focuses on the grasping execution stage. We utilise tactile sensors to capture and compensate for the uncertainty. The proposed algorithm is based on the following presumptions:

- The fingertips of a multi-fingered hand are covered by tactile sensors that provide the positions of contacts.
- A grasp planner is available to provide a feasible grasp for a given object and the robotic hand.
- Before the start of the grasping adjustment, the object pose has been roughly estimated and the hand has been moved to the location given by grasp planning, with all the fingers closed to make contact with the object. The contact locations on the object and fingertips are in the neighbourhood of the planned ones.
- The accurate object pose is not observable by external sensors (e.g. cameras) due to occlusion after the closing of fingers, neither is the relative pose between the object and the hand. The grasping state is hence partially observable (Fig. 2).

These presumptions are reasonable because 1) many robotic hands have been equipped with tactile sensors [11] and 2) a range of grasp planning algorithms and object pose estimation algorithms have been developed for known objects [12]–[14]. In most cases, the object pose estimation is obtained by robotic vision and yet hand-eye calibration, which transforms the object pose to the robotic hand frame, can be difficult to obtain with high accuracy. Therefore,

† denotes the corresponding author.
* This work was done during their internship at Tencent Robotics X.
[1]H. Liu is with South China University of Technology, China.
[2]B. Huang, Y. Zheng, Y. Ling, W. Lee are with Tencent Robotics X, Shenzhen, Guangdong Province, China (email: bidanhuang@tencent.com).
[3]Q. Li is with Neuroinformatics Group, Center for Cognitive Interaction Technology (CITEC), Bielefeld University, 33619, Bielefeld, Germany. He would like to acknowledge the support by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Projektnummer (No.410916101).
[4]Y. Liu is with Guangdong University of Technology, China.
[5]Y.-Y. Tsai is with Imperial College London, UK.
[6]C. Yang is with Bristol Robotics Laboratory, the University of the West of England, UK.

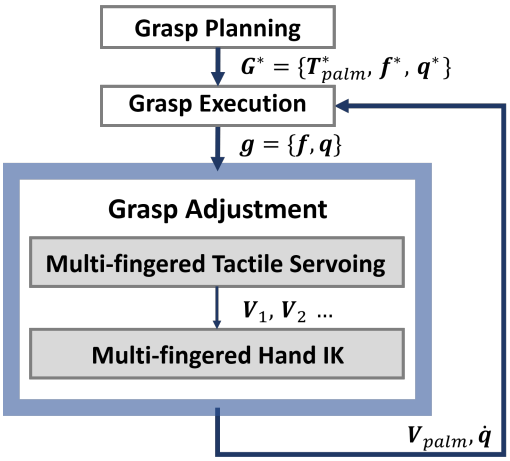

Fig. 2. Block diagram of the proposed grasping control framework.

in the third presumption, the contacts are highly likely to deviate from the desired locations which can lead to failure in a open-loop grasping. In practise, this deviation always exists and is hard to be completely eliminated. Besides, vision based algorithms may be subject to occlusion when the object is in hand. In the proposed algorithm, the deviation from the desired grasp is evaluated by the tactile feedback and compensated by the robotic hand and finger movements without re-estimating the object pose.

The proposed method is inspired by our previous tactile servoing work [15], which was originally evaluated in the context of tactile exploration with a single arm. In this paper, we mainly investigate how tactile servoing can be extended to guide multiple fingers work cooperatively and adjust a grasp online. To this end, the workflow is split into two phases. The first phase is aimed at selecting the best tactile Jacobian matrix that transforms the tactile features into fingertip velocities. In the second phase, an extended inverse kinematics (IK) algorithm is used to compute the palm twist and finger joint velocities.

The main contributions of our work are as follows:

1) We have proposed a novel grasping adjustment strategy, which selects a suitable tactile Jacobian matrix and computes the movement of each fingertip to reach its desired location.
2) We have implemented an inverse kinematics (IK) solution for a multi-fingered hand to make all the fingers and the palm move cooperatively to achieve the desired fingertips' movements. By projecting finger joint velocities to the null-space of tactile servoing solution, the grasping posture tracking is improved, and thus the success rate of the grasp.

II. Related Work

A. Grasping

Robotic grasping of an object can be achieved within two steps, grasp planning and grasp execution (illustrated in Fig. 2). In grasp planning, a stable grasp, defined by the desired grasp posture and hand pose relative to the targeting object, is computed. In grasp execution, a control strategy is then adopted to move the robotic hand and fingers to the planned grasping poses. In reality, robotic grasping often faces uncertainties, which can come from inaccurate model used during grasp planning, pose estimation and control error during grasp execution [16]–[19], and these are leading factors that result in grasp failure.

Prior works have attempted to tackle such uncertainties during grasp planning [8]. In [20], the authors take shape uncertainty as constraints in grasp planning and develop a compliant finger closing scheme. However, the uncertainties may still present in the downstream grasp execution phase. Other works focus on addressing the uncertainties in execution stage. [21] proposes a compliant approach-to-grasp strategy for multi-fingered hand and tackles the uncertainties during the fingers approaching stage using a joint-torque based impedance controller. More recently, an end-to-end approach has also been proposed to address the uncertainties in planning and execution [22]–[24].

In contrast to the aforementioned works, grasp adjustment focuses on how to correct the unstable grasp after the grasping execution and achieve a stable grasp [25], [26]. For example, [27] proposes to estimate the grasp stability after an initial grasp is established. When an unstable grasp is encountered, a hand adjustment that leverages tactile feedback is carried out to correct the grasping pose. Li et al. [28] propose a grasp adaptation method to tackle the unstable grasp after execution. By comparing the tactile feedback to the grasping model learned from demonstration, either an object-level impedance adaptation or a local exploration corrective action is taken. This process is iterated until a stable grasp is estimated. Reinforcement learning (RL) approaches have also been studied to adjust the initial grasp [29]. Taking proprioceptive and tactile feedback observations, the RL policy determines whether a grasp is stable. In the case of an unstable grasp, the policy reopens the gripper and adjusts the pose of the end effector before attempting another grasp.

B. Tactile Servoing

Tactile servoing is a robot control strategy which computes the desired robot motion such that it minimises the differences between the desired and observed tactile features. Unlike re-grasping strategy, tactile servoing makes continuous adjustment between fingers and the object without breaking contacts. In the context of tactile servoing, tactile features describe the state of each contact [15]. It can come in different representations, but the commonly used ones include contact position, pressure and contact edge. Various tactile servoing algorithms were evaluated on different types of sensors [15], [30], [31], and they can be broadly divided into two categories depending on the feedback required in the controller: image-based and pose-based [30]. Image-based approaches use the features computed from the tactile image. Zhang [32] proposed a strategy to control the contact pattern and then was lately improved by [15]. In the framework, a task-independent Jacobian matrix and a task-dependent projection matrix were used to implement the mapping function. [31] extended this control framework from four dimensions [15] to six by including the center of pressure. Image-based approaches are widely used for piezoresistive tactile arrays because the array type tactile feedback is

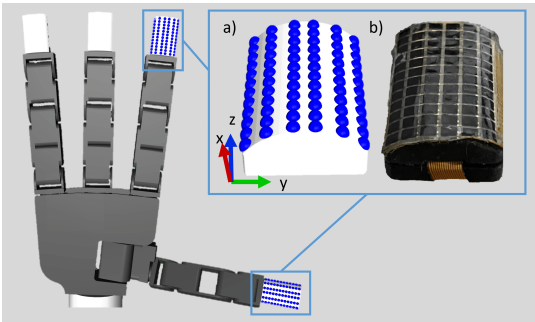

Fig. 3. The Allegro Hand integrated with tactile sensors. Right: a) A simulation model for the fingertip. Each blue dot represents a taxel. b) A real fingertip. Each cell is a taxel.

intuitively analogous to camera image output. The tactile features can be easily computed by image-based method. However, it is not the case for many optical tactile sensors such as Tactip [30]. To this end, Lepora et al. [30] proposed the pose-based tactile servoing controller. In their controller, the tactile feature-contact edge is represented by a vector in Cartesian space, which is learned from a collected image dataset observed by the camera of Tactip. All of these works study tactile servoing using single end-effector.

Despite the fact that human naturally use tactile servoing to adjust their grasps in daily life, e.g. sliding and rotating fingers along the object surface until reach a stable grasp, this strategy has not been fully explored in robotic grasping. In this paper, we propose a strategy for continuous fingertip adjustment and guiding the hand towards the desired configuration. To the best of our knowledge, this is the first study that considers motions of multi-fingered hand in tactile servoing for the purpose of grasping adjustment.

The rest of this paper is structured as follows. Section III explains the details of our method. Section IV provides the experimental results. Section V concludes the paper and discusses possible future work.

III. Methodology

A. Overview

Our grasping control framework is illustrated in Fig. 2. Considering the fingertip-based grasp, we define the desired grasping feature as $\boldsymbol{G}^* = \{\boldsymbol{T}^*_{palm}, \boldsymbol{q}^*, \boldsymbol{f}^*\}$, the desired palm pose $\boldsymbol{T}^*_{palm}$, desired joint angles for each finger $\boldsymbol{q}^*$, and desired tactile features $\boldsymbol{f}^*$ for each fingertip. After the object pose is estimated, the robotic hand approaches the object to the desired palm pose $\boldsymbol{T}^*_{palm}$ and closes its fingers until contacts are detected by the tactile sensors. This forms a current grasp $\boldsymbol{G} = \{\boldsymbol{T}_{palm}, \boldsymbol{q}, \boldsymbol{f}\}$. For the existence of the small noise/error in the object pose estimation or grasping control, $\boldsymbol{G}$ is in the neighbourhood of $\boldsymbol{G}^*$. The grasping adjustment starts after all the fingertips make contact with the object. This algorithm is aimed at minimizing the difference between $\boldsymbol{G}^*$ and $\boldsymbol{G}$ via tactile servoing.

Theoretically, the grasping adjustment of $\boldsymbol{G}$ can be achieved by re-estimating the object pose and then re-grasping the object with $\boldsymbol{G}^*$. In practice, however, after finger closing, the object is occluded by the hand and vision-based pose estimation is intractable. As a result, the current palm pose in the object frame $\boldsymbol{T}_{palm}$ is infeasible, leading to a partially observable current grasp $\boldsymbol{g} = \{\boldsymbol{q}, \boldsymbol{f}\}$, with the current joint state $\boldsymbol{q}$ and the current tactile features $\boldsymbol{f}$. In our proposed methods, we tried to approach the $\boldsymbol{g}^* = \{\boldsymbol{q}^*, \boldsymbol{f}^*\}$ by observing $\boldsymbol{g}$ and making the fingertips move along the object surface, i.e. sliding and rotating. These finger movements are commonly used in human grasping for adjustment but have not been well studied in robot grasping. Different from the popular re-grasping strategy, our adjustment strategy does not require re-opening and re-closing the fingers. It has the advantage of not breaking the contacts between the fingers and the object, which can result in unexpected changes of object states, and hence it reduces the uncertainties.

Algorithm 1 Multi-fingered tactile servoing

Input:
Deviation of tactile features $\boldsymbol{e}_{index} = \boldsymbol{f}^*_{index} - \boldsymbol{f}_{index}$, $\boldsymbol{e}_{thumb} = \boldsymbol{f}^*_{thumb} - \boldsymbol{f}_{thumb}$
Desired twist $\boldsymbol{V}^*_{index}$ by the desired relative-to-thumb pose
Desired finger joint velocities $\dot{\boldsymbol{q}}^* \leftarrow \boldsymbol{q}^* - \boldsymbol{q}$
Output:
Twists of index finger and thumb for IK phase: $\boldsymbol{V}_{index}$ and $\boldsymbol{V}_{thumb}$

for $i \leftarrow 1$ to 4 **do**
  $\boldsymbol{J}_{s_{index}} = \boldsymbol{J}_{s_i}$
  $\tilde{\boldsymbol{V}}_{index_i} = \boldsymbol{J}^{\dagger}_{s_{index}} \boldsymbol{e}_{index}$ ▷ † is for pseudo-inverse.
end for
$\boldsymbol{V}_{index} = \text{argmin}(\text{dist}(\tilde{\boldsymbol{V}}_{index_i}, \boldsymbol{V}^*_{index}))$
for $i \leftarrow 1$ to 4 **do**
  $\boldsymbol{J}_{s_{thumb}} = \boldsymbol{J}_{s_i}$
  $\tilde{\boldsymbol{V}}_{thumb_i} = \boldsymbol{J}^{\dagger}_{s_{thumb}} \boldsymbol{e}_{thumb}$
  $\tilde{\boldsymbol{q}}_i = \text{IK}(\boldsymbol{V}_{index}, \tilde{\boldsymbol{V}}_{thumb_i})$ ▷ IK is for inverse kinematics of velocity.
end for
$\tilde{\boldsymbol{q}} = \text{argmin}(\text{dist}(\tilde{\boldsymbol{q}}_i, \dot{\boldsymbol{q}}^*))$
$\boldsymbol{V}_{thumb} = \text{FK}(\tilde{\boldsymbol{q}})$ ▷ FK is for forward kinematics.

The grasping adjustment is a real-time algorithm. The inputs at each time step are the desired grasp $\boldsymbol{G}^*$ and the partially observed current grasp $\boldsymbol{g}$, while the outputs are the palm twist $\boldsymbol{V}_{palm}$ and the finger joint velocities $\dot{\boldsymbol{q}}$.

The algorithm has two phases: tactile servoing and inverse kinematics (IK). In the first phase, the fingertip twists are computed according to the deviation of the tactile features, with the consideration of the deviation of the finger joint angles and the relative poses between fingers (Sec. III-B). In the second phase, the IK solver takes the fingertip twists and computes the palm twist and finger joint velocities with the hand Jacobian (Sec. III-C). This approach works iteratively until the grasping error, namely the difference between $\boldsymbol{g}^*$ and $\boldsymbol{g}$, converges.

In the following discussion, we will detail the algorithm with an example of two-fingered precision grasp (thumb and index finger). Nevertheless, it is not limited to two fingers and can be easily extended to more fingers.

Algorithm 2 Modified hand Jacobian matrix computation

Input:
Hand Jacobian from Eq. (11) $\boldsymbol{J}_k \in \mathbb{R}^{6n\times(6+mn)}$
▷ where n is the number of fingers and m is the DoF of each finger.
Output: Censored hand Jacobian $\boldsymbol{J}_k^* \in \mathbb{R}^{5n\times(6+mn)}$

```
Initialize the censored Jacobian J_k^* ← []
for i ← 0 to n − 1 do
    for j ← 1,2,3,5,6 do
        J_k^* ← [J_k^*; J_{6i+j}]   ▷ J_{6i+j} is the (6i+j)^th row of J_k
    end for
end for
```

B. Tactile servoing

1) Tactile Sensor: The tactile sensors used in the proposed tactile servoing method is a resistive tactile sensing array developed in our lab, which has gone through multiple iterations since the original development. As shown in Fig. 3, there are 6×12 taxels fixed on a plane of $20\times 30mm$, bent into a curved surface to fit the shape of the fingertip. Each taxel, i.e. each sensor unit shown in Fig. 3, responses in voltage proportional to the pressure applied on it. The voltage is then transferred to tactile signal and published at a frequency of 100 Hz. Four tactile sensors are mounted on the four fingertips of the Allegro Hand.

2) Tactile Feature Specification: In the proposed algorithm, the tactile feature is defined in 3 dimensions as

$$\boldsymbol{f} = \begin{bmatrix} f_x & f_y & f_z \end{bmatrix}^T \tag{1}$$

where $(*)^T$ means the transpose of a matrix or a vector, and the tactile features are detailed as follows:

$$f_x = \sum f_{ni}, \quad f_y = \frac{\sum f_{ni} p_{yi}}{\sum f_{ni}}, \quad f_z = \frac{\sum f_{ni} p_{zi}}{\sum f_{ni}} \tag{2}$$

where f_{ni} represents the tactile value of the ith taxel, and p_{yi} and p_{zi} are the projecting positions of the taxel in y and z axes, respectively, with respect to the tip frame (See. Fig. 3).

In the tactile feature $\boldsymbol{f}$, f_x represents the normal contact force, and f_y, f_z represent the position of the center of the contact area. The goal of tactile servoing for each fingertip is to maintain the desired tactile features in $\boldsymbol{g}^*$, i.e. a suitable normal force f_x^* and a desired location (f_y^*, f_z^*) for the center of the contact area.

3) Tactile Feature Jacobian: In the earlier work [15], a "tactile Jacobian" for one kinematic chain was defined and here we extended it to multiple chains. The inverse of the tactile Jacobian turns the deviation of the tactile feature to the twist of the fingertip as

$$\begin{aligned} \dot{\boldsymbol{f}} &= \boldsymbol{J}_s \boldsymbol{V} \\ \boldsymbol{V} &= \boldsymbol{J}_s^{\dagger} \dot{\boldsymbol{f}} \end{aligned} \tag{3}$$

where $\boldsymbol{V} = \begin{bmatrix} v_x & v_y & v_z & \omega_y & \omega_z \end{bmatrix}^T$ is the twist of the fingertip, v represents the linear velocity, ω represents the angular velocity, and $(*)^{\dagger}$ is the pseudo-inverse of a matrix. Here the ω_x is neglected as the rotation along the normal of the contact area is not in consideration.

In this work, $\boldsymbol{J}_s$ is defined as

$$\boldsymbol{J}_s = \begin{bmatrix} \frac{\partial f_x}{\partial p_x} & 0 & 0 & 0 & 0 \\ 0 & \frac{\partial f_y}{\partial p_y} & 0 & 0 & \frac{\partial f_y}{\partial \theta_z} \\ 0 & 0 & \frac{\partial f_z}{\partial p_z} & \frac{\partial f_z}{\partial \theta_y} & 0 \end{bmatrix} \tag{4}$$

where p_x, p_y and p_z represent the projecting displacements of the fingertip in the x, y and z axis of the tip frame, respectively, and θ_y and θ_z are the rotations about the corresponding axes. The five non-zero elements are the key of this algorithm and they quantify the correlation between the fingertip movement and the deviation of the tactile feature with constant values. In this definition, tactile servoing is used to maintain a contact between the fingertip and the object and meanwhile allow the fingertip to explore the object surface with sliding or rotating movement. The five non-zero elements have the following meanings:

1) $\frac{\partial f_x}{\partial p_x}$: tactile feature deviation in pressure corresponds to fingertip movement along its surface normal. This term implies that Hooke's law of a spring is used here. This term keeps the fingertip pressing on the object.
2) $\frac{\partial f_y}{\partial p_y}$ and $\frac{\partial f_y}{\partial \theta_z}$: tactile feature deviations in the y direction correspond to sliding in the y axis and rotation about the z axis.
3) $\frac{\partial f_z}{\partial p_z}$ and $\frac{\partial f_z}{\partial \theta_y}$: tactile feature deviations in the z direction correspond to sliding in the z axis and rotation about the y axis.

The zero elements mean that the tactile feature deviation does not lead to any movement in the corresponding dimensions.

According to $\boldsymbol{J}_s$, there are two different actions to take based on the tactile feature deviation in the y and z directions: sliding or rotation. In this work, we separate the two actions and the fingertip can be either sliding or rotating at each time step. This hence requires the fingertip to take the best action. To this end, we propose a selecting strategy to choose a tactile feature Jacobian $\boldsymbol{J}_s$ during tactile servoing at each time step that represents the best action. The selecting strategy will be explained in the next part and the selecting matrix is defined as

$$\boldsymbol{A}_s = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 & 0 \\ 0 & 0 & s_z & 0 & 0 \\ 0 & 0 & 0 & r_y & 0 \\ 0 & 0 & 0 & 0 & r_z \end{bmatrix} \tag{5}$$

The separation of the sliding and rotation makes four different choices of A_s in total as

$$\begin{cases} A_{s1} : s_y = 0 \quad r_z = 1 \quad s_z = 0 \quad r_y = 1 \\ A_{s2} : s_y = 1 \quad r_z = 0 \quad s_z = 0 \quad r_y = 1 \\ A_{s3} : s_y = 0 \quad r_z = 1 \quad s_z = 1 \quad r_y = 0 \\ A_{s4} : s_y = 1 \quad r_z = 0 \quad s_z = 1 \quad r_y = 0 \end{cases} \tag{6}$$

Hence there are four choice of $\boldsymbol{J}_s$, representing four different actions responding to the tactile feature deviations:

$$\boldsymbol{J}_{si} = \boldsymbol{J}_s \boldsymbol{A}_{si} \tag{7}$$

where $\boldsymbol{J}_s$ refers to the Jacobian in (4) and $\boldsymbol{J}_{si}$ is the selected tactile Jacobian at a certain time step. It should be noted that all the $\boldsymbol{J}_s$ below refers to the selected one on the left side of (7).

According to (4)–(7), there must be one and only one nonzero entry in each row of $\boldsymbol{J}_s$, which means that only one kind of action, i.e. sliding or rotation, is provided to modify the corresponding tactile feature. All the fingers need to make decision on the value of $\boldsymbol{A}_s$. In the next part, we take a two-fingered grasping case as an example to explain the selection criteria, i.e. how to make the decision of the $\boldsymbol{A}_s$ for the thumb ($\boldsymbol{A}_s^{thumb}$) and the index finger ($\boldsymbol{A}_s^{index}$). The selecting over a period of time leads to an adaptive behavior.

4) Tactile Jacobian Selection: In human grasping, the thumb plays a special role and is the most independent one [33]. Therefore, in our multi-fingered tactile servoing strategy, we make the thumb the leader and the other fingers the followers. In this example, the index finger first computes its relative-to-thumb pose and iterates through the four different selecting matrix $\boldsymbol{A}_s$. It makes its decision of $\boldsymbol{A}_s^{index}$ according to the desired relative-to-thumb pose computed from the desired grasp and computes its twist command $\boldsymbol{V}_{index}$ (Fig. 4). The thumb then iterates through the $\boldsymbol{A}_s$ and generates four different twists. The four different thumb twists, along with the index finger twist command $\boldsymbol{V}_{index}$, are then sent to the IK (Sec.III-C) to calculate the finger joint velocities and palm twist. Among the four solutions, the one that takes the highest velocity to the desired joint angles is selected, and the thumb twist command $\boldsymbol{V}_{thumb}$ is generated with this thumb twist (Algorithm 1). The IK solution of the $\boldsymbol{V}_{thumb}$ and $\boldsymbol{V}_{index}$, which is already calculated during the selection procedure, is used as the robot command.

Figure 4 explains further about the selecting strategy of the index finger. The desired index pose related to thumb ${}^{thumb}\boldsymbol{T}^*_{index}$ can be obtained from the desired grasp. The current relative pose between the two fingers is expressed as ${}^{thumb}\boldsymbol{T}_{index}$. This hence allows us to compute a desired twist for the index finger $\boldsymbol{V}^*_{index}$. However, the index finger can not move straightly to its desired pose as it has to move on the surface of the object. Hence a tactile Jacobian $\boldsymbol{J}_{si}$ is chosen to generate a twist for the finger, which allows the fastest approach to the desired pose while keeping the finger touching the object surface.

C. Multi-fingered Hand IK

As mentioned above, tactile feature Jacobian $\boldsymbol{J}_s$ describes the twist-tactile feature system and the fingertip twists command $\boldsymbol{V}$ can be calculated according to deviations of the tactile features $\boldsymbol{e_f}$ and the relative-to-thumb poses. To achieve the commanded twists of the fingertips, another Jocobian $\boldsymbol{J}_k$ is adopted to describe the kinematics of the multi-fingered hand system. Since the robotic hand is mounted on a robot end-effector, the controlled variable can be chosen as

$$\boldsymbol{U} = \begin{bmatrix} \boldsymbol{V}_{palm}^T & \dot{\boldsymbol{q}}_1^T & \dot{\boldsymbol{q}}_2^T & \cdots & \dot{\boldsymbol{q}}_i^T \end{bmatrix}^T \tag{8}$$

where $\boldsymbol{V}_{palm}$ is the palm twist, relative to and expressed in the static world frame. $\dot{\boldsymbol{q}}_i$ is the joint angles velocities of the ith finger. The input of the multi-fingered IK is the commanded twists of all the fingertips relative to the ground. To match the coordinates in tactile servoing, we express the twists in the frame of the tactile feature of each fingertip, i.e., each fingertip's frame. Here, we stack these commanded twists into an augmented twist as

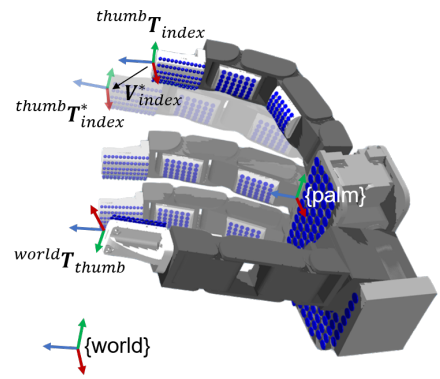

Fig. 4. Thumb guided desired twist generated with the relative-to-thumb pose.

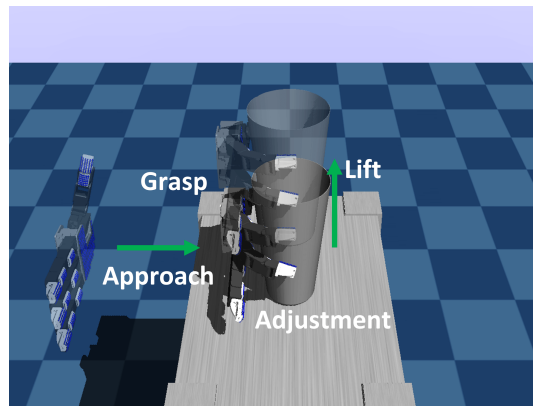

Fig. 5. Process of experiments in simulation.

$$\boldsymbol{V}_{aug} = \begin{bmatrix} \boldsymbol{V}_1^T & \boldsymbol{V}_2^T & \cdots & \boldsymbol{V}_i^T \end{bmatrix}^T \tag{9}$$

in which $\boldsymbol{V}_i$ is the commanded twist of the ith fingertip, in its own frame, and thus we have

$$\boldsymbol{V}_{aug} = \boldsymbol{J}_k \boldsymbol{U} \tag{10}$$

where $\boldsymbol{J}_k$ is written as

$$\boldsymbol{J}_k = \begin{bmatrix} \boldsymbol{J}_{palm_1} & \boldsymbol{J}_1 & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{J}_{palm_2} & \boldsymbol{0} & \boldsymbol{J}_2 & \cdots & \boldsymbol{0} \\ \vdots & \vdots & & \ddots & \\ \boldsymbol{J}_{palm_i} & \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{J}_i \end{bmatrix} \tag{11}$$

where $\boldsymbol{J}_{palm_i}$ describes the contribution from palm twist to the twist of the ith fingertip and it is defined as

$$\boldsymbol{J}_{palm_i} = \begin{bmatrix} \boldsymbol{I}_{3\times3} & -[{}^{palm}\boldsymbol{p}_i]_\times \\ \boldsymbol{0} & \boldsymbol{I}_{3\times3} \end{bmatrix} \tag{12}$$

where ${}^{palm}\boldsymbol{p}_i$ represents the position of the ith fingertip in the palm frame and $[{}^{palm}\boldsymbol{p}_i]_\times$ is its corresponding skew-symmetric matrix. By simply applying coordinate transformation on both sides of $\boldsymbol{J}_{palm_i}$, the definition of coordinates

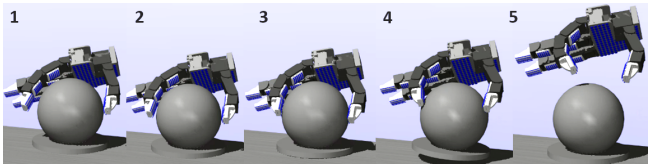

Fig. 6. Grasping a sphere without adjustment.

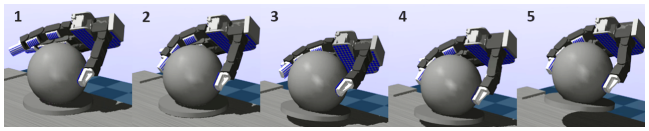

Fig. 7. Grasping a sphere with the proposed grasping adjustment framework. 1 - 2: the fingers are closed up until tactile sensors detect contacts. 3 - 4: the hand posture, the palm pose and the contact areas are adjusted by tactile servoing. 5: The object is lifted.

above is satisfied, and we have

$$\boldsymbol{V}_i = \boldsymbol{J}_{palm_i} \boldsymbol{V}_{palm} \tag{13}$$

when the finger joints stay still.

The finger Jacobian $\boldsymbol{J}_i$ can be calculated with current joints state, which satisfies

$$\boldsymbol{V}_i = \boldsymbol{J}_i \dot{\boldsymbol{q}}_i \tag{14}$$

when the palm keeps still.

As mentioned in Section III-B, the input of IK, i.e., the augmented commanded twist in (9) has a dimension of $6+5n$, where n is the number of fingers involved in the task and some dimensions are left free by the tactile servoing. Thus, the hand Jacobian in (11) needs to be modified correspondingly, which is shown in Algorithm 2.

According to the above definition, we obtain a suitable hand Jacobian $\boldsymbol{J}_k$ for such a multi-fingered robotic hand. Now we can formulate our IK method:

$$\boldsymbol{U} = \boldsymbol{J}_k^{\dagger} \boldsymbol{V}_{aug} + (\boldsymbol{I} - \boldsymbol{J}_k^{\dagger} \boldsymbol{J}_k) \boldsymbol{U}^* \tag{15}$$

where $\boldsymbol{U}^*$ can be any control input. Because this input will be projected into the null-space of the main task (minimizing the tactile feature deviation), it will not affect the implementation of the main task but leave the controllable freedom to move the robotic hand. Here we define the second task as to maintain the palm pose at the beginning of the task and move the hand posture towards the desired grasping posture, whose expected gradient corresponding to the palm and finger motion is simplified as the distance between the initialized palm pose or the desired hand posture to the current ones.

IV. Experiments

The performance of the proposed grasping adjustment framework was validated on a grasping task with different objects, in both simulation and reality. As shown in Fig. 1, the experiment was carried out on a UR10-Allegro arm-hand robotic platform. The Allegro hand was covered with in-house customized tactile sensors [34].

A. Simulation Grasp

In the simulation, we tested whether the proposed grasping adjustment could be generalized to objects with different shapes. The performance of the proposed method was compared to the baseline, i.e. a standard grasping approach without grasping adjustment.

We used MuJoCo, which offers fast, accurate and convenient modelling and simulation for our grasping experiments. The process of our experiments and the simulation platform are shown in Fig. 5. In the simulation environment, the object for grasping test was placed on the table. The control of finger joint angles was achieved with the PD controller provided by MuJoCo, and the "mocap" feature in MuJoCo was used to directly control the position and orientation of the palm. In our setup, we assumed we have obtained a planned or desired grasp, which contains the desired tactile features, hand pose and hand posture for an object. At task execution, the object pose was given, but with error.

A baseline followed the standard grasp execution approach, which directly moved the palm towards the given object pose and performed the grasping action. When the initial grasp was established, the fingers barely made contact with the object, and hence the grasp was unstable. The proposed method took the same approach to first reach the object, but then took the grasping adjustment approach as described in Sec. III, to gradually correct itself towards a stable grasp using tactile sensing as feedback. Both the baseline and the proposed method would attempt a pick up action once they finished their attempt to grasp. The evaluation of the grasp performance was done on 4 objects of different shapes, as listed in Table I and the comparison was made to the baseline. For the cube, cup and cylinder, we randomly introduced a pose estimation error between -1.5cm to 1.5cm and the result was calculated based on 16 tests. We have performed a further experiment on the sphere, with a pose estimation error up to 3 cm and totally 546 tests evenly distributed in this range. The grasping result is illustrated in Fig. 10, where the unspecified dimensions of the pose error were randomly generated but remained consistent between the proposed method and the baseline in each control experiment. The grasp was considered successful, if the robot could pick up the object at the end of the grasp without falling. As shown in the figure and the table, the proposed grasping adjustment approach performed better than the baseline for the cylinder, cube and sphere and on par with the baseline for the cup.

In Figs. 6 and 7, we present and compare the grasping process of a spherical object involving pose estimation error. Figure 6 shows the grasping process of the baseline. Without grasping adjustment, the Allegro Hand picked up the object directly from an unstable grasp, and hence failed the task. For comparison, the proposed approach shown in Fig. 7 gave another trial. The index finger and thumb started off closing up until contacts were detected by the tactile sensors. Then, the index finger and thumb gradually followed the surface of the sphere and adapted the grasp. Finally, the object was successfully picked up. Figures 8 and 9 show the corresponding tracking of the tactile features and the hand posture during grasping control. At the end of the grasping

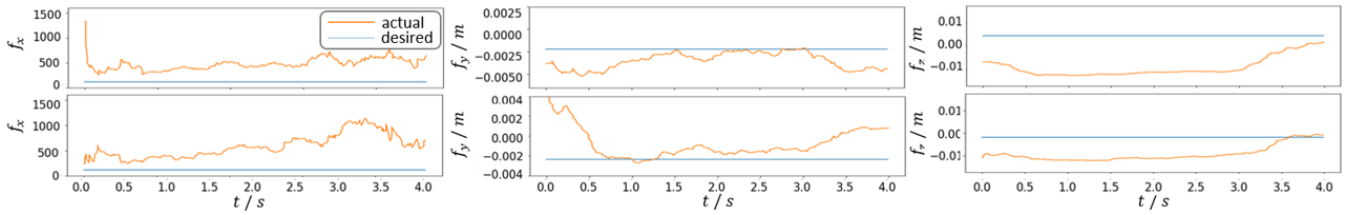

Fig. 8. Tactile features of the index (top) and thumb (bottom) fingertips during grasping control.

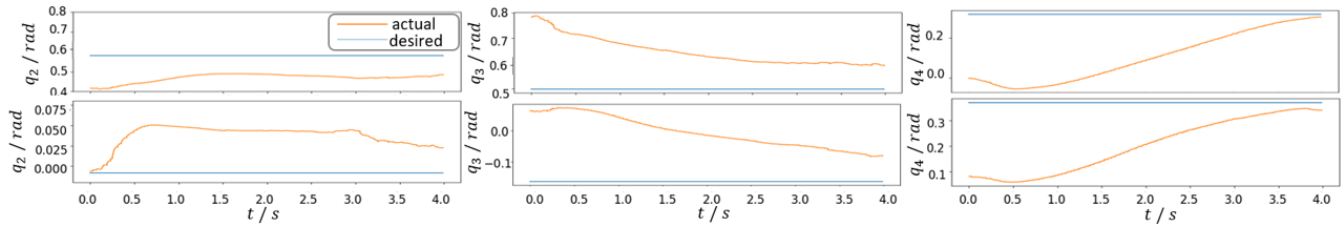

Fig. 9. Joint angles of the index finger (top) and thumb (bottom) during grasping control. The first joint of each finger, which is closest to the palm, would probably achieve the joint limits during grasping control, therefore only three joints of each finger are utilized in the experiments.

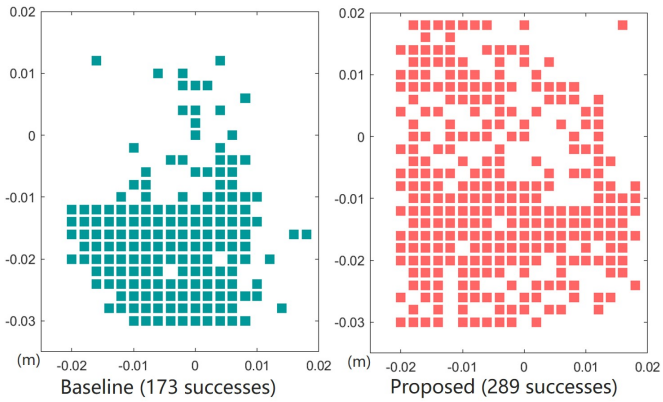

Fig. 10. Grasping a spherical object at 546 different locations around the desired grasp: $(0,0)$ is the location of the desired grasp and each green (baseline) and red (the proposed method) box represents a successful grasp.

TABLE I. Grasping success rate for different objects.

Object name	Cube	Cup	Cylinder	Sphere
Proposed	87.5%	75.0%	87.5%	52.93%
Baseline	50.0%	75.0%	68.75%	31.69%

control, tactile features f_y and f_z are quite close to the desired values. A greater tracking error in f_x is likely caused by the rigid contact, but does not lead to a significant impact on the adjustment performance. Figure 9 shows the tracking of the finger joints. As can be seen, the finger joint angles, or the hand posture gradually approach the desired values. Finally, we validated the robustness of the proposed grasping adjustment method against the pose estimation error.

B. Grasp in Real World

In this experiment, we considered a practical setup and validated whether the proposed grasping adjustment can still be feasible in reality where more uncertainties may be present. The experiment was carried out on an UR10-Allegro arm-hand platform.

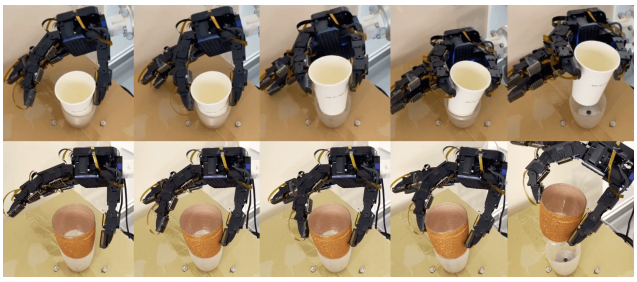

Fig. 11. Every row shows a set of grasping processes. The first object is a paper cup, the second one is a cambered glass. The initial posture of the hand in the leftmost column was adjusted by the experimenter.

As shown in Fig. 1, this platform contains two parts, the UR10 robotic arm and the Allegro Hand, which is equipped with tactile sensors. The overall system was implemented on the ROS platform. The sampling frequency of tactile sensor and the Allegro Hand joint encoder were 100 and 50 Hz, respectively. The kinematics of the UR10 equipped with Allegro Hand was solved by ROS-Moveit!, which limited the maximum acceleration and the maximum speed of the UR10. To ensure the effectiveness of the UR10 control, we set the control frequency of the UR10 near 30Hz. The evaluation was done on two additional objects, a paper cup and a cambered glass as shown in Fig. 11. For each grasping trial, the initial palm pose relative to the target object was randomly and manually set, but in the fingers' workspace where both fingertips (thumb and index finger) could make contacts with the object. The robot then followed the grasp execution and grasping adjustment. The grasping adjustment procedure of the two evaluated objects are shown in Fig. 11. The experimental results have confirmed that the proposed grasping adjustment strategy can still work in reality.

V. Conclusion and Future Work

In this paper, we propose a multi-fingered adjustment strategy for grasping. This is for compensating the grasping error generated by grasp planning and grasp execution. We have utilized the tactile features and the hand posture given

by the desired grasp to guide the adjustment. During the experiments, we have also identified a few facts about this algorithm. Firstly, this tactile servoing algorithm highly relies on the sensitivity and stability of the tactile sensor. In the case that the object is very light, the pressure applied on it can easily cause the change of poses and hence leads to task failure. We are working upgrading our tactile sensors to improve its sensitivity. Secondly, in this work only the fingertips are taken in consideration. In the future work, we will also consider the other parts of the hand, e.g. the rest of the finger and the palm, such that a whole hand grasp can be achieved, which is a more common situation and would improve the stability while grasping. On the other hand, we would like to adopt compliant control methods, e.g. impedance control, with which we may achieve a more compliant behavior of the fingers and the diversity between the actual finger positions and commanded positions caused by collision could be modelled.

REFERENCES

[1] J. Bohg, A. Morales, T. Asfour, and D. Kragic, "Data-driven grasp synthesis—a survey," *IEEE Transactions on robotics*, vol. 30, no. 2, pp. 289–309, 2013.

[2] A. Bicchi and V. Kumar, "Robotic grasping and contact: A review," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, vol. 1. IEEE, 2000, pp. 348–353.

[3] M. Grebenstein, M. Chalon, W. Friedl, S. Haddadin, T. Wimböck, G. Hirzinger, and R. Siegwart, "The hand of the dlr hand arm system: Designed for interaction," *The International Journal of Robotics Research*, vol. 31, no. 13, pp. 1531–1555, 2012.

[4] "Shadow robots," https://www.shadowrobot.com/, accessed: 2022-01-30.

[5] T. Asfour, K. Regenstein, P. Azad, J. Schroder, A. Bierbaum, N. Vahrenkamp, and R. Dillmann, "Armar-iii: An integrated humanoid platform for sensory-motor control," in *2006 6th IEEE-RAS international conference on humanoid robots*. IEEE, 2006, pp. 169–175.

[6] C. Borst, M. Fischer, and G. Hirzinger, "A fast and robust grasp planner for arbitrary 3d objects," in *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No. 99CH36288C)*, vol. 3. IEEE, 1999, pp. 1890–1896.

[7] Y.-H. Liu, M.-L. Lam, and D. Ding, "A complete and efficient algorithm for searching 3-d form-closure grasps in the discrete domain," *IEEE Transactions on Robotics*, vol. 20, no. 5, pp. 805–816, 2004.

[8] Y. Zheng and W.-H. Qian, "Coping with the grasping uncertainties in force-closure analysis," *The International Journal of Robotics Research*, vol. 24, no. 4, pp. 311–327, 2005.

[9] B. Bounab, D. Sidobre, and A. Zaatri, "Central axis approach for computing n-finger force-closure grasps," in *2008 IEEE International Conference on Robotics and Automation*. IEEE, 2008, pp. 1169–1174.

[10] T. Liu, Z. Liu, Z. Jiao, Y. Zhu, and S.-C. Zhu, "Synthesizing diverse and physically stable grasps with arbitrary hand structures using differentiable force closure estimator," *IEEE Robotics and Automation Letters*, vol. 7, no. 1, pp. 470–477, 2021.

[11] Q. Li, O. Kroemer, Z. Su, F. F. Veiga, M. Kaboli, and H. J. Ritter, "A review of tactile information: Perception and action through touch," *IEEE Transactions on Robotics*, vol. 36, no. 6, pp. 1619–1634, 2020.

[12] A. T. Miller, S. Knoop, H. I. Christensen, and P. K. Allen, "Automatic grasp planning using shape primitives," in *Proceedings of the 2003 IEEE International Conference on Robotics and Automation, ICRA 2003, September 14-19, 2003, Taipei, Taiwan*, 2003.

[13] D. Dan, Y. H. Lee, and S. Wang, "Computation of 3-d form-closure grasps," *Robotics & Automation IEEE Transactions on*, vol. 17, no. 4, pp. 515–522, 2001.

[14] A. T. Miller and P. K. Allen, "Graspit!: A versatile simulator for robotic grasping," *IEEE Robotics & Automation Magazine*, vol. 11, no. 4, pp. 110–122, 2005.

[15] Q. Li, C. Schürmann, R. Haschke, and H. J. Ritter, "A control framework for tactile servoing." in *Robotics: Science and systems*. Citeseer, 2013.

[16] B. Kehoe, D. Berenson, and K. Goldberg, "Toward cloud-based grasping with uncertainty in shape: Estimating lower bounds on achieving force closure with zero-slip push grasps," in *2012 IEEE International Conference on Robotics and Automation*, 2012, pp. 576–583.

[17] J. Laaksonen, E. Nikandrova, and V. Kyrki, "Probabilistic sensor-based grasping," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 2019–2026.

[18] K. Hsiao, S. Chitta, M. Ciocarlie, and E. Gil Jones, "Contact-reactive grasping of objects with partial shape information," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 1228–1235.

[19] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes," 2018.

[20] M. Li, K. Hang, D. Kragic, and A. Billard, "Dexterous grasping under shape uncertainty," *Robotics and Autonomous Systems*, vol. 75, pp. 352–364, 2016.

[21] Z. Chen, T. Wimböck, M. A. Roa, B. Pleintinger, M. Neves, C. Ott, C. Borst, and N. Y. Lii, "An adaptive compliant multi-finger approach-to-grasp strategy for objects with position uncertainties," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 4911–4918.

[22] R. Calandra, A. Owens, D. Jayaraman, J. Lin, W. Yuan, J. Malik, E. H. Adelson, and S. Levine, "More than a feeling: Learning to grasp and regrasp using vision and touch," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3300–3307, 2018.

[23] S. Levine, P. Pastor, A. Krizhevsky, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," 2016.

[24] E. Johns, S. Leutenegger, and A. J. Davison, "Deep learning a grasp function for grasping under gripper pose uncertainty," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 4461–4468.

[25] H. Dang and P. K. Allen, "Grasp adjustment on novel objects using tactile experience from similar local geometry," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 4007–4012.

[26] P. Pastor, L. Righetti, M. Kalakrishnan, and S. Schaal, "Online movement adaptation based on previous sensor experiences," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011, pp. 365–371.

[27] H. Dang and P. K. Allen, "Stable grasping under pose uncertainty using tactile feedback," *Auton. Robots*, vol. 36, no. 4, p. 309–330, apr 2014. [Online]. Available: https://doi.org/10.1007/s10514-013-9355-y

[28] M. Li, Y. Bekiroglu, D. Kragic, and A. Billard, "Learning of grasp adaptation through experience and tactile sensing," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 3339–3346.

[29] B. Wu, I. Akinola, J. Varley, and P. Allen, "Mat: Multi-fingered adaptive tactile grasping via deep reinforcement learning," 2019.

[30] N. F. Lepora and J. Lloyd, "Pose-based tactile servoing: Controlled soft touch using deep learning," *IEEE Robotics & Automation Magazine*, vol. 28, no. 4, pp. 43–55, 2021.

[31] Z. Kappassov, J.-A. Corrales, and V. Perdereau, "Touch driven controller and tactile features for physical interactions," *Robotics and Autonomous Systems*, vol. 123, p. 103332, 2020.

[32] H. Zhang and N. N. Chen, "Control of contact via tactile sensing," *IEEE Transactions on Robotics and Automation*, vol. 16, no. 5, pp. 482–495, 2000.

[33] V. K. Nanayakkara, G. Cotugno, N. Vitzilaios, D. Venetsanos, T. Nanayakkara, and M. N. Sahinkaya, "The role of morphology of the thumb in anthropomorphic grasping: a review," *Frontiers in mechanical engineering*, vol. 3, p. 5, 2017.

[34] Z. Ding, Y.-Y. Tsai, W. W. Lee, and B. Huang, "Sim-to-real transfer for robotic manipulation with tactile sensory," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 6778–6785.