Multi-contact haptic exploration and grasping with tactile sensors

Nicolas Sommer, Aude Billard

Algorithms and Systems Laboratory (LASA), Ecole Polytechnique Federale de Lausanne (EPFL), Switzerland

Abstract

Haptic exploration has received a great deal of attention of late thanks to the variety of commercially available tactile sensors. While the majority of previous works consider control of a single contact point at a time, we tackle simultaneous control of multiple contact points on several links. In addition, we use information from the existing tactile signals to increase the number of points in contact. We demonstrate the usefulness of this form of control to speed up exploration, scanning and to compliantly grasp unknown objects. Our controller requires to know only the parts of the robot on which it is desirable to make contact and does not need a model of the environment besides the robot itself. We validate the algorithm in a set of experiments using a robotic hand covered with tactile sensors and arm. In a grasping application, the active adaptation of the fingers to the shape of the object ensures that the hand encloses the object with multiple contact points. We show that this improves the robustness of the grasp compared to simple enclosing strategies. When combined with an exploration strategy, our multi-contact approach offers an efficient use of tactile sensors on the whole surface of robotic fingers, and enables the robot to perform a rapid exploration of complex, non convex shapes while maintaining low contact forces. It is robust to variation in the approach angle and to changes in the geometry and orientation of the object.

Keywords: Tactile Sensing, Haptic Exploration, Multiple Contacts, Compliant Grasping

1. Introduction

With robots moving into human-inhabited environment, haptic exploration becomes of primary importance to be able to interact with everyday objects. Other means of identification such as computer vision are limited by occlusion, illumination conditions and only provide partial information about texture and other surface properties.

In robotics, collisions are ordinarily avoided and in the cases when contact is allowed, it is usually limited to a single contact point at the end-effector. However, recent progresses in tactile sensing offer a range of research directions in robotics for allowing robots to be in contact at multiple points on the body. Moreover, thanks to advances in the design of dexterous humanoid hands that can manipulate complex shape, we can now consider manipulation that exploits the entire shape of the fingers. Such manipulation requires precise control of multiple contact points along the fingers.

Most research on haptic exploration has focused on a single contact [1, 2] on the end-effector or sequences of multi contact grasps [3], much less work has been done on continuous multi contact exploration. In order to map a surface or search for an object on it, it is more efficient to keep all fingers in contact while moving than to touch sequentially several points. Increasing the number of contact points also improves the overall time for the search or the reconstruction. Keeping contact during exploration becomes particularly crucial when the mapping must be precise and when the object being scanned is moving. This allows to keep a precise estimation of the relative position between the robot and the object.

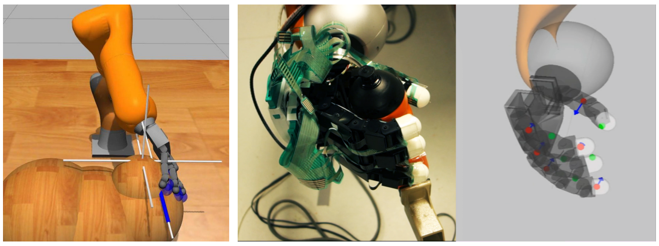
We propose an algorithm to maximize the number of points in contact when the hand is scanning or grasping an object. To this end, we project the forces/torques required for the exploration in the nullspace of the contact forces. Additionally, we control the forces at each contact point to prevent an uneven distribution of contact force. We show that this improves the robot's ability to make contact with unknown surfaces by using tactile sensors. This is crucial for tactile exploration and is very useful for grasping under uncertainty as tactile signals can guide the fingers to actively comply with the sensed shape.

2. Related work

2.1. Tactile exploration

Tactile exploration can be classified into two main categories: local and global exploration. Local exploration gathers local information about an object's surface, for instance by estimating the curvature at a given point on the surface, and matching that curvature profile to a database

Email addresses: `n.sommer@epfl.ch` (Nicolas Sommer), `aude.billard@epfl.ch` (Aude Billard)



(a) Surface exploration in simulation.

(b) Compliant grasping and attractors.

Figure 1: Two of our experiments: a robotic arm and hand system explores a shape in simulation and grasps compliantly an object on a real platform.

of objects for identification [4]. Another type of local exploration consists in identifying surface properties, such as roughness, fineness and traction, and using these properties to classify materials [5].

However, we are more interested in the global exploration category, as it implies a more complex motion of the probe. Most of the research focuses on moving a sensorized end-effector on the surface of an object. For instance, a robotic finger equipped with a tactile sensor is moved on the surface of an object to detect fine surface features [1], or a tactile array on top of a robot arm moves around the surface of an object to reconstruct a 2-D pressure profile [2]. Another example of haptic exploration adapts the stiffness of the exploration controller depending on the resistance to the motion [6], following inspiration from neuroscience studies. In [7], the authors tackle the task of exploring a discontinuous surface with a rolling end-effector and force-torque information with a compliant controller. This involves adjusting the controller to the orientation of the surface normal to maintain a desired normal force. In [8], both shape and friction coefficient are modeled under a probabilistic framework during the exploration of an object's surface with both a tactile probe and an RGB-D camera.

While exploration with a single end-effector simplifies the control, it has limitations: when the probe is small, the exploration process is very slow, especially if the surface to be covered is large. However, if the probe is large, it cannot comply with arbitrary shapes (especially for convex objects) or cannot reach some areas. Another area of research focuses on using robotic hands and fingers or grippers, and tactile or force sensors to model the object's shape. One approach is to grasp objects sequentially, at different locations, and reconstruct a "bag-of-features" representation [9]; this allows to keep a sense of continuity and represent the object globally from local features without the need for precise localization during the exploration. A systematic approach is also used for reconstructing 3-D point cloud models of objects with a 3-fingered hand and tactile sensors [3]; however, this is a very slow method because of the systematic probing (the fingers are opened and closed a hundred times for an object smaller than 9cm), and is restricted to small objects that can fit between the robot's fingertips.

Finally, we are particularly interested in continuous exploration with several fingers, as it is better suited to reconstruct the shape of an object thanks to the flexibility of the multiple degrees of freedom available. The first works focused on the reconstruction of parametric models of objects: already in 1990, Allen, inspired by exploratory procedures from Lederman and Klatzky [10], explored objects modeled by superquadrics with a contour following method that used the model's parameters to compute a trajectory [11].

In our previous work [12], we tackled tactile exploration of human-like faces for classification, using fingertip tactile sensors and statistical encoding of the face models for robustness to proprioceptive noise. We also developed an algorithm for bi-manual haptic exploration of objects [13], using multiple phalanxes per finger, covered with tactile sensors, for the 3-d reconstruction of the objects' shapes. However, these approaches were limited in the range of object's shapes that could be explored.

A recent work by Bierbaum et al. [14] introduces the use of potential fields to drive the exploration of a five-fingered hand in simulation. While this allows autonomous reconstruction of several simple objects, the hand is controlled in velocity and thus the interaction forces are not taken into account. Besides, the exploration only uses the fingertips. There is no contact with the other finger links, as only fingertips are subject to the potential field.

One of the most advanced works tackling tactile interaction is probably the one from Jain et al. [15] in which multiple contacts occur on the arm of a robot, not for exploration explicitly but to help the robot reach trough cluttered space. They use model predictive control with a linear model of the contacts that assumes linear stiffness and optimizes for reaching a desired position with the end-effector, with constraints on contact forces. However, the objective of the controller is to reach a point with the end effector. It ignores the posture of the rest of the arm: there is no focus on the tactile exploration itself. Our exploratory approach is also different as it requires to command the robot in torque and thus is better suited at controlling contact forces.

2.2. Prioritized controllers

Haptic exploration fits well within the prioritized controller scheme, as some tasks – managing contact forces, avoiding joint limits – can be interpreted as constraints and be given a very high priority, while other tasks are less important. This framework is commonly used for humanoid control, including constraints on contact forces, but not for haptic exploration. Khatib's operational space framework [16] allows to express the dynamics of the robot in task coordinates, and the prioritized simultaneous control of several tasks through cascaded null space projections [17]. More recently, this framework was used to control several contacts on different links of a robot arm [18], but there have not been results showing cases where the robot makes additional unpredicted contacts or looses some of its contacts. In [19], commands are automatically scaled down if they violate hard bounds at the joint level (position, velocity or acceleration constraints). This allows to have explicit hard constraints, which was usually not possible in that framework. Another approach towards prioritizing tasks is to formulate the inversion of the Jacobian as a quadratic problem. For instance, the Stack of Tasks approach (SoT) [20] provides an interface to add and remove tasks automatically with a pre-specified hierarchy. Recently, hierarchical control schemes based on a sequence of quadratic programs (QP) can also handle inequality constraints for kinematic control [21] and dynamic control [22]. Efforts have also been made to solve these problems fast enough for real-time control of humanoid robots with many degrees of freedom [23].

In this work, we follow the null-space approach to prioritizing tasks and introduce a controller based on a modified null-space projection matrix that allows to take into account inequality constraints. While this is not as efficient as the latest QP-based methods, this is an alternative approach simpler to implement since it only relies on matrix inversion and does not require an otherwise complex solver.

2.3. Grasping under uncertainty with multiple DOFs hands and tactile sensors

Besides stability conditions, complying with the shape of an unknown object during grasping shares similarities with the exploration of unknown objects. In both cases, external sensory information is necessary to actively comply if there are uncertainties in the position or the shape of the object, or in the robotic system itself. Most of the work in grasping consists in planning grasps for known or partially known objects [24, 25, 26]. However, reliably controlling robotic fingers to realize generated grasps on a real platform with position and shape uncertainties remains a problem. Indeed, it is difficult to realize the planned grasps with a real robot hand, and this makes the quality evaluation less relevant in practice, as the realized grasps are less optimal than the planned ones [27]. While *soft* systems approach this problem using passive mechanical compliance to adapt to position or shape uncertainties, active compliance is the only way to control rigid robotic hands with multiple degrees of freedom. Using additional sensory information, one can improve grasping success rates by detecting position errors. For instance in [28], torque sensors in the fingers are used to detect the first contact and compliantly pause the finger in contact before it tips over the object to be grasped. In [29], the authors exploit tactile sensors on the fingertips to control the finger contact force under shape uncertainty. Platt's nullspace grasping control [30] uses local object geometry measurements to guide grasps and converge to unit frictional equilibrium. This involves following the negative gradient of two functions: force and moment residuals which are zero at this equilibirum. Because the force residual controller displacements are tangential to the surface, and the moment residual controller displacements are projected on the null space of the gradient of the unit frictionless force residual, the resulting motion corresponds to the fingertips sliding on the surface of the object. However, the ensured improvement of the chosen grasp metric is based on several assumptions: convex objects, 2nd order continuity of the surface and only two contact points. Finally, in these works, only the fingertips are taken into account and no attempt is made at controlling grasps with contacts on all links of the hand.

When the whole hand (not limited to the fingertips) is used to grasp an object, realizing planned grasps becomes even more difficult as multiple contacts should be made between the fingers and the grasped object. Grasping synergies is an efficient concept to simplify control of high-dofs hands inspired by human grasping. Whether the synergy is integrated in the mechanical design of sub-actuated hands [31] or simulated in software [32], it decreases the dimensionality of the control problem. However, synergy-based grasping strategies can also lead to unsuccessful grasps and they do not seek to maximize the contact surface. For this reason, our algorithm can provide active compliance at all the desired contacts points on the fingers.

3. Controller structure

In order to explore its environment or grasp an object, the robot needs to create contacts. However, some contact points can be desired while others might just occur during the exploration and not be desirable. Here, we explain how we differentiate between these two types of contact and how both of them are taken into account by our controller. First, we introduce the operational space coordinates using the contact points.

Operational space coordinates using contact normals. At each timestep, for each contact point $i \in \{1,..,N_c\}$ detected by tactile sensors, we define its position $p_c^i \in \mathbb{R}^3$, normal direction $n_c^i \in \mathbb{R}^3$, parent joint $l_c^i \in \{1,\dots,N\}$, and Jacobian $J_c^i \in \mathbb{R}^{1\times N}$. The set C contains the joints attached directly above a link that currently hosts a contact point and the set n_c the normals of contact:

$$C = \{l_c^i\}_{i=1}^{N_c}, \qquad n_c = \{n_c^i\}_{i=1}^{N_c} \tag{1}$$

with N_c the number of contact points and N the total number of DOFs of the robot. The operational space contact Jacobian J_c^i is computed as $J_c^i = {n^i}^T J^i$, where J^i is the contact Jacobian expressed in the robot base frame.

The Jacobian for the operational space coordinates is given by the concatenation of all these contact Jacobians:

$$J_c = \begin{pmatrix} J_c^1 \\ J_c^2 \\ \vdots \\ J_c^{N_c} \end{pmatrix} \tag{2}$$

Table 1: Notation table

Variable	Description
C	Current contact points
D	Desired contact points (not time dependent)
$C \setminus D$	Current undesired contacts
$D \cap C$	Current desired contacts
$D \setminus C$	Desired contacts not yet in contact
$p_c^i \quad (p_d^i)$	Position of (desired) contact point i
$n_c^i \quad (n_d^i)$	Normal direction of (desired)contact point i
$l_c^i \quad (l_d^i)$	Parent link of (desired) contact point i
$J_c^i \quad (J_d^i)$	Jacobian of (desired) contact point i
$N_c \quad (N_d)$	Number of (desired) contact points
J_c	Operational space Jacobian
N	Total robot DOFs
q	Vector of joint angles
M_q	Joint-space inertia matrix
b	Coriolis and centrifugal torques
g	Gravity torques
f	Contact forces
τ	Commanded robot torques
$\ddot{x}$	Operational space acceleration of contact points
τ_0	Lower priority torques
N_{τ_0}	Null-space projection matrix, dependent on τ_0
$J_{C \cap D}$	Jacobian of desired contact points
$J_{C \setminus D}$	Jacobian of undesired contact points
J_{τ_0}	Modified operational space Jacobian
K_d	Stiffness matrix of impedance controller
D_d	Damping matrix of impedance controller
p_r^j	Reference position of desired contact point j
p^j	Current position of desired contact point j
v^j	Desired velocity of desired contact point j

Set of areas that can be in contact. During the task execution, the more contact points between the robot and its environment, the more information is retrieved at the same time. For instance, it would be inefficient to try to localize an object on a table using only one fingertip. However, increasing the number of contacts also decreases the manipulability of the robot as each contact introduces a dynamic constraint, as detailed in section 3. A mechanism for deciding whether a contact is desired or not is thus required.

In addition to the contacts points defined in the previous section, we define the desired contacts points $i \in \{1, .., N_d\}$ with position $p_d^i \in \mathbb{R}^3$, parent joint $l_d^i \in \{1, \dots, N\}$, and Jacobian $J_d^i \in \mathbb{R}^{1 \times N}$. The set D contains the joints which directly control a link that hosts a desired contact point.

$$D = \{l_d^i\}_{i=1}^{N_d} \tag{3}$$

with N_d the number of desired contact points.

The mechanism of choosing the desired contact points depends on many criteria, including the robot platform, the task and possible prior on the shape to explore or grasp[1]. The combinations of the sets C and D and what they represent are detailed at the beginning of Table 1.

Control of the robot in contact. The dynamics of the robot are of the form:

$$M_q(q)\ddot{q} + b(q, \dot{q}) + g(q) + J_c^T(q) f = \tau \tag{4}$$

where $q, M_q(q), b(q, \dot{q}), g(q), f$ and τ are respectively the vector of joint angles, the joint-space inertia matrix, the Coriolis and centrifugal torques, the gravity torques, the contact forces and the vector of joint torques[2]. The torques τ applied to the robot are chosen in the form of a prioritized controller:

$$\tau = \tau_1 + N_{\tau_0}(q)\tau_0 \tag{5}$$

where τ_1 are the torques for the highest-priority task, e.g. the contact force control, N_{τ_0} a modified null space projection matrix that depends on τ_0, the vector of lower priority torques.

The null space projection matrix $N(q)$ is usually chosen so that any torques projected on it do not affect the operational space acceleration $\ddot{x}$. The manipulator dynamics in the operational space are given by pre-multiplying (4) with $J(q)M_q(q)^{-1}$. For better readability, we do not specify the dependency on the joint angles vector q and its derivatives from now on:

$$\ddot{x} - \dot{J}\dot{q} + JM_q^{-1}(b + g) = JM_q^{-1}(\tau - J_c^T f) \tag{6}$$

The terms $\dot{J}\dot{q}$ and $JM_q^{-1}(b + g)$ can be compensated for, therefore $\ddot{x}$ is not affected by projected lower-priority torques whenever:

$$JM_q^{-1}N = 0 \tag{7}$$

We want to avoid creating accelerations or forces for existing desired contacts. This is in order to avoid disturbing the control of the contact force. However, the robot might also host contacts on links which do not accept contacts, i.e. $i \in C \setminus D$. In that case, strictly positive operational space accelerations – towards the surface – should be avoided, but negative accelerations can be accepted as they will break the undesired contact. The strategy adopted here is to take into account the null-space torques τ_0 in the computation of a modified null space projection matrix. We separate the Jacobian space operational matrix J_c in two sub-matrices $J_{C \cap D} \in \mathbb{R}^{N \times N_{C \cap D}}$ and $J_{C \setminus D} \in \mathbb{R}^{N \times (N_c - N_{C \cap D})}$ containing the concatenated desired contact Jacobians and undesired contact Jacobians, with $N_{C \cap D}$ the number of existing desired contacts. The new conditions are expressed with a modified null space projection matrix N_{τ_0} that respects the following constraints:

$$J_{C \cap D} M_q^{-1} N_{\tau_0} = 0 \tag{8}$$

$$\text{and } \forall \tau_0 \in \mathbb{R}^N, \quad J_{C \setminus D} M_q^{-1} N_{\tau_0} \tau_0 \leq 0 \tag{9}$$

This can be ensured by constructing a modified operational space Jacobian matrix J_{τ_0} that contains only the Jacobians of the desired contacts, plus the Jacobians of the undesired contacts which would create undesired contact forces because of the torques τ_0.

[1] In this work, we assigned a desired contact point on each link of the robot's hand, hence 3 per finger.

[2] We do not model joint friction, this is an approximation, especially for the joints in the hand. See the discussions about neglecting friction in the results section of the experiments.

Algorithm 1: Modified Jacobian for null space computation

```
Data: The Jacobians $J_{C\cap D}$ and $J_{C\setminus D}$
Result: The modified Jacobian $J_{\tau 0}$
$J_{\tau 0} = [J_{C\cap D}]$;
for $i \leftarrow 1$ to $(N_c - N_{C\cap D})$ do
    if $J^i_{C\setminus D} M_q^{-1} \tau_0 > 0$ then
        $J_{\tau 0} = \begin{bmatrix} J_{\tau 0} \\ J^i_{C\setminus D} \end{bmatrix}$ // Concatenate jacobians
return $J_{\tau 0}$;
```

Finally, we compute the modified null space projection matrix $N_{\tau 0}$ from the modified Jacobian $J_{\tau 0}$, obtained with Alg. 1:

$$N_{\tau 0} = I - J_{\tau 0}^T \bar{J}_{\tau 0}^T \tag{10}$$

$$\text{with} \quad \bar{J}_{\tau 0} = M_q^{-1} J_{\tau 0}^T (J_{\tau 0} M_q^{-1} J_{\tau 0}^T)^{-1} \tag{11}$$

The equation (5) for controlling the robot is finally detailed as:

$$\tau = \tau_c + N_{\tau 0}(\tau_d + \tau_r + \tau_e) \tag{12}$$

The torques τ_c, τ_d, τ_r, and τ_e are described in the following sections, they respectively represent torques for the contact forces, for increasing the number of contacts, for tracking a rest position (for some of the joints), and for driving the exploration of the robot in the corresponding experiment.

This is easily extended to multiple levels of task priorities. The procedure is described in Alg. 2. This is important for instance to add joint limits and joint centering tasks.

Algorithm 2: Multi-priority algorithm

```
Data: For each task $task$, its priority level $i$, torques $\tau_{task}$ and
      Jacobian $J_{task}$. For the contact tasks, this jacobian is
      computed according to Alg. 1
Result: Torques $\tau$
$\tau_{prev} = 0$
for priority $i \leftarrow 1$ to $p_{max}$ do
    $J_i = []$,    $\tau_i = 0$
    for $task \in tasks_i$ do
        $J_i = \begin{bmatrix} J_i \\ J_{task} \end{bmatrix}$
        $\tau_i$ += $\tau_{task}$
    $N_i = I - J_i^T \bar{J}_i^T$
    $\tau = \tau_i + N_i \tau_{prev}$
return $\tau$;
```

4. Increase contact area

In order to gather information about its environment by means of touch, or to grasp an object, the robot must at first make contact. Then, once in contact, increase as much as possible the area in contact. To this end, we proceed to switching across two modes of control: one mode controls links not yet in contact, and the other mode controls the contact force at the joints already in contact. We start by explaining how we determine which mode of control to use.

All joints in the fingers that affect control of the force at the contact points and the desired contact points are controlled in torque according to Eq. 14-15. The other joints in the fingers are controlled by a PD controller.

At each time step, the control mode for each joint depends on its position relative to the contact points and desired contact points. The set of joints with existing desired contact points is given by $C \cap D$ – both part of the current existing contacts and desired contacts –, the set of joints with desired contact points is given by $D \setminus C$ – desired contact but not a contact yet –. The set K contains all joints that are not on the same kinematic chain as either a contact point or a desired contact point, and all joints when the robot is not in contact:

$$K = \{i \mid \forall j \in S_i, j \notin C \cup D \parallel C = \varnothing\}_{i=1}^N \tag{13}$$

with $N \in \mathbb{R}$ the total number of DOFs, and S_i the set of joints belonging to the kinematic chain starting from the base of the robot, passing through the joint j until one of the fingertips.
The set K hence contains all joints not used for control of force at contact or desired contact point. These joints track a predefined rest position with a PD controller. In the case that the robot is not in contact, a higher-level controller is in charge of bringing the robot into contact, as explained in the experiments section.

Control of existing contact points. For each existing contact point that is located on a link i which accepts contact points, e.g. for $i \in C \cap D$, we apply a small normal force to maintain the contact:

$$\tau_c = \sum_{i \in C\cap D} J_c^{i\,T} f_n \tag{14}$$

where $f_n \in \mathbb{R}$ is the desired normal force applied at the contact point[3].

Control of desired contact points. The desired contact points are used to increase the number of contact areas. For each desired contact point that is not on a link where there is already a contact point, a corresponding Cartesian reference position is computed and tracked with a Cartesian impedance controller. The position of each desired contact point is predefined for each link of the robot[4].

[3] In our experiments, we chose $f_n = 0.5N$. This value must be above the sensitivity of the tactile sensors to be able to sense contacts. If there is friction in the joints, a higher value is useful because friction can lead to a non-zero acceleration at a contact point despite the nullspace projection, and potentially loose contact.

[4] In this work, the position is defined as the geometrical center of the links.

The desired contact points are controlled by impedance control, thus the total torque for the desired contacts:

$$\tau_d = \sum_{j \in D \backslash C} {J_d^j}^T (K_d x_e^j + D_d \dot{x}_e^j), \tag{15}$$

$$\text{with } x_e^j = p_r^j - p^j \tag{16}$$

with isotropic stiffness and damping matrices $K_d = k_d \cdot I_{3\times3}$ and $D_d = d_d \cdot I_{3\times3}$[5], p_r^j, $p^j \in \mathbb{R}^3$ the reference and current Cartesian positions of the desired contact point j. The reference Cartesian position is computed by integrating the desired velocity v^j:

$$p_r^j = p_{init}^j + \int_{t_{init}^j}^{t} v^j \, \mathrm{d}t \tag{17}$$

The desired velocity v^j should be chosen according to the task and can take into account prior about the explored surface. The details are given for each of our applications in the experiments section. The initial Cartesian position p_{init}^j and initial time t_{init}^j are reset to the current values p^j and t when the desired contact j is created. This happens either when an existing contact is lost and turns into a desired contact or when the set of desired contact changes.

5. Experiments

The algorithm we propose here is meant to be used with robots that have the ability to sense contacts at multiple joints. Unfortunately, to date, there is no commercially available technology to cover a robot entirely with a sensitive skin. In our experiments, we hence first conduct simulations, emulating a perfect sense of touch on all sides of the fingers. We then perform smaller scale - proof of concept - implementations using a real robotic hand covered with patches of tactile sensors. The first setup is a simulation of a robotic arm with a robotic hand attached at the end: there are 7 DOFs for the arm and 4*4 DOFs for 4 fingers. Second, a similar configuration with a real robot equipped with tactile sensors. The simulation allows to control the whole robot and hand in contact with an unknown environment, without the risk of contacts not being detected (e.g. if they occur on an unsensorized part of the robot) and cause damage[6]. The experiments were conducted in order to prove the effectiveness of the control strategy and the ability to interact with unknown objects and surfaces.

The experiments consisted of an exploration part, conducted in simulation, and an active compliance for grasping part, both in simulation and on a robot.

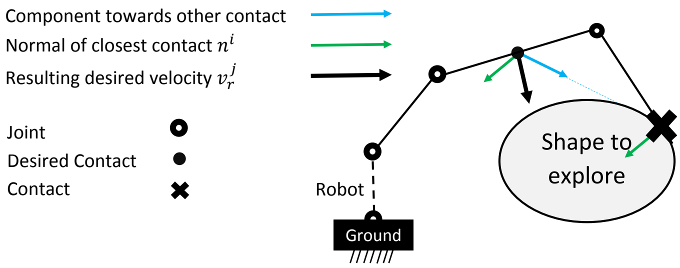

Component towards other contact
Normal of closest contact n^i
Resulting desired velocity v_r^j
Joint
Desired Contact
Contact
Robot
Ground
Shape to explore

Figure 2: Exploration: illustration of the computation of the velocity vector v^j for the desired contact points.

5.1. Exploration

The first applications consist of the exploration of unknown surfaces, using our algorithm to actively comply with the unknown shape of the surfaces. The first experiment consists in the full autonomous exploration of a random shape, and the second experiment in the exploration of the inside surface of cups, using all edges of the fingers. A third experiment tests our algorithm with two different hand models on two new objects. All three of these experiments are carried out in simulation. We aim to demonstrate that maximizing the number of contacts during exploration helps at reconstructing the surfaces and thus gaining information about it, while keeping all contact forces low.

For the exploration, the desired velocity v^j introduced in Eq. (17) of the desired contact point p_r^j is defined by the average of the closest point's normal and the direction towards the closest point (see Fig 2).

$$v^j = \lambda \cdot \frac{n^{i^*} + (p^{i_c^*} - p^j)}{\|n^{i^*} + (p^{i_c^*} - p^j)\|}, \quad i^* = \underset{i \in \{1,\dots,N_c\}}{\arg\min} \{p^j - p_c^i\} \tag{18}$$

where n^{i^*} is the normal of the closest contact point $p^{i_c^*}$, and λ a predefined scalar velocity.

We use a simulated 7 degrees of freedom (DOF) Kuka Light Weight Robot arm with a 16 DOF AllegroHand robotic hand. We aim at showing that the robot can autonomously reconstruct a random shape. Because these experiments are carried out in simulation, there is no error in the reconstruction as each datapoint lies perfectly on the explored surface. The coverage of the surface reconstruction however depends on the chosen exploration strategy, which is in our case very simple. It also directly depends on the number of contacts during the exploration, hence the goal to maximize it. The simulation is run at 1000 Hz in Gazebo with ODE and the simulated robot is directly controlled in torque. The computation of the modified null-space projection matrix also runs at about 500-1000Hz (depending on the number of contacts) in a different thread on a PC with a Core i7 processor at 3.6Ghz.

[5] We choose $k_d = 5N.m^{-1}$, $d_d = 1N.s.m^{-1}$

[6] We do not currently have tactile sensors covering the robot's arm. However, we are designing a method to reconstruct the point of contact from the torque sensing at joint level.

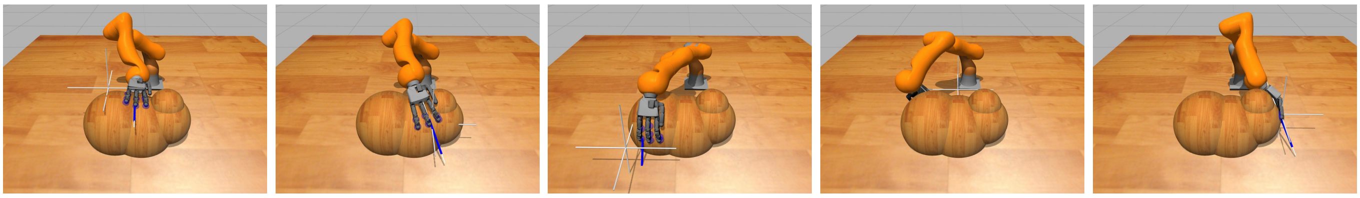


Figure 3: **Exp 1** Exploration of the shape (50 seconds). On each image, the thin white frame indicates the currently tracked reference frame.

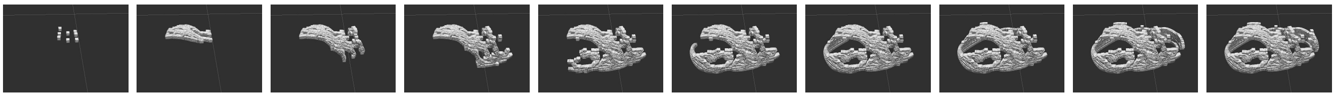


Figure 4: **Exp 1**: 3-D reconstruction of the shape during the exploration.

Since one of the desired constraints is to avoid high contact forces, we record the average and the maximum interaction forces during the experiment (at each timestep, among all contact points). We also record the number of contact points during the exploration.

5.1.1. Exp 1: Exploration of a surface

A complex shape composed of several spheres of different diameters is explored. The composition of spheres with variable radiuses creates a non-convex shape, hence the orientation of the hand is critical since the shape needs to be approached from different angles. The reference positions are distributed sequentially around the shape. The goal of the experiment is to autonomously explore and reconstruct the surface of an unknown arbitrary shape, with only a few given key reference positions around the shape to drive the direction of exploration. In order to follow the surface to be explored, the reference position and orientation of the controller need to be defined. They are determined using information from the tactile contacts. A controller is implemented to direct the hand towards the desired exploration locations. This controller is detailed in the appendix section Appendix A. The exploration is performed 10 times.

Results. The shape is properly reconstructed from the tactile sensing information, progressively as can be seen in Fig 3 after on average 45 seconds of exploration[7]. The hand successfully changes orientation autonomously to explore the different faces of the explored shape. In a few runs of the experiment, one of the fingers bends and the contact occurs on the back of the finger, see Fig 6. While this does not lead to an increase in the contact force, it creates a hand configuration that is less optimal for exploration as less contacts can be made. This issue is further discussed in the discussion section.

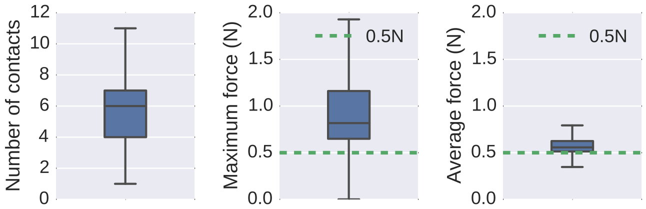


Figure 5: **Exp 1**: Number of contact points, maximum and average contact force during the exploration. The green dotted line represents the desired contact force of 0.5N.

During the exploration, the maximum simultaneous contact force is on average of $1.00 \pm 0.57N$, with a maximum of $4.59N$, while the average contact force is of $0.56 \pm 0.12N$. The distribution of these forces and the number of contact points can be seen as a boxplot representation in Fig 5. These forces should be compared with the desired force at each contact point, controlled in open-loop through the Jacobian transpose method, and set to be 0.5N. Indeed, the null-space projection prevents the commanded torques from influencing the contact forces by construction. However, the forces due to the dynamics of the robot are not compensated and can therefore influence the contact torques. Because the robot moves slowly in this experiment, as a robot should while it is in contact, the forces due to the dynamics of the robot are low and the contact forces do not vary far from the reference contact force of 0.5N.

The number of contacts points during the experiment oscillate between 1 and 11, including when the hand starts making contact with the object at the beginning. The thumb is not used in the experiment because its kinematic configuration does not allow it to comply properly with the shape. The average of 6 simultaneous contacts means that each finger has on average two links in contact.

5.1.2. Exp 2: Exploration of the inside of a cup

This experiment consists in exploring the inside of a cup with a robotic hand. This requires to establish contacts

[7] Video of the experiments:
`http://lasa.epfl.ch/videos/downloads/sommer_he.mp4`

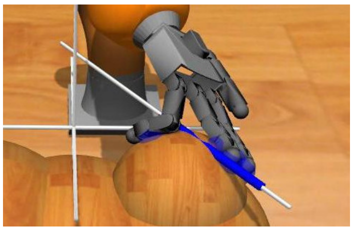


Figure 6: **Exp 1**: In some runs of the experiment, one of the fingers makes contact with its back side. While this is not a problem in terms of contact forces, it leads to a hand configuration that is less optimal for the exploration of certain surfaces.

on all sides of the fingers and comply with very curvy and non-convex shapes.

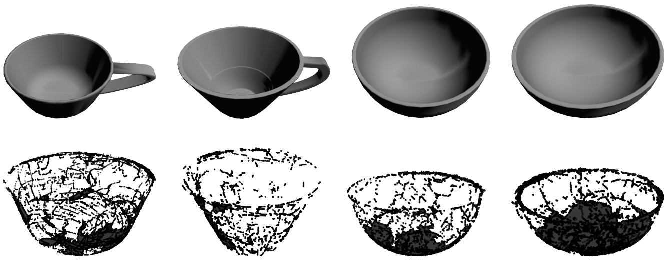
The arm controls the position and orientation of the hand. The hand is introduced vertically inside the cup at its center until all fingers are in contact, then it is slowly rotated around the axis of symmetry of the cup for a given angle. The hand is then moved up out of the cup. Before any contact occurs, the fingers are kept in a resting position with all joints slightly bent. The objective is to be able to gather information about the explored object without creating too high contact forces. We hence record at each timestep the number of simultaneous contacts and the current average and highest contact force (of all the current contacts). While the arm is controlled in position, we compared our algorithm, which tries to maximize the contacts, on the hand, against a simple compliant controller with two different sets of gains for the finger joints (see Table 2). The reference position of the joints can be seen in the first image of Fig 8, which also displays the progression of the experiment. It is also the rest position used by the active exploration algorithm. We explored 4 different models of cups presented in Fig 7.

Results. Table 3 sums up the results for each control method, with the average and standard deviation of the two previous values. Fig 9 gives the distribution of the number of contacts and the maximum contact force by cup and by control method.

Our method provides more contact points during the exploration of the cups than the other two methods (5.4±1.5 vs 2.6±1.3 and 2.4±1.1). Since the contact forces are set to be at $0.5N$ with our algorithm, the average measured force is of $0.53 \pm 0.09N$. It also keeps a lower maximum contact force, $0.77{\pm}0.26N$, slightly higher than the $0.5N$ reference (go back to section 5.1.1 for a discussion about the maximum contact forces). As expected, the controller with a higher compliance has lower contact forces $(0.61{\pm}0.0.67N)$ than the one with a lower compliance $(1.15 \pm 1.32N)$, but both are higher than with our method. Besides, the number of contacts is similar for both of the compliance controllers, which makes the one with high compliance more interesting. However, this might not hold on a real robot as friction in the joints might prevent the use of low gains.

Table 2: **Exp 2**: Parameters of the tested controllers

	Active adaptation	**Low compliance**	**High compliance**
P gain	-	0.05	0.005
D gain	-	0.01	0.01



(e) Cup1 (f) Cup2 (g) Cup3 (h) Cup4

Figure 7: **Exp 2**: Models of the explored cups above, and reconstructed versions in rviz after exploration below.

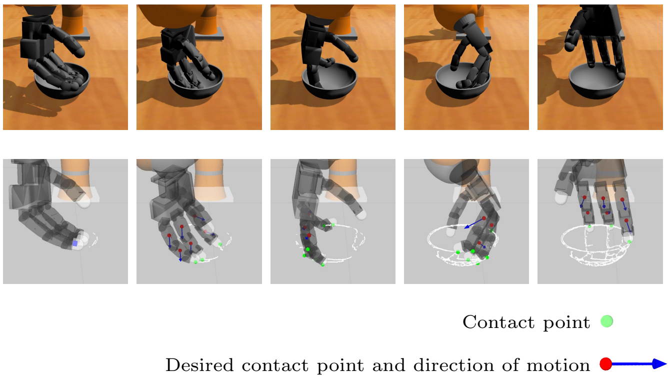
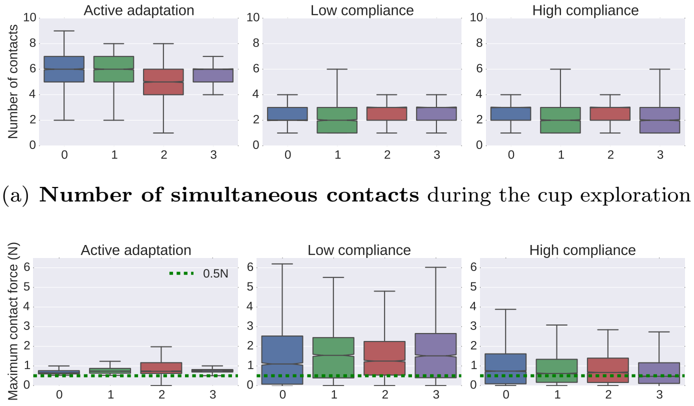


Figure 8: **Exp 2**: Progression of the exploration of *cup 4* with the active compliance algorithm in gazebo (top), with reconstruction of the shape with a tactile point cloud in rviz (bottom). Green dots are the actual contact points, red dots are desired contact points and the blue arrows their desired direction of motion v^j.



(a) **Number of simultaneous contacts** during the cup exploration

(b) **Maximum contact force (N)** during the cup exploration

Figure 9: **Exp 2**: Number of contacts and maximum contact force for each control method and cup.

Table 3: **Exp 2**: Results

	Active adaptation	**Low compliance**	**High compliance**
Nb. contacts	5.39 ± 1.46	2.62 ± 1.25	2.41 ± 1.14
Max force (N)	0.77 ± 0.26	1.67 ± 1.71	0.86 ± 0.84
Average force (N)	0.53 ± 0.09	1.15 ± 1.32	0.61 ± 0.67

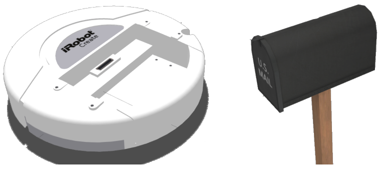


Figure 10: **Exp 3**: Objects explored: the IRobot Create and a mailbox.

5.1.3. *Exp 3: Exploration with other hand configurations*

In addition to the exploration of the shape based on spheres in Exp 1 and cups in Exp 2, we also tested our algorithm with two different hand models. The first configuration, called here *Hand 1*, is the same hand as used in the previous experiments, with released joint limits in order to allow the phalanxes to bend both forward and backward. This allows the fingers to take new configurations during the exploration. This can be especially interesting to explore non-convex objects. The second configuration, *Hand 2*, is also based on the same hand, with an additional finger. This gives a total of 4 fingers and thumb, similar to a human hand. For this experiment, we tested our algorithm and other hand models on two new objects, an *IRobot Create* robot and a mailbox, see Fig 10. The vacuuming robot contains a concave shape created by the empty dust holder, while the mailbox is made of many sharp edges, which makes it more difficult to keep all contacts.

Results. Predictably, there are in average a few more contacts made with *Hand 2* than with *Hand 1*, since it has one additional finger (Fig 13). The difference between *Hand 1* and *Hand 2* in terms of number of contacts is smaller on the first object as the released joint limits of *Hand 1* give it an advantage for complying to the complex shape of the vacuuming robot. The maximum and average contact forces are in the same range and match the results obtained with the original AllegroHand model in the previous experiments. The point cloud representation of the objects after exploration can be found in Fig 11.

5.2. *Exp 4: Compliant grasping*

Another application of our active compliance algorithm is to grasp objects by enclosure, maximizing the contacts between the object and the hand and fingers. We compared our method, called here *active adaptation*, with two simple grasping heuristics for enclosing, which can correspond to very simple synergy-based grasps.

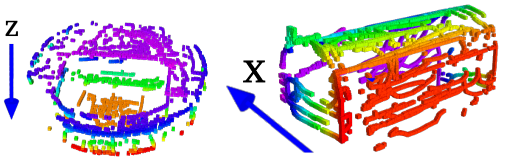


Figure 11: **Exp 3**: Reconstructed point cloud of the objects in rviz. For readability of the 3D representation, the color of the points corresponds to their coordinate on the axis indicated by the blue arrow.

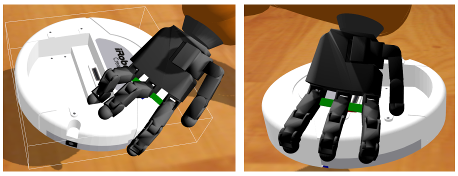


Figure 12: **Exp 3**: Snapshots of the exploration of the vacuuming robot with *Hand 1*. The released joint limits allow the fingers to comply to the concave shape in the middle of the robot.

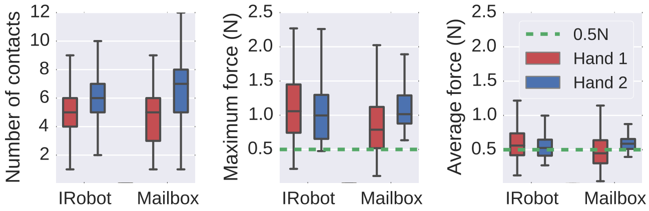


Figure 13: **Exp 3**: Number of contact points, maximum and average contact force during the exploration of the IRobot Create and Mailbox models. The green dotted line represents the desired contact force of 0.5N.

The first method (*Enclose1*) consists in closing the joints of the fingers one by one from the base to the tip, until a contact is reached. The second method (*Enclose2*) is similar, but all joints close simultaneously until there is a contact above the joint on the same finger. It is thus faster, but there is a risk that less contacts are made if a link at the end of the finger touches first.

The chosen grasp preshape is inspired by the grasp opposition of the thumb vs. the other fingers from [33], which defines a grasp intention by a sum of patches (finger links) oppositions, with for each opposition set, a dominant patch per side. The chosen grasp intention is that of a power grasp for a cylinder of about 3cm of diameter, which is a description that roughly matches all of our tested objects. This provides us with a grasp preshape. It also provides our algorithm with a list of desired contact points (all the patches) and an opposition direction to help define our desired contact points velocities v^j. We only use one opposition direction, therefore we have two dominant patches (one per side of the opposition) and use the virtual line between them to define the direction of the desired velocity[8]

[8] The norm of the desired velocity is set to $5cm/s$.

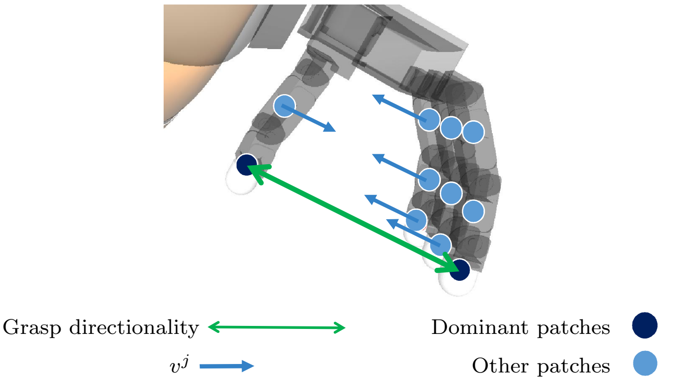


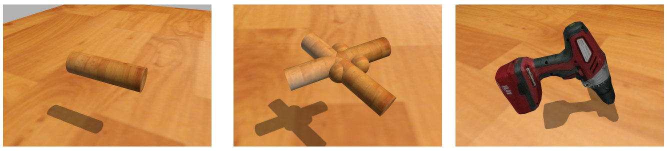
Figure 14: **Grasping**: illustration of the grasp preshape position and the computation of the velocity vector v^j (aligned with the grasp directionality) for the desired contact points using grasp opposition.

for each desired contact point (see Fig 14).

We tested the three methods systematically both on a simulated and a real robot. The first part of the experiment (**Exp 4a**) consists in enclosing objects at a predefined grasp position. The object is then sequentially released and grasped again in four other configurations shifted by 2cm in two different directions, and shifted by 17° in two different orientations. These shifts correspond to potential position and orientation uncertainties that the robot might have to deal with in a real application.

In the second part of the experiment (**Exp 4b**), the robot first grasps the objects at the initial position, and we apply sequentially the position and orientation perturbations while the object is grasped. This is important to test how the algorithm adapts to external perturbations. This creates five possible enclosure configurations for Exp 4a, and 4 possible perturbations for Exp 4b. Similarly to the previous experiments, we record the number of contacts made between the object and the hand (when the grasp is finished), and the contact force. However on the real robot, our tactile sensors do not provide values convertible into contact forces since their output depends on the type of material and the area in contact. They do not depend directly on the contact force. We however provide the average and maximum values for the signal given by the tactile sensors. This signal corresponds to the sum of all taxels readings for each patch.

In addition to the number of contacts made with the object, we also compute two grasping metrics based on the Grasp Wrench Space (GWS) [34]: the largest-minimum resisted wrench (or largest ball, or ϵ quality metric), and the volume of the GWS. These metrics describe what external wrenches can be applied to the object without loosing stability. Thus, the higher, the better. While the ϵ metric considers only the weakest direction, the volume of the GWS provides information about the global robustness of the grasp. Since we expect our method to make more contacts around the grasped object, we also expect a better performance on the grasp metrics.



(a) Cylinder (b) Object 1 (c) Drill

Figure 15: **Exp 4**: Models of the grasped objects in simulation

5.2.1. *Simulation*

Setup. The perturbations on the objects are applied in the simulation by changing their position in the simulation environment. To avoid discontinuities, the position of the object is defined by attaching a virtual spring and damper to it (i.e. cartesian impedance with high stiffness). Its position and orientation are changed by moving the reference pose.

The selected objects are presented in Fig 15: we start with a simple cylinder, then an artificial more complex shape – very non-convex – composed by cylinders and spheres, and a drill. The grasping of *Object 1* can be seen in Fig 16.

Results. Fig 18 details the number of contact points for each control method and object. These results are summed up in Table 4 with additional data about the contact forces and grasp metrics. Our algorithm allows to create a high number of contacts with the object compared to the other two controllers. On average, about 9 contacts are made (i.e. a little more than 2 per finger out of 3 possible contacts for the 3 separate links), while only 5 to 6 for the other methods (a little more than 1 contact on each finger).

If we go more into details in the transition from Exp 4a to Exp 4b, our algorithm keeps about the same number of contacts (8.73 vs. 8.67), whereas *Enclose1* (6.00 vs. 5.75) and especially *Enclose2* (5.67 vs. 4.92) loose a lot of contact points. This is expected as Exp 4b is about adapting to perturbations after the grasp, which the other algorithms cannot do properly. The results are similar for the volume of GWS: our algorithm outperforms the other approaches, by providing a larger volume in both sets of experiments, and the difference increases in Exp 4b. The ϵ metric also follows the same trend.

The contact forces are pretty similar for each algorithm in Exp 4a ($< 1N$), whereas for Exp 4b, the other methods based on position control do not adapt to the perturbations and hence create high contact forces.

5.2.2. *On the robot*

Setup. We equipped the 7 DOFs Kuka LWR Robot with a 16-DOFs AllegroHand, controlled at 300 Hz using open-loop torques, and partially covered with Tekscan tactile sensors on the inside surface of the phalanxes, see Fig 17(d). The *AllegroHand* has 4 fingers with 4 DOFs each. The sensors come in patches of 4 by 3-15 taxels

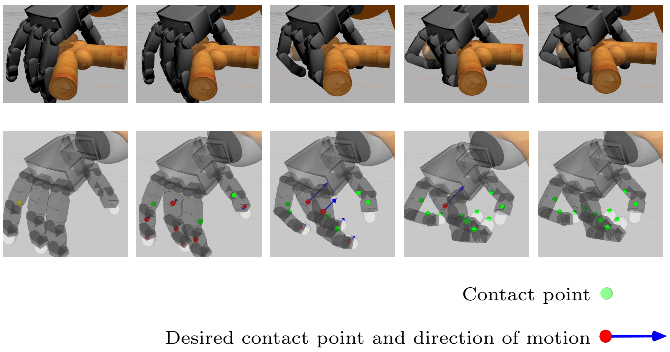
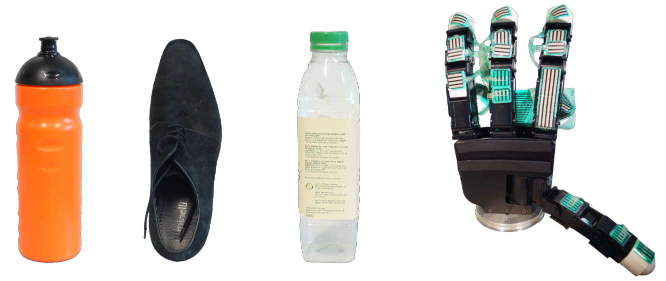
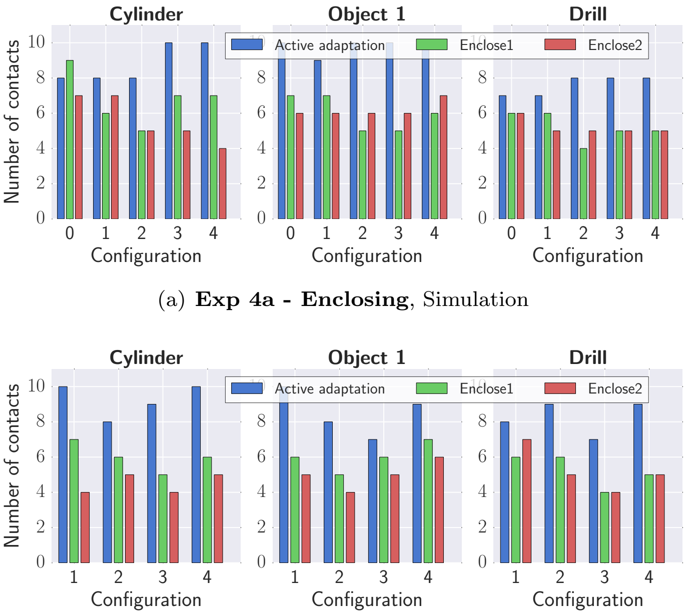


Figure 16: **Exp 4a**: Progression of the enclosure of *Object 1* with the active compliance algorithm in gazebo (top) and rviz (bottom).



(a) Bottle (b) Shoe (c) Square bottle (d) AllegroHand and sensors

Figure 17: (abc) **Exp 4**: Models of the grasped objects by the real robot. (d) The AllegroHand with fingers covered with Tekscan tactile sensors. Each patch is a matrix of 4*3 or 4*4 taxels (4*15 for the base of the index finger)



(a) **Exp 4a - Enclosing**, Simulation

(b) **Exp 4b - Perturbation**, Simulation

Figure 18: **Exp 4a-b, Simulation**: number of contacts for each position/orientation configuration

Table 4: **Exp 4 simulation**: Results

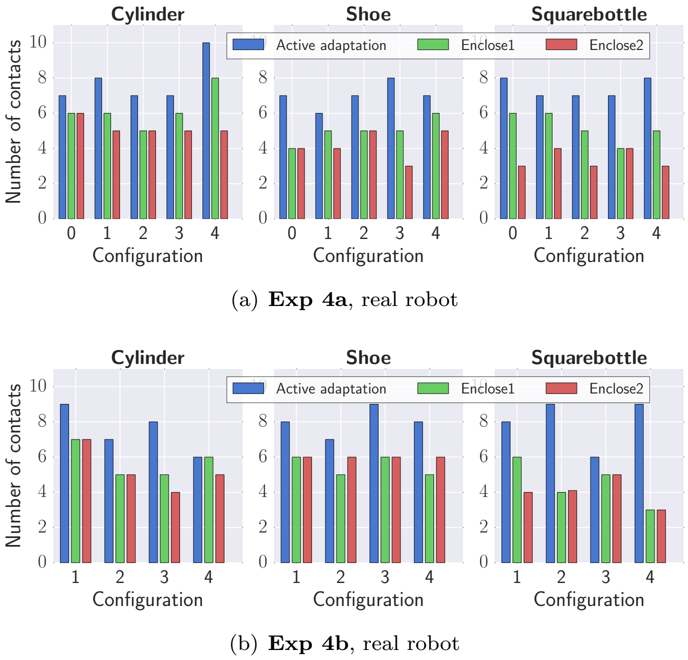
Exp 4a: Enclose	**Active adaptation**	**Enclose1**	**Enclose2**
Nb. contacts	8.73 ± 1.12	6.00 ± 1.21	5.67 ± 0.87
Max force (N)	0.92 ± 0.32	0.93 ± 0.58	0.72 ± 0.31
Average force (N)	0.55 ± 0.06	0.62 ± 0.56	0.52 ± 0.27
GWS volume	1.35 ± 0.45	0.95 ± 0.40	0.82 ± 0.52
GWS ϵ metric	0.13 ± 0.02	0.12 ± 0.03	0.13 ± 0.05
Exp 4b: Perturb.	**Active adaptation**	**Enclose1**	**Enclose2**
Nb. contacts	8.67 ± 1.03	5.75 ± 0.83	4.92 ± 0.86
Max force (N)	0.93 ± 0.31	2.13 ± 1.30	1.65 ± 0.86
Average force (N)	0.54 ± 0.07	1.16 ± 0.67	1.21 ± 0.58
GWS volume	1.31 ± 0.35	0.64 ± 0.40	0.51 ± 0.33
GWS ϵ metric	0.13 ± 0.03	0.10 ± 0.04	0.10 ± 0.02

which are designed to fit the human hand, but we adapted their configuration for this particular hand. The density of taxels allows to determine the position of contact in addition to an estimate of the pressure. Knowing the geometry of the fingers, we are also able to determine the normal of the contact. For each link, when the summed response is above a noise threshold, we define one contact point as the weighted average of all taxels readings on that link.

On the real robot, the perturbations are applied by giving the inverse perturbation command to the robot (the arm moves instead of the object). The chosen objects are presented in Fig 17(abc): we start with a cylindrical bottle, we also grasp a soft shoe, which is easily deformable, and a plastic bottle with a square section.

Results. The number of contact points is detailed in Fig 19 and summed up in Table 5 with the other measures and quality metrics. The number of contacts is again higher with our algorithms. On average, about 7 to 8 contacts are made, while only 4 to 6 for the other methods (a little more than 1 contact on each finger). The results for Enclose2 are still slightly worse than for Enclose1, as predicted.

On the real robot, our algorithm performs this time better during the perturbations (Exp 4b, 7.8 contacts) than the simple grasping (Exp 4a, 7.4 contacts). This can be explained by the effect of the perturbations helping the fingers slide on the surface of the object and thus creating more contacts. In simulation, this behavior relying on friction may not have been properly simulated. For *Enclose2*, the number of contacts actually increases with the perturbations (5.1 vs. 4.3 contacts). This is due to the fact that deformable objects can naturally comply with a non compliant controller and create more contacts, at the expense of high contact forces. Indeed, the set of objects is here more compliant than in simulation, especially the shoe. These results are also reflected in the grasp metrics, with the active adaption creating more robust grasps than the other methods. The tactile signal values are similar in range for all algorithms, with higher values during the perturbations – going from about 12 during enclosing to 20 (no unit). Similarly as in the simulation, it is expected that the values increase during the perturbations, and it seems that the tactile readings increase for the active adaptation algorithm too, probably due to friction in



(a) **Exp 4a**, real robot

(b) **Exp 4b**, real robot

Figure 19: **Exp 4a-b, real robot**: number of contacts for each position/orientation configuration (Exp 4a: enclosing and Exp 4b: perturbation)

Table 5: **Exp 4 real robot**: Results

Exp 4a: Enclose	**Active adaptation**	**Enclose1**	**Enclose2**
Nb. contacts	7.40 ± 0.88	5.47 ± 0.96	4.27 ± 0.93
Maximum *signal*	12.8 ± 6.6	13.0 ± 6.3	11.4 ± 7.4
Average *signal*	6.2 ± 3.6	7.2 ± 3.1	7.0 ± 3.3
GWS volume	1.4 ± 0.6	0.6 ± 0.5	0.4 ± 0.5
GWS ϵ metric	0.13 ± 0.05	0.10 ± 0.05	0.05 ± 0.05
Exp 4b: Perturb.	**Active adaptation**	**Enclose1**	**Enclose2**
Nb. contacts	7.83 ± 1.07	5.25 ± 1.01	5.09 ± 1.11
Maximum *signal*	19.9 ± 7.7	20.8 ± 14.7	18.3 ± 12.0
Average *signal*	9.3 ± 3.8	9.6 ± 5.9	9.0 ± 5.6
GWS volume	1.2 ± 0.5	0.7 ± 0.4	0.7 ± 0.6
GWS ϵ metric	0.14 ± 0.05	0.09 ± 0.05	0.08 ± 0.06

the joints and with the object. However, these values from the tactile sensors cannot be precisely translated into contact forces: it is not possible to decouple the intensity of the signal, the area in contact and the material in contact.

6. Discussion

In this paper, we presented a method to actively comply with unknown surfaces with a multi-fingered robot equipped with tactile sensors. This method has applications both in haptic exploration and in grasping. To our knowledge, this is the first demonstration of active compliance between a complex system such as a robotic arm and hand, and unknown surfaces, by keeping and creating desired new contacts using tactile information. Our method allows to create and maintain contacts at desired positions on the robot while having unilateral constraints on undesired contacts, in the prioritized tasks framework. While the high priority tasks take care of the interaction forces and contact constraints, the lower priority tasks allow to increase the contact area and to drive the exploration motion. Contacts occurring on parts of the robot that are not desired do not disturb the exploration nor create undesired forces thanks to the modified nullspace control. We demonstrated the possibility to actively explore around arbitrary shapes with a simulated robot arm and hand. This is useful in the context of search, particularly for occluded areas, by only providing approximate positions for the robot to explore. The robot can then manage to move around the surface creating and loosing contacts while keeping low contact forces.

In the current implementation of the exploration strategy, there are situations when the robot can get stuck in local minima. We did not tackle here the high-level planning as it is not purpose of this work. Simple approaches based on information gain, coupled with detection of local minima would probably be enough to further automatize the exploration process.

The algorithm does not currently handle several desired contact points on one link. This could be useful for large areas on one link (for instance the palm of the hand) that could host several contact points simultaneously. Currently, if there is already an existing desired contact point on a link, it is not possible to deliberately increase the number of contacts points on that link. This would involve classifying whether each existing contact corresponds to a particular desired contact point.

One particularity of the high-DOFs platforms such as robotic hands is that they can take many different configurations during the exploration, some of which are not optimal to maximize the area in contact. For instance, simultaneous contact on the back of one finger and the front of another finger while exploring a flat area. However, this is an advantage for the exploration of certain shapes, for instance the inside of a cup in which some fingers make contact with one side while other stick to the other side. It also allows to hold two objects at the same time between the fingers, see Fig 20[9].

We also demonstrated the ability of this algorithm to comply to arbitrary shapes with an application to grasping. While a lot of the grasp planning research does not consider in detail the actual control strategy, uncertainties make precise grasp planning less relevant on the execution side. Our controller resulted in more contact points and provided more stable grasps than other uninformed enclosing algorithms. It could be a possible solution to implement planned grasps on actual robotic platforms.

[9] For holding two objects, the closest point of contact used to compute the velocity of a desired point is valid only if its normal is opposite to the direction from the desired point to this point: $n^i \cdot (p_c^i - p^j) < 0$ as a condition to Eq (18).
Video of the experiments: `http://lasa.epfl.ch/videos/downloads/sommer_he.mp4`

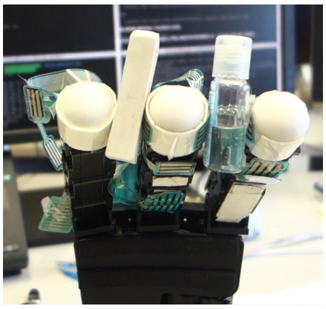


Figure 20: Additional illustration of use of the algorithm. The fingers hold two objects between them.

Appendix A. Details of the control for Exp 1

Median of normal of contact and attached frame. The median normal of contact $n_m \in \mathbb{R}^3$ corresponds to the average of the two most distant normals of contact between the robot and the surface. It is used both for determining the desired orientation of the hand during the exploration and the allowed plan of motion to reach the final Cartesian target (detailed in the next paragraphs).

$$n_m = \begin{cases} \frac{n_i+n_j}{2} & \text{if } n_i + n_j \neq 0 \\ n_i & \text{else.} \end{cases} \tag{A.1}$$

where $(i,j) = \text{argmax}_{(i,j)\in n_c}\{acos(n_i, n_j)\}$ are the indices of the two contacts which have the most different normals.

This is useful, because taking only the average of all the contact normals would give little weight to outliers, which are very important as they represent crucial information about the surface's profile.

Orientation reference of the impedance controller. We create a rotational frame R_{n_m,r_x}, using the above normal direction of contact n_m and the orthogonal projection of the hand's proximo-distal direction (palm towards fingers) r_x on n_m:

$$r'_x = r_x - (r_x \cdot n_m)n_m \tag{A.2}$$

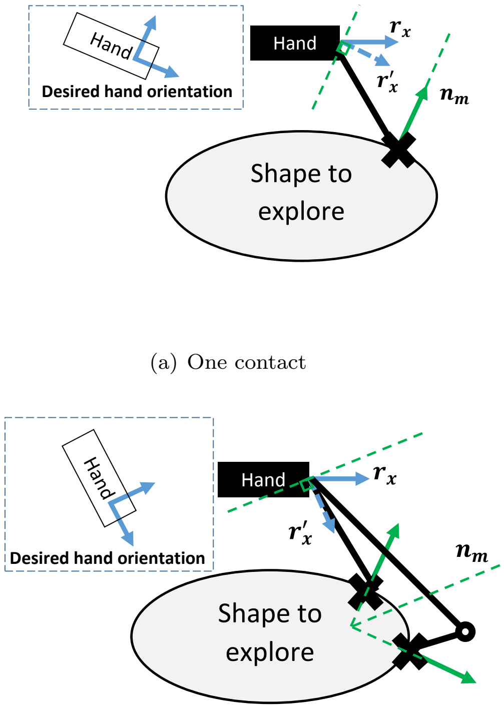
$$R_{n_m,r'_x} = \begin{bmatrix} n_m, & r'_x, & n_m \times r'_x \end{bmatrix} \tag{A.3}$$

This ensures that the palm of the hand stays perpendicular to the contact normal.

Position reference of the impedance controller. The reference position $p_r \in \mathbb{R}^3$ of the impedance controller is computed from the desired final position of the motion $p_f \in \mathbb{R}^3$, given by a higher-level controller, the current position $p \in \mathbb{R}^3$, and the computed median normal of contact n_m. The error p_e between the current and final position is projected on a plane normal to n_m in order to create a motion tangential to the surface:

$$p_e = p_f - p \tag{A.4}$$

$$p'_e = p'_e - (p'_e \cdot n_m) \cdot n_m \tag{A.5}$$



(a) One contact

(b) Two contacts

Figure A.21: Schematic of the computation of the reference hand orientation for the impedance controller. In the case of one contact, the normal of the contact is chosen as a reference for the desired hand orientation. For two or more contacts, the average of the two most distant normals is chosen.

The reference position is then proportional with gain $G \in \mathbb{R}^+$ to the projected error, and saturated if that distance is bigger than a scalar threshold $d \in \mathbb{R}^+$.

$$p_r = \begin{cases} p + \frac{p'_e}{\|p'_e\|} \cdot G & \text{if } \parallel p'_e \parallel > d \\ p + p'_e \cdot \frac{G}{d} & \text{else.} \end{cases} \tag{A.6}$$

Impedance control. Because the robot operates in contact with its environment, a compliant controller provides a safe way to interact with the areas in contact.

Given the reference and actual positions $p_r, p \in \mathbb{R}^3$ and orientations $R_r, R \in \mathbb{R}^{3\times3}$ of the end effector (here defined at the base of the middle-finger), we define the Cartesian error term as:

$$x_e = \begin{bmatrix} p_e \\ \Psi \end{bmatrix}, \ \Psi = \text{angleaxis}(R^T R_r) \tag{A.7}$$

where angleaxis(R^*) represents the angle-axis representation corresponding to a rotation matrix R.

The torques for the Cartesian impedance control task are computed by multiplication of the transposed Jacobian $J_e(q)$ with the Cartesian feedback control forces:

$$\tau_e = J_e^T(Kx_e + D\dot{x}_e) \tag{A.8}$$

The stiffness and damping matrices $K, D \in \mathbb{R}^{6\times6}$ are symmetric positive definite:

$$K = \begin{bmatrix} K_p & 0 \\ 0 & K_r \end{bmatrix}, \; D = \begin{bmatrix} D_p & 0 \\ 0 & D_r \end{bmatrix} \tag{A.9}$$

where $K_p, K_r \in \mathbb{R}^{3\times3}$ and $D_p, D_r \in \mathbb{R}^{3\times3}$ are sub-matrices respectively relating forces to positional errors, torques to rotational errors, forces to positional velocity and torques to rotational velocities.

Definition of the stiffness and damping matrices. During the exploration, the purpose of the impedance control is to drive the motion of the robotic hand, not to ensure contact with the surface. For this reason, the stiffness matrices are defined in the rotational frame R attached to the end-effector, as:

$$K'_p = \begin{bmatrix} 0 & 0 & 0 \\ 0 & k_p & 0 \\ 0 & 0 & k_p \end{bmatrix}, \qquad K'_r = \begin{bmatrix} 0 & 0 & 0 \\ 0 & k_r & 0 \\ 0 & 0 & k_r \end{bmatrix} \tag{A.10}$$

where the first axis corresponds to the dorso-palmar direction, and the second axis to the proximo-distal direction. In the robot's frame, we use the rotated stiffness matrices $K_p = RK'_pR^T$ and $K_r = RK'_rR^T$. The rotational stiffness value is zero in the dorso-palmar direction as the orientation of the hand along that axis is not crucial for the exploration and this releases a degree of freedom and allows more dexterous motions.

Both positional and rotational damping matrices are isotropic[10]:

$$D_r = d_r \cdot I_{3\times3}, \qquad D_p = d_p \cdot I_{3\times3} \qquad \text{with } d_r, d_p \in \mathbb{R}. \tag{A.11}$$

Reference position. Thanks to our algorithm, the reference positions described as p_f in Appendix A do not need to lie on the surface since the controller navigates to the closest point on the surface. Therefore, they can be randomly distributed around the estimated position of the area to explore. We define a list of these reference positions spread around the object to explore. When the end-effector reaches within a threshold of the orthogonal projection of the current reference position on the surface's estimated tangential plane, the next reference position in the list is tracked. This way, the end-effector goes sequentially through all the positions in the list. There are more informed ways to choose the reference positions, for instance using entropy and information gain about the surface being reconstructed, but the target of this experiment is to demonstrate the possibility to be in contact with an unknown surface and to navigate smoothly around it, not the search process itself.

[10] We used $k_p = 300N.m^{-1}$, $d_p = 300N.s.m^{-1}$, $k_r = 10N.rad^{-1}$, $d_r = 3N.s.rad^{-1}$

Acknowledgment

This research was supported by the Swiss National Science Foundation through the National Centre of Competence in Research (NCCR) Robotics.

[1] A. Okamura, M. Cutkosky, Haptic exploration of fine surface features, in: 1999 IEEE International Conference on Robotics and Automation, 1999. Proceedings, Vol. 4, 1999, pp. 2930–2936 vol.4. doi:10.1109/ROBOT.1999.774042.

[2] G. Heidemann, M. Schopfer, Dynamic tactile sensing for object identification, in: Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on, Vol. 1, 2004, pp. 813 – 818 Vol.1.

[3] M. Meier, M. Schopfer, R. Haschke, H. Ritter, A Probabilistic Approach to Tactile Shape Reconstruction, IEEE Transactions on Robotics 27 (3) (2011) 630–635. doi:10.1109/TRO.2011.2120830.

[4] R. Ibrayev, Y.-B. Jia, Surface Recognition by Registering Data Curves from Touch, in: Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on, 2006, pp. 55 –60.

[5] J. A. Fishel, G. E. Loeb, Bayesian exploration for intelligent identification of textures, Frontiers in Neurorobotics 6 (2012) 4. doi:10.3389/fnbot.2012.00004.
URL http://journal.frontiersin.org/article/10.3389/fnbot.2012.00004/full

[6] G. Ganesh, N. Jarrasse, S. Haddadin, A. Albu-Schaeffer, E. Burdet, A versatile biomimetic controller for contact tooling and haptic exploration, in: Robotics and Automation (ICRA), 2012 IEEE International Conference on, 2012, pp. 3329–3334. doi:10.1109/ICRA.2012.6225057.

[7] R. S. Jamisola Jr, P. Kormushev, A. Bicchi, D. G. Caldwell, Haptic Exploration of Unknown Surfaces with Discontinuities.

[8] C. Rosales, A. Ajoudani, M. Gabiccini, A. Bicchi, Active gathering of frictional properties from objects, in: Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on, 2014, pp. 3982–3987. doi:10.1109/IROS.2014.6943122.

[9] A. Schneider, J. Sturm, C. Stachniss, M. Reisert, H. Burkhardt, W. Burgard, Object identification with tactile sensors using bag-of-features, in: Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on, 2009, pp. 243 – 248.

[10] S. J. Lederman, R. L. Klatzky, Hand movements: A window into haptic object recognition, Cognitive Psychology 19 (3) (1987) 342–368. doi:10.1016/0010-0285(87)90008-9.
URL http://www.sciencedirect.com/science/article/pii/0010028587900089

[11] P. Allen, P. Michelman, Acquisition and interpretation of 3-D sensor data from touch, IEEE Transactions on Robotics and Automation 6 (4) (1990) 397–404. doi:10.1109/70.59353.

[12] N. Sommer, A. Billard, Face classification using touch with a humanoid robot hand, in: 2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids), 2012, pp. 120–125. doi:10.1109/HUMANOIDS.2012.6651508.

[13] N. Sommer, M. Li, A. Billard, Bimanual compliant tactile exploration for grasping unknown objects, in: Robotics and Automation (ICRA), 2014 IEEE International Conference on, 2014, pp. 6400–6407. doi:10.1109/ICRA.2014.6907804.

[14] A. Bierbaum, M. Rambow, T. Asfour, R. Dillmann, A potential field approach to dexterous tactile exploration of unknown objects, in: Humanoid Robots, 2008. Humanoids 2008. 8th IEEE-RAS International Conference on, 2008, pp. 360–366.

[15] A. Jain, M. D. Killpack, A. Edsinger, C. C. Kemp, Reaching in clutter with whole-arm tactile sensing, The International Journal of Robotics Research 32 (4) (2013) 458–482. doi:10.1177/0278364912471865.
URL http://ijr.sagepub.com/content/32/4/458

[16] O. Khatib, A unified approach for motion and force control of robot manipulators: The operational space formulation, Robotics and Automation, IEEE Journal of 3 (1) (1987) 43–53.

[17] O. Khatib, L. Sentis, J. Park, J. Warren, Whole-body dynamic behavior and control of human-like robots, International Journal of Humanoid Robotics 1 (01) (2004) 29–43.

[18] J. Park, O. Khatib, Robot multiple contact control, Robotica 26 (05) (2008) 667–677. doi:10.1017/S0263574708004281.

[19] F. Flacco, A. De Luca, O. Khatib, Prioritized multi-task motion control of redundant robots under hard joint constraints, in: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2012, pp. 3970–3977. doi:10.1109/IROS.2012.6385619.

[20] N. Mansard, O. Stasse, P. Evrard, A. Kheddar, A versatile generalized inverted kinematics implementation for collaborative working humanoid robots: The stack of tasks, in: Advanced Robotics, 2009. ICAR 2009. International Conference on, IEEE, 2009, pp. 1–6.

[21] O. Kanoun, F. Lamiraux, P. B. Wieber, Kinematic Control of Redundant Manipulators: Generalizing the Task-Priority Framework to Inequality Task, IEEE Transactions on Robotics 27 (4) (2011) 785–792. doi:10.1109/TRO.2011.2142450.

[22] L. Saab, O. E. Ramos, F. Keith, N. Mansard, P. Souères, J. Y. Fourquet, Dynamic Whole-Body Motion Generation Under Rigid Contacts and Other Unilateral Constraints, IEEE Transactions on Robotics 29 (2) (2013) 346–362. doi:10.1109/TRO.2012.2234351.

[23] A. Escande, N. Mansard, P.-B. Wieber, Hierarchical quadratic programming: Fast online humanoid-robot motion generation, The International Journal of Robotics Research 33 (7) (2014) 1006–1028. doi:10.1177/0278364914521306.
URL http://ijr.sagepub.com/content/33/7/1006

[24] A. Bicchi, V. Kumar, Robotic grasping and contact: a review, in: IEEE International Conference on Robotics and Automation, 2000. Proceedings. ICRA '00, Vol. 1, 2000, pp. 348–353 vol.1. doi:10.1109/ROBOT.2000.844081.

[25] M. A. Roa, R. Suárez, Grasp quality measures: review and performance, Autonomous Robots 38 (1) (2014) 65–88. doi:10.1007/s10514-014-9402-3.
URL http://link.springer.com/article/10.1007/s10514-014-9402-3

[26] C. Goldfeder, P. K. Allen, Data-driven grasping, Autonomous Robots 31 (1) (2011) 1–20. doi:10.1007/s10514-011-9228-1.
URL http://link.springer.com/article/10.1007/s10514-011-9228-1

[27] J. Kim, K. Iwamoto, J. Kuffner, Y. Ota, N. Pollard, Physically Based Grasp Quality Evaluation Under Pose Uncertainty, IEEE Transactions on Robotics 29 (6) (2013) 1424–1439. doi:10.1109/TRO.2013.2273846.

[28] Z. Chen, T. Wimbock, M. Roa, B. Pleintinger, M. Neves, C. Ott, C. Borst, N. 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), 2015, pp. 4911–4918. doi:10.1109/ICRA.2015.7139881.

[29] M. Li, K. Hang, D. Kragic, A. Billard, Dexterous grasping under shape uncertainty, Robotics and Autonomous Systems 75, Part B (2016) 352–364. doi:10.1016/j.robot.2015.09.008.
URL http://www.sciencedirect.com/science/article/pii/S0921889015001967

[30] R. P. Jr, A. H. Fagg, R. A. Grupen, Null-Space Grasp Control: Theory and Experiments, IEEE Transactions on Robotics 26 (2) (2010) 282–295. doi:10.1109/TRO.2010.2042754.

[31] M. Catalano, G. Grioli, A. Serio, E. Farnioli, C. Piazza, A. Bicchi, Adaptive synergies for a humanoid robot hand, in: 2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids), 2012, pp. 7–14. doi:10.1109/HUMANOIDS.2012.6651492.

[32] M. Ciocarlie, C. Goldfeder, P. Allen, Dexterous Grasping via Eigengrasps: A Low-Dimensional Approach to a High-Complexity Problem, Robotics: Science and Systems - Robot Manipulation: Sensing and Adapting to the Real World.
URL http://www.coreygoldfeder.com/papers/RSS07.pdf

[33] R. de Souza, S. El-Khoury, J. Santos-Victor, A. Billard, Recognizing the grasp intention from human demonstration, Robotics and Autonomous Systems 74, Part A (2015) 108–121. doi:10.1016/j.robot.2015.07.006.
URL http://www.sciencedirect.com/science/article/pii/S0921889015001505

[34] N. Pollard, Synthesizing grasps from generalized prototypes, in: , 1996 IEEE International Conference on Robotics and Automation, 1996. Proceedings, Vol. 3, 1996, pp. 2124–2130 vol.3. doi:10.1109/ROBOT.1996.506184.