Model of Tactile Sensors Using Soft Contacts and its Application in Robot Grasping Simulation

Sami Moisio[a], Beatriz León[b,*], Pasi Korkealaakso[a], Antonio Morales[b]

[a] *Laboratory of Intelligent Machines at the Centre of Computational Engineering and Integrated Design (CEID), Lappeenranta University of Technology, P.O. Box 20, 53851 Lappeenranta, Finland*

[b] *Robotic Intelligence Laboratory at the Department of Computer Science and Engineering, Universitat Jaume I, 12006 Castellón, Spain*

Abstract

In the context of robot grasping and manipulation, realistic dynamic simulation requires accurate modeling of contacts between bodies and, in a practical level, accurate simulation of touch sensors. This paper addresses the problem of creating a simulation of a tactile sensor as well as its implementation in a simulation environment. The simulated tactile sensor model utilizes collision detection and response methods using soft contacts as well as a full friction description. The tactile element is created based on a geometry enabling the creation of a variety of different shape tactile sensors. The tactile sensor element can be used to detect touch against triangularized geometries. This independence in shape enables the use of the sensor model for various applications, ranging from regular tactile sensors to more complex geometries as the human hand which makes it possible to explore human-like touch. The developed tactile sensor model is implemented within OpenGRASP and is available in the open-source plugin. The model has been validated through several experiments ranging from physical properties verification to testing on robot grasping applications. This simulated sensor can provide researchers with a valuable tool for robotic grasping research, especially in cases where the real sensors are not accurate enough yet.

Keywords: Tactile sensing, Dynamic simulation, Robot Grasping, Soft

[*] Corresponding author: Universitat Jaume I, Campus Riu Sec, Avd. Sos Baynat S/N. E-12071 Castellon, Spain. Tel: +34964728306, Fax: +34964728486

Email address: beatriz.leon@uji.es (Beatriz León)

contacts

1. Introduction

Robotic manipulators have been used extensively for many years in both research and commercial applications. Over the years, the change from structured to unstructured environments has made the development of different sensors a priority to enable robots to cope with significant uncertainties.

Touch, combined with vision, are the main senses that allow humans to perform dexterous manipulation. For this reason, sensors that can retrieve tactile information have been developed in order to equip robot hands with such a sense.

Tactile sensors are defined as devices that can measure different properties of an object through physical contact between the sensor and the object [1]. They can measure mechanical properties including pressure, normal and shear forces, torques, slip and vibrations, or other properties like temperature or moisture. In robotic manipulation, only the mechanical properties of the contact are studied, typically sensing normal forces and contact positions.

Different methods exist for constructing tactile sensors (for a review see [2, 3]). There are tactile sensors based on various principles such as resistive, capacitive, optical, ultrasonic, magnetic or piezo-eletric sensors.

The performance of the real tactile sensors developed until now is far from human sensing capabilities. Nevertheless, they have been used in robot manipulation in the last few years for different purposes including reactive robot control, collision detection and object recognition.

In reactive control, the robot has to cope with the inaccuracy of the vision systems when working in unstructured environments [4, 5, 6]. Recent studies have also shown how tactile-sensing-based algorithms can be employed to detect and react to contacts encountered during the execution of a grasp [7] as well as how tactile sensor information can be used to infer knowledge about grasp stability [8].

In the field of object recognition, tactile sensors are used to explore the 3D shape of unknown objects and use their feedback to create or improve the object's model. Using the tactile sensor matrix, a small imprint of the object can be taken and used to recognize surface features [9].

These various applications of tactile sensors show the importance of their use in robot manipulation. In this area, simulation is a major tool used to support research, adding both flexibility and reproducibility to the experiments. Having a tactile sensor model that enables tactile sensing simulation and, more generally, complete simulation of robot grasping, will be of great benefit to the robotics community.

Simulation of tactile sensors for robot grasping is a fairly new field of research. Using the available simulation environments and physics engines, there has been some proposed models, for example [10, 11, 12]. However, all the presented models have weaknesses. In most of the cases, simulations are solved using rigid contacts which can not model the deformation of a compliant fingertip or holding torque around the contact. When a real collision occurs, both bodies deform to create a unified collision surface. This is not modelled when dealing with rigid bodies. Solving the deformation of the bodies is currently too time consuming to calculate in real-time. Therefore, the soft contact approach has been used where the local deformation is taken into account by allowing the objects to penetrate each other. It enables the creation of a collision surface without modeling the actual deformation of the bodies. A proper collision surface offers possibilities for modeling holding torque around the colliding surface as well as making the collisions more realistic in comparison to non-penetrating point contacts. Additionally, friction has been modeled using friction models that do not include the stick-slip phenomenon which causes errors at near zero sliding velocities. This leads to the need for a soft contact model with the ability to form a contact surface as well as being able to introduce holding torque.

This study presents the development of a simulated tactile sensor array with some of the same physical properties as a real tactile sensor array has, such as compressibility and friction. In order to create a model of the sensor dynamics three different areas were addressed: tactile sensor model construction, modeling soft contacts and friction modeling. The tactile sensor array (referred to as “tactile sensor” in the rest of the paper) is based on a geometry patch enabling the creation of various shape tactile sensors. A contact force model was created that enables the calculation of surface forces as well as the holding torque around the contact surface and the stick-slip phenomenon.

The model does not include the load spreading to adjacent texels generated by the material thickness [13]. Also manufacturing and mechanical imperfections are ignored at this point.

The proposed sensor model was implemented in a simulation environment as a plugin for OpenRAVE [14] and it is available open source (see Section 4 for more details).

In order to test the proposed model, different experiments were conducted. First, the physical properties of the simulated tactile sensor were validated using static and dynamic tests (sections 5.1 and 5.2). Also an experiment on simulated robot grasping vs. real robot grasping was carried out by a robot hand grasping an object and by the corresponding model on the simulator performing the same actions (Section 5.2.2). Finally, the sensor model potential was explored by applying it to modeling human touch (Section 5.3).

2. Previous Work

This section first presents a review of the contact models used in simulation, explaining their advantages and disadvantages followed by the relevant work on simulation of robot grasping using tactile sensors.

2.1. Contact Models

Contact models can be divided into three different categories: impulse, analytical and penalty methods. Rigid body assumption for collisions is used in the analytical [15] and in the impulse methods [16, 17] while continuous contact models are used in the penalty methods [18, 19, 20]. In this context rigid body assumption means non-penetrative or colliding contact in which the exact impact moment is solved after which the surfaces are prevented from penetrating each other.

In the impulse based approach contact between bodies is considered as a collision at a specific point in time and without needing to solve the contact forces. Instead the change in the objects' velocities is applied directly to the bodies over one time-step. The method is fast and easy to implement but a problem arises with steady contacts in static configurations. Analytical methods are based on the use of constraints to handle contacts. In contrast to impulse-based methods, these methods are stable in steady contacts, however, due to simultaneous solving of all contacts, they are also computationally expensive. Penalty methods are called penetrative or soft contacts (also non-colliding contacts) because they allow for small penetrations in the colliding objects. Consequently, contact forces are obtained using temporal nonlinear spring-damper elements at the contact point. Based on the elasticity of the bodies in contact, the parameters of the spring-damper element can be defined using the Herzian contact theory [21]. Brogoliato et al. [22] presented a comprehensive survey of these methods and recently Drumwright and Shell [23] performed an evaluation of some of them quantifying the performance with respect to robustness and speed.

Analytical and impulse methods give accurate descriptions for contacts and are often used when no interpenetrations are allowed between the contacting bodies. These methods also allow longer time-steps compared to penalty methods with stiff springs. However these methods lead to complicated equations especially in the case of multiple contact points and contacts with friction. Furthermore, in the case of mechatronic machines such as robots, the machine dynamics require the use of small time steps making penalty methods more suitable, especially for real time applications. It is also important to note that rigid body assumptions do not take into account small deformations during collisions, instead they occur instantaneously changing the velocities. For this reason, continuous contact models give more accurate results in applications where deformation plays an important role. Such cases are for example, calculating surface pressure and holding torque around the contact patch.

Penalty methods suffer from certain instability and parameter sensitivity as the system is very stiff. This does not imply a major problem provided that the simulation parameters and the stability limits are considered. There are different methods for determining the stability limits of stiff systems for penalty methods. Shinya et al. [24] introduced a method for determining the allowable parameters (such as the maximum allowable time-step) in order to enable the simulation to remain stable. Such calculations to determine the correct parameters make the stability of the system more determinable and reduce the need for guesswork, therefore they are highly recommended.

One of the advantages in using penalty methods is the straightforward applicability for solving surface pressures from contacts due to the fact that the objects are allowed to form a real contact surface. In non-penetrative contacts the surface has to be formed using guesses or assumptions since the objects are not allowed to penetrate. This in turn complicates various calculations such as holding torque around the contact area. For these reasons a penalty method was used in the sensor model developed in this paper (see Section 3).

At each contact, the evaluation of friction plays an important role in grasping [25]. It is a very complex phenomenon and different models exist to evaluate it (see [26] for a review). There are multiple aspects that affect the amount of friction between the two objects ranging from lubrication to surface roughness. In specific cases and under laboratory conditions, the friction properties of a certain collision could be acquired. However, with exception of these special conditions, it is nearly impossible to verify and measure the frictional properties of naturally occurring collisions. On the other hand, the number of effecting factors is usually unknown and have to be generalized in the simulation case. Some verification measurements can be determined using simplified simulation models but the circumstances always change for a general case. This means that even a calibration through a simplified model will result only in a base line definition of the parameters. The friction models can be divided into two categories: Static friction models and dynamic friction models. Static friction models do not have any memory of previous states where as the dynamic friction models have some knowledge of prior states, for example through integration. Dynamic friction models are a step further from the more primitive static friction models given that they can represent the friction in a more complete way. The static friction models suffer from discontinuities at near zero sliding velocities leading to creep. This is specially relevant for robotic grasping using force closure grasps with friction, where the relative velocity between the graspable object and the robot gripper is near zero throughout the duration of the grasp. Using static models, this will lead to oscillations which will cause the graspable object to slip away from the robot gripper. Using a dynamic friction model, these problems can be avoided. For our work, the LuGre model [27] was chosen. Comparative studies [28] show that the LuGre model is advantageous in comparison to the Bliman-Sorine model [29]. One of the biggest advantages is that it is of a lower order then the Bliman-Sorine model. The addition of a friction model to a penalty method is a fairly straight forward operation in comparison to analytical or impulse methods.

2.2. Grasping in Simulation using Tactile Sensors

In general, robot grasping simulations have traditionally been using kinematics instead of dynamics. This is due to the fact that robot dynamic simulation is a very challenging problem. The most common simulation method for robot grasping has been the analytical method (GraspIt! [30], ODE [31], Bullet [32], etc.).

Using the available simulation environments and physics engines, there are some existing models of tactile sensors. Tegin and Wikander [10, 33] presented a model using GraspIt! which provide contact and force feedback. Also, Scharfe et al. [34] have recently used virtual tactile sensors to simulate the the sensors included on the Shadow hand fingertips. However the problems mentioned by using the impulse methods to model dynamics limited the simulation. Also the use of rigid contact can not model the deformation of a compliant fingertip or holding torque around the contact.

Jorgensen et al. [12] presented a model for a tactile sensor array as part of RobWorkSim where they use an isotropic function to describe the deformation as a function of the distance from the point force. This method requires an equal distribution of point forces on the surface of the sensor. There has been some attempts to use penalty methods for rigid body simulations as [11, 35, 36], but they used only Coulomb friction which leads to errors at near zero sliding velocities.

Pezzementi et al. [37] used a point spread function to model the response of the tactile sensors and characterize them via robotic experiments. However the effect of friction is ignored as well as the actual sensor values.

In the following section, a tactile sensor model is proposed which overcome most of the weakness of the previous tactile sensor simulations.

3. Tactile Sensor Model

The purpose of this work was to make a simulation model of a tactile sensor, not just by emulating the function but by modeling the actual physical properties starting from the formation of an actual contact patch to including a full friction description.

Different tactile sensors are available with a variety of shapes. Rigid sensors range from the simple planar sensors to ones shaped to curve around a robot fingertip. Also some flexible sensor types are available which can be for example wrapped around a humanoid robot arm. The idea was to create a tactile sensor that could be adapted to model any shape. For this reason, the tactile sensor is created based on a geometry defined by a triangularized mesh, which can be obtained from a CAD model.

The tactile sensor model is designed to measure the object mechanical properties, such as contact forces and contact positions. Acquiring the force enacting on the tactile sensor from an existing simulation environment can be troublesome. For example, if the simulation environment uses a non-penetrating collision method the conversion from a collision force to tactile data leads to extensive assumptions. A non-penetrating collision method will not give reliable results on force build up as the objects are not allowed to interpenetrate. Even if the objects are allowed to interpenetrate, the collision will change the object velocity in a single time-step making the tactile data unreliable.

The solution proposed is to calculate the contact forces within the tactile sensor by using a soft contact method and then apply them to the bodies. This means disabling all other collisions between the two bodies making the tactile force the only force acting between the two bodies. That way the pressure detected by the sensor and the pressure applied to the body are identical. Also this will ensure that the tactile sensor can acquire all the related data for producing tactile information as it is all calculated within it.

The chosen method consists of using a contact patch with several texels in order to form the tactile sensor. Single texels are used in order to determine collisions against other objects and to calculate the resulting collision forces. This provides an accurate and fast solution for solving the collision equations. Currently the number of contact points is equal to the number of texels in the tactile sensor but in the future a single texel can be modified to include several contact points in order to further increase the resolution of the sensor.

The contact forces are calculated on each contact point and are used by the simulator to grasp the object and for the tactile sensor feedback.

In the following sections, details of the tactile sensor creation based on a geometry, calculation of contact forces, and finally parameterization of the tactile sensor are presented.

3.1. Geometry-based tactile sensor

The simulated tactile sensor can be formed based on a triangularized geometry. This was done so that differently shaped sensors could be easily defined.

In Fig. 1 different variations of a tactile sensor geometries are presented: a simple grid (1a), a spherical surface (1b) and a geometry modelling a human fingertip (1c). The arrows represent the normal directions of the different triangles.

The texels of the simulated tactile sensor are constructed using the vertices from the sensor geometry.

For example, in the case of a planar tactile sensor consisting of an array with 8 rows and 6 columns, one would draw a 7x5 grid having 8x6 vertices to represent the centers of the tactile cells (see Fig. 2).

For each vertex, the normalized sum of all normals of the triangles connected to it is calculated and used as a normal direction to the sensor element.

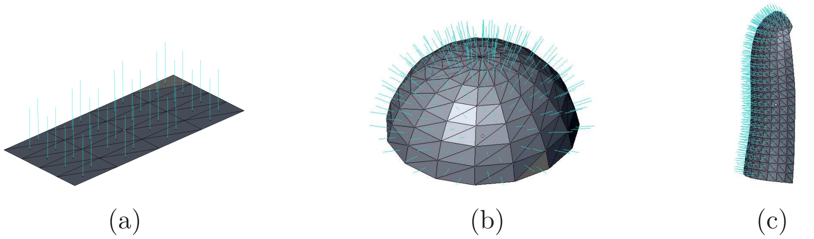

(a) (b) (c)

Figure 1: Example of tactile sensor geometries: (a) a simple grid (b) a spherical surface and (c) a human fingertip.

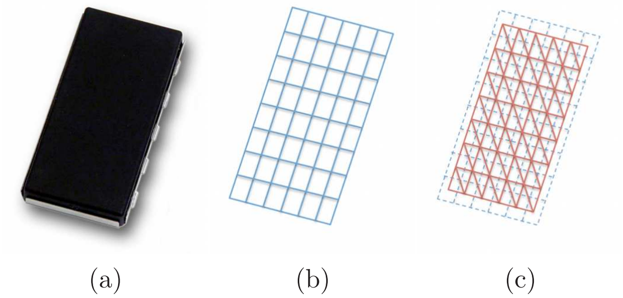

(a) (b) (c)

Figure 2: Example of a simulated tactile sensor construction: (a) real tactile sensor, (b) geometry of the sensor and (c) simulated tactile sensor elements.

The sensor element's maximum penetration needs to be defined in the sensor parameters. It is used to place the beginning of a vector pointing in the normal direction to the vertex (see Fig. 3). This vector in turn is used to calculate the intersection against all possible triangularized target geometries. The forces calculated at this point are explained in the following section.

3.2. Contact Force Model

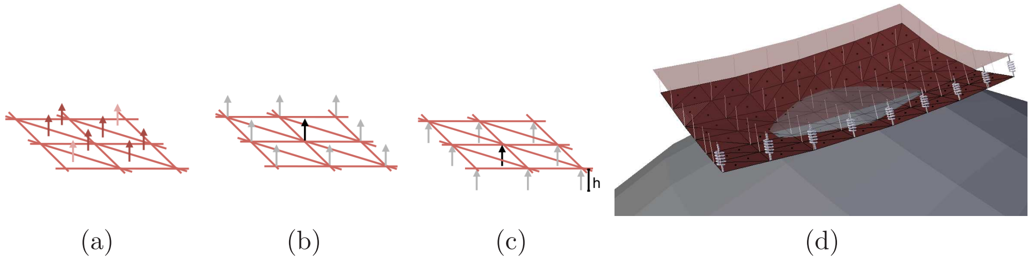
When a collision between the sensor and an object occurs, the contact information (position, relative velocity, penetration, etc.) is used to calculate the force in a single texel. For readers' convenience, a list of notations is given in Table 1.

The kinematics of the contact points between two bodies i and j can be described using knowledge of the geometries and states of the bodies (see Fig. 4).

The distance between contact points P_i and P_j can be written as follows:

Table 1: Notations

P_i	Contact point i
r^{P_i}	Position vector of contact i in the global reference frame
$\bar{u}^{P_i}$	Position vector of the contact i within the body reference frame.
R^i	Center position vector of body i
A^i	Rotation matrix from the body i reference frame to the global reference frame
s	Distance vector between contact points
$\dot{s}$	Relative velocity between contact points
$\dot{R}^i$	Velocity vector of body i
$\tilde{\omega}^i$	Skew-symmetric matrix of the body i angular velocity
s^T	Transposse vector of s
$n^{P_{ij}}$	Normal vector of the contact plane.
d	Distance vector in the direction of $n^{P_{ij}}$
$\dot{d}$	Velocity in the direction of $n^{P_{ij}}$
$\dot{s}_t$	Relative velocity in the tangential direction of $n^{P_{ij}}$
F_C	Contact force
F_n	Normal force produced by the soft contact
f_n	Magnitude of the normal force produced by the soft contact
F_t	Tangential force represented by friction
k	Spring coefficient
c_t	Tangential damping coefficient
c_n	Normal damping coefficient
z	Bristle deflection
σ_0	Stiffness coefficient of the contacting surfaces
σ_1	Friction damping coefficient
x_0	Stribeck velocity
μ_s	Static friction coefficient
μ_d	Dynamic friction coefficient
$\widetilde{A^i\bar{u}^{P_i}}$	Skew-symmetric matrix of body i
F_C^i	Contact Force applied to body i
$T_{F_C}^i$	Contact Torque applied to body i



(a) (b) (c) (d)

Figure 3: Determining collision vector for each vertex:(a) Triangle normals, (b) Normalized sum of triangles connected for each vertex (c) vertex vectors placed at the maximum penetration (h) and (d) overview of the sensor in touching an object and the contact area

$$s = r^{P_j} - r^{P_i} \tag{1}$$

where r^{P_i} and r^{P_j} are the position vectors of each contact point in the global reference frame. If R^i and R^j are defined as the center position vector of each body, the distance can be written as:

$$s = R^j + A^j \bar{u}^{P_j} - R^i - A^i \bar{u}^{P_i} \tag{2}$$

where A^i and A^j are rotation matrices from the body reference frame to the global reference frame and $\bar{u}^{P_i}$ and $\bar{u}^{P_j}$ are the position vectors of the contact points within the body reference frames.

In this case, P_i is equal to P_j then the distance is zero, but it is used to calculate the relative velocity between the contact points:

$$\dot{s} = \dot{R}^j + \tilde{\omega}^j A^j \bar{u}^{P_j} - \dot{R}^i - \tilde{\omega}^i A^i \bar{u}^{P_i} \tag{3}$$

where $\dot{R}^i$ and $\dot{R}^j$ are the velocity vectors of bodies i and j, and $\tilde{\omega}^i$ and $\tilde{\omega}^j$ are skew-symmetric matrices of the angular velocities.

By defining a contact plane between the bodies as the tangential plane to the normal of the graspable object's body surface at the contact point, the distance between contact points in this normal direction can be written as:

$$d = s^T n^{P_{ij}} \tag{4}$$

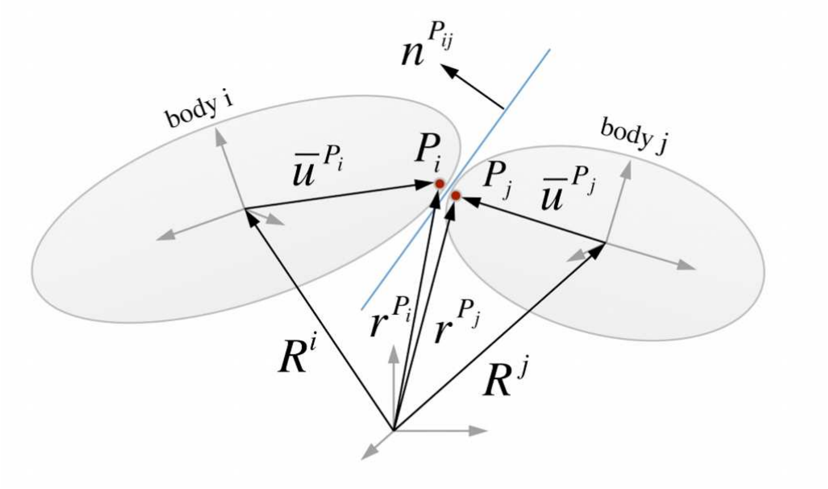


Figure 4: Contact between bodies i and j.

where s^T is the transpose vector of s and $n^{P_{ij}}$ is the normal vector of the contact plane. Accordingly, the velocity in the direction of the normal of the contact plane can be written as:

$$\dot{d} = \dot{s}^T n^{P_{ij}} \tag{5}$$

The relative velocity in the tangential direction of the contact plane can be obtained as follows:

$$\dot{s}_t = \dot{s} - \dot{d} n^{P_{ij}} \tag{6}$$

Contact forces are described using the soft contact approach which allows small penetration between contacting bodies taking into account local deformations. The amount of this penetration is calculated accordingly with the maximum penetration defined for each sensor.

On each contact point, the contact force (F_C) can be written as:

$$F_C = F_n + F_t \tag{7}$$

where F_n is the normal force produced by the soft contact and F_t is the tangential force represented by friction.

In its simplest form, the contact force in the normal direction of the plane (F_n) can be written as a linear spring-damper element:

$$F_n = -(kd + c_n\dot{d})n^{P_{ij}} = f_n n^{P_{ij}} \tag{8}$$

where k and c_n are spring and damping coefficients, respectively, and f_n is the magnitude of the normal force component.

The tangential friction forces can be evaluated using the LuGre friction model [27] which accounts for both static and sliding phenomena based on a bristle deflection interpretation. Accordingly, the LuGre model captures the dynamic behavior of the contact surface using the first order differential equation for bristle deflections, which can be written in vector form as follows:

$$\dot{z} = \dot{s}_t - \sigma_0 \frac{\mid \dot{s}_t \mid}{g(\dot{s}_t)} z \tag{9}$$

where z is bristle deflection and σ_0 is the stiffness coefficient of the contacting surfaces. In (9), $g(\dot{s}_t)$ is used to capture the Stribeck effect [38] in order to describe stick-slip phenomena, and can be calculated as follows:

$$g(\dot{s}_t) = \alpha_0 + \alpha_1^{-(\frac{\dot{s}_t^T \dot{s}_t}{\dot{x}_0^2})} \tag{10}$$

where x_0 is the Stribeck velocity and the parameters α_0 and α_1 are defined as follows:

$$\alpha_0 = F_n \mu_d \tag{11}$$

$$\alpha_1 = F_n(\mu_s - \mu_d) \tag{12}$$

where F_n is contact force in the direction of the normal of the contact surface, and μ_s and μ_d are the static and dynamic friction coefficients, respectively. Using state variables of friction and adding a viscous term, the friction force can be written as follows:

$$F_t = \sigma_0 z + \sigma_1 \dot{z} + c_t \dot{s}_t \tag{13}$$

where σ_1 is the friction damping coefficient and c_t is the tangential viscose damping. For bodies i and j, the resulting contact force can be applied as follows:

$$F_C^i = F_C \tag{14}$$

$$F_C^j = -F_C \tag{15}$$

Accordingly, the resulting torque of contact can be written as follows:

$$T^i_{F_C} = \widetilde{A^i \bar{u}^{P_i}} F^i_C \tag{16}$$

$$T^j_{F_C} = \widetilde{A^j \bar{u}^{P_j}} F^j_C \tag{17}$$

where $\widetilde{A^i \bar{u}^{P_i}}$ and $\widetilde{A^j \bar{u}^{P_j}}$ are skew-symmetric matrices.

Having these equations, the forces and torques can be calculated on each contact point. These forces are applied to the body where the tactile sensor is attached as well as to the body that the tactile sensor is colliding with. They are also used to retrieve sensor feedback information.

A summary of the steps needed to construct the simulated tactile sensor and to calculate the contact forces are outlined in Algorithm 1.

3.3. Determining the parameters for a real tactile sensor

In order to create a realistic model of a tactile sensor, the parameters needed to calculate the contact forces should be specified. These tactile sensor parameters can be divided into two different groups. The stiffness and damping coefficients for the normal force and the parameters for the tangential friction force. Both of these require the parameters to be determined independently.

In this section, we will show how to determine the parameters for a resistive tactile sensor. Weiss Robotics has developed piezo-resistive touch sensors which are made of materials whose resistance changes with force/pressure [39]. Given that these sensors have been used in several anthropomorphic robot hands [40, 4], and is the one available on our hardware setup we have chosen them as the example for parameterizing a tactile sensor, although our model is applicable to other types of sensors and as well.

This type of sensor has a foam on top of the tactile cells that compresses as the sensor is pressed against a surface. This tactile foam then changes the resistance over an electrical circuit and the pressure on the tactile can be deduced from the resistance value [41]. The foam is approximately linear in nature up to the point when it is fully compressed [39, 37]. At this point, the tactile sensor will saturate and the measurement will not be applicable. The foam used in the example is used at setup used in the University Jaume I in an experiment to increase the measuring range of the sensors and it is therefore a lot softer then the typical foam used for example in the Weiss sensor.

Algorithm 1: How to construct a simulated tactile sensor and obtain the sensor data

```
Initialization:
Using a mesh representing the real sensor geometry, create the simulated sensor texels;
Parametrize the sensor elements with the maximum penetration and the values needed to calculate the normal force ($k$, $c_n$) and friction ($\mu_s$, $\mu_d$, $\sigma_0$, $\sigma_1$,$c_t$);
Result: Tactile Sensor Readings
begin
    for each vertex of the simulated sensor element do
        Place a vector pointing to it, with a magnitude equal to the maximum penetration and with a direction equal to the sum of normals of the triangles connected to it;
    for each time-step do
        for each vertex do
            Calculate the intersection of the vector with the target objects;
            if they are in collision then
                Create a contact point on the intersection;
        for each contact point do
            Calculate the contact force in the normal direction (8);
            Calculate the tangential friction (13) ;
            Add these two components to get the total contact force.;
        for the sensor's body and each target body do
            Calculate and apply the forces (14,13) and torques (16, 17);
        Convert the forces to tactile values;
end
```

3.3.1. Stiffness coefficient (k)

The normal force parameters can be determined by measuring the stiffness properties of the foam covering the tactile sensor. In [39] it is shown that the normal directional stiffness of the sensor is approximately linear in nature. Therefore, the linear behaviour of the simulation model normal force is valid.

By placing a load on a piece of foam resulting in pressure p_l the compression of the foam x_c can be measured as a function of pressure. This relation of pressure and compression can then be used to determine the stiffness k of the tactile element as follows.

$$k = \ p_l * A_t/(n_{tex} * x_c) \tag{18}$$

where A_t is the area of the tactile, n_{tex} is the number of texels.

With a different resolution sensor, the stiffness changes and the stiffness coefficient has to be determined again. An example of defining the parameter for a tactile unit is shown in Section 3.4.

3.3.2. Damping coefficient (c)

The damping behaviour of the foam is difficult to determine and has not been investigated as it is negligible for the tactile readings when working at low velocities. Tactile sensors are not meant for bouncing objects off of them and therefore the operating velocities are low. In the simulation case the damping ratio is adjusted in order to keep the simulation stable. Through simulator testing it has been determined that the damping coefficient is approximately 1% of the stiffness coefficient.

3.3.3. Maximum penetration

The simulated tactile sensor will work linearly all the way to the maximum depth after which the force will cease to effect. Therefore it is necessary to take the maximum depth into consideration. A good number for the maximum depth is the foam thickness times two. This will ensure the operation of the tactile in the desired force region. It has to be considered that if in the simulation the object hits the sensor with a greater force than the real sensor could handle, it will most likely penetrate the simulated sensor and the simulation will appear nonoperational although it is due to the fact that it will be into the saturated region of the real sensor where the results are not valid.

3.3.4. Friction properties

Measuring the friction properties of a tactile sensor is very difficult. Obtaining reliable scientific data requires a complicated test setup as well as specific hardware. The most reliable way of producing proper friction parameters is to use a measured friction force (F_t) vs sliding velocity ($\dot{s}_t$) curve using Eq. 13 and use curve fitting to obtain the parameters . As the friction curve of a real tactile sensor is difficult to measure in most cases, it is convenient to use a desired shape of the friction curve. A verification of the friction model is presented in [42].

3.3.5. General properties

The foam also has other properties that affect the behaviour of the real tactile sensor. When a point load is spread across several tactile elements, the foam spreads the load. This effect is not modelled in the simulated tactile sensor. The spread is a function of the foam thickness as presented in [13]. The effect of the foam on load spreading is difficult to measure and varies greatly with different sensor types and cover materials. The shape of the foam cover also affects the spread. At the edges of the sensor, the force will spread more as the edges of the sensor foam are stiffer than the center of the foam. Therefore the load spreading is not taken into account in the simulated sensor. This could be done by using an algorithm such as the one proposed in [37].

On the real sensor the hysteresis behaviour of the sensor can cause the tactile to show slightly different readings with the same compression depending whether it is being compressed or decompressed. This is due to the hysteresis behaviour of the covering foam. This behaviour is not included in the tactile sensor model.

3.4. An example of defining the parameters of a tactile sensor

The tactile sensor model was fitted to the RS-55-066 low density conductive foam used on the experimental tactile sensors for the Tombatossals Robot in the Robotic Intelligence Lab at the University Jaume I (see Fig. 5). This foam increases the sensibility of the sensors from the one measured with the original sensor material. The following section shows the definition for parameters for the foam in question.

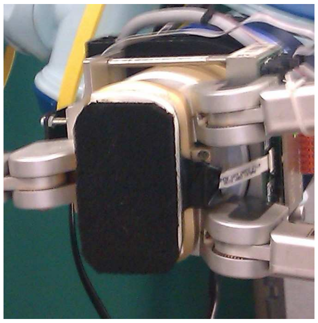


Figure 5: Low density conductive foam (RS-55-066) used on the experimental tactile sensors for the Tombatossals Robot in the Robotic Intelligence Lab at UJI

3.4.1. Normal force parameters (k,c_n)

In order to determine the stiffness of the foam different weights were placed on the tactile foam and the compression of the foam was then measured. The thickness of the foam is 6 mm and the weight area is 804.25 mm^2. The measurements are shown in Fig.6. The data have been approximate to a line, given the assumption that the foam has a linear stiffness coefficient shown in previous studies. However, when dealing with different materials, the coefficient could be modeled as nonlinear. The linear fit to the data (red dashed line) does not intersect the zero. This means that the real tactile foam has a certain tolerance before it starts showing readings, which is undesirable. In order to have the correct behaviour as well as zero intersection another linear fit with a zero intersection was used as the reference measurement (black solid line in Fig. 6). If the real sensor presents this behavior and it want to be modeled, it can be included when the force provided by the simulated sensor model is converted to tactile readings, to make them more similar to those provided by the real sensor.

As the fit is linear, a single point from the curve can be used as the reference pressure to deduce the spring constant for the tactile. At compression 2.5 mm the weight would be 304.15 g and the pressure on the tactile would be 9810 Pa.

Using the Eq. 18 with a tactile that has 84 texels in an area of 0.00125 m^2, the tactile spring constant k would be 58.5 N/m. The damping coefficient c_n is estimated as 1% of k which is 0.0585 N/m. The softer foam decreases the maximum load applicable to the tactile but it makes it more sensitive.

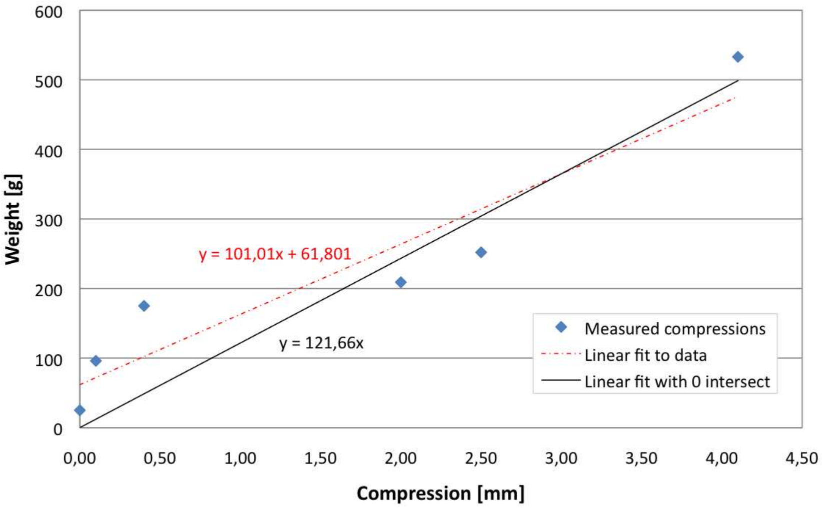


Figure 6: Foam (RS-55-066) compression measurements varying applied weight

3.4.2. Friction force parameters

An experiment was conducted in order to determine the friction parameters for the foam. A piece of foam was attached to a robot arm using a cable. A weight (588 g) was put on top of a piece of foam with an area of 900 cm^2) and then the robot arm was moved at an increasing velocity while recording the force from the cable. This enables the creation of a friction force vs sliding velocity graph.

This experiment presented some problems given the limited measuring capabilities of the the existing hardware. The thickness of the foam complicates the measurements, as it tilts before it starts sliding which causes an offset in the breakaway velocity of the measurements. Also, the thickness causes some waviness in the measurements as the foam tilts. However, as it is not the purpose of this section to study the friction properties of objects, but to show a general procedure to get the sensor parameters, the obtained results were accepted.

The acquired friction force vs sliding velocity graph was then used to manually create a plot that fit it approximately using the friction force model defined by Eq. 13. Although a curve fitting algorithm would have been more appropriate, it was not used at this point as it is a topic for future research. The measured friction force was divided by the number of texels in the tactile

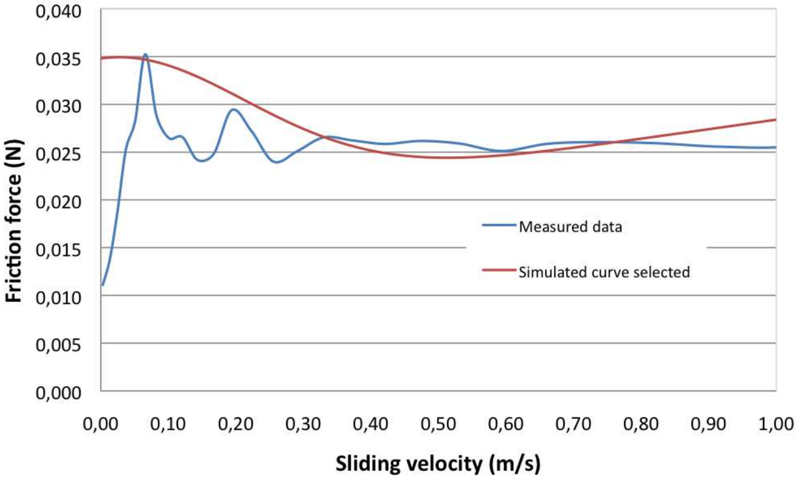


Figure 7: Friction force (F_T) vs sliding velocity ($\dot{s}_t$) graphs

sensor as the parameters for a single texel were searched.

Using a mathematical loop program (MeVEA), the fitted curve was obtained with the following friction parameters: F_n=0.08, σ_0=1, σ_1=0.1, x_0=0.3, c_t=0.01, μ_s=0.435 and μ_d=0.23. Fig. 7 shows the measured values and the manually fitted curve.

4. Implementation on the OpenGRASP Toolkit

The tactile sensor model presented in the previous section has been implemented using OpenRAVE [14], a planning architecture developed at the Carnegie Mellon University Robotics Institute. It is an open architecture targeting a simple integration of simulation, visualization, planning, scripting and control of robot systems. It enables the user to easily extend its functionality developing its own custom plugins.

Following its design, the Simulated Tactile Sensor Plugin has been developed and is available in OpenGRASP [43, 44], a simulation toolkit for grasping and dexterous manipulation consisting of a set of OpenRAVE plugins and other tools like the RobotEditor.

The development of the tactile sensor plugin included the definition of the tactile sensor geometry, the tactile sensor data and the implementation of the sensor interface specified by OpenRAVE.

Table 2: Example of an OpenRAVE tactile sensor definition

```
< AttachedSensor >
< link > object1 < /link >
< sensorname = “T1”  type = “SimTactileSensor” >
 < thickness > 0.006 < /thickness >
 < k > 58.5 < /k >
 < cn > 0.0583 < /cn >
 < sigma0 > 1.0 < /sigma0 >
 < sigma1 > 0.1 < /sigma1 >
 < x0 > 0.3 < /x0 >
 < ct > 0.01 < /ct >
 < mus > 0.435 < /mus >
 < mud > 0.23 < /mud >
< /sensor >
< /AttachedSensor >
```

The tactile sensor requires a set of parameters to be specified for each sensor. The model presented on this study allows the creation of sensors based on any geometry which is specified by the mesh of the body to which the sensor is attached. For each sensor, the parameters necessary to calculate the contact forces (see Section 3) and its thickness need to be defined. An example of a sensor definition using the OpenRAVE XML format can be seen in Table 2 with the results obtained in Section 3.4.

The tactile sensor data is the structure returned by the tactile sensor when it is requested. It contains the size of the tactile array, a vector with the tactile values calculated in each cell and the sum of all the tactile values. This structure is the same as the one used by the real tactile sensor, which makes the real and simulated sensors feedback appear identical to the controllers.

The plugin is an implementation of the sensor interface. OpenRAVE creates a new tactile sensor when specified on a robot definition. Each time step, when the sensor is updated, it gets the positions and velocities of the sensor and objects and checks if they are colliding. On each contact point, it calculates the contact forces with the equations detailed in Section 3.2. These forces and torques are applied to the sensor and objects. Finally, given that a real tactile sensor produce tactile intensities instead of forces, these intensities can be calculated using a linear conversion and the tactile data structure is updated. Controllers can query and use this tactile sensor feedback as required.

5. Experiments on Robot Grasping

In order to determine the simulated tactile sensor performance, some experiments were carried out. The tests can be divided into two groups. Tests for validating the physical properties and features of the tactile sensor and tests for applying the model to situations where the real sensor is commonly used. In the following sections these experiments are described in detail.

All the experiment were carried out with a simulated model of the Schunk PG70 parallel jaw gripper or by using a single finger model of the Schunk gripper. Each finger of the gripper had a tactile sensor attached to it. This is a resistive tactile sensor, with an array of cells of 14 rows by 6 columns.

5.1. Validation of the physical properties of the simulated tactile sensor

The basic physical properties of the tactile sensor element were tested by two different static methods. First, the capability of the sensor to return correct force feedback is validated. Second, the capability of the sensor to bear loads in static situations was tested. This is important as the sensor needs to be able to hold the objects for an extended period of time. This shows that the proposed method does not suffer from the same problem as some other models with near zero velocity.

5.1.1. Force sensor feedback

The load sensing capability of the sensor was tested by loading the sensor with a varying weight object and the total force from the tactile elements was then compared to the total force required to keep the object static (see Fig. 8a). The weight test range was from 0.1 to 10 kg. The parameters used in this simulation were a spring constant(k) of 1000 N/m and a damping coefficient(c_n) of 10 Ns/m, with a time-step size of 0.0001 s. Similar results can be acquired with smaller and greater masses by changing the k and c_n values of the sensor in order to keep the cube horizontal and stable.

A summary of the tests for 3 different weights is presented in Table 3. The table shows the object weight, the theoretically required force to hold the object static and the sum of the forces obtained by each texel from the tactile sensor. Given that in this case all the forces are pointing to the same direction, we can calculate the total force as the algebraic sum of the

Table 3: Tactile weight sensing capability

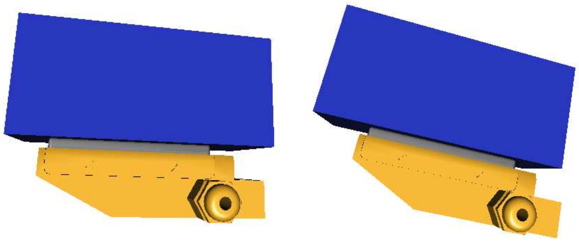
Object weight (kg)	Required force [N]	Tactile Force Feedback [N]
0.1	0.9800	0.9800
1	9.8000	9.7999
10	98.0000	97.9899

force values. The sum of the forces does not exactly match the required force due to the slight tilting of the weighing object. Therefore a fraction of the load is carried by the friction force which is not included in the normal force returned. Based on the results, the tactile sensor give out correct force readings.

5.1.2. Static Friction Bearing Capability

The static friction holding capability of the tactile sensor was tested by tilting the sensor by 10 degrees and seeing if the friction will hold the object static (see Fig. 8b). One of the main problems for collision models is the static holding friction. Most of the collision response methods will gradually let the object slip away from the grasp as near zero sliding velocity is handled poorly by the friction model [23].

The static friction bearing capability was tested by tilting the finger by 10 degrees and setting the 1 kg cube on top of it as shown in Fig. 8b.



(a) (b)

Figure 8: Simulated Robot finger conducting static tests: (a) 0 degrees tilt angle (b) 10 degrees tilt angle

In Fig. 9 the results of the test are shown. The creep value is the position difference to the initial position. The cube moves slightly from the starting position and then settles in position due to friction. The cube settles to the

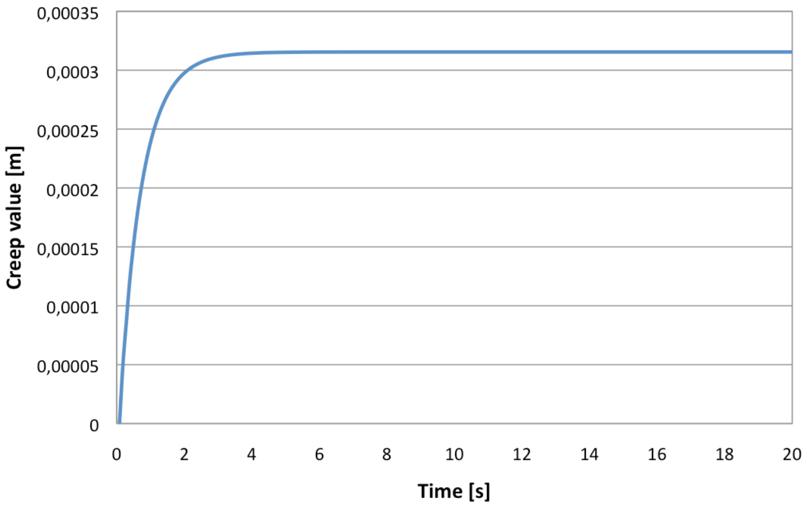


Figure 9: Creep value as position difference to initial position [m] of the cube when placed on a slanted tactile.

same position within 10 decimal accuracy within 10 seconds. The simulation was continued for 20 minutes confirming that the cube does not move from that position.

This proves that the tactile sensor is capable of holding loads at a static position and the importance to include a proper friction model to a grasping simulation.

5.2. Testing the model on robot grasping applications

5.2.1. Grasping various shape objects

Tactile sensors are normally used to grasp various shape objects. Therefore it is important to verify that the tactile sensor model can also handle grasping variously shaped objects. Four different objects were chosen representing typical graspable objects such as: a sphere, a box, a cylinder and a more complex form like a spray bottle. The objects were placed between two robot fingers and the tactile image and contact forces were observed during a simulation run. In Fig. 10 the object, the corresponding mesh, a grasping position the corresponding tactile images and the normal contact forces are shown. It can be seen that the tactile sensor is able to detect as well as hold on to all of the shapes. The tactile resolution can be changed in the simulation model by increasing the density of the mesh used to create the sensor.

It can also be seen in Fig. 10 that the simulated tactile sensor gives out highly accurate impression imprints of the grasped objects. As mentioned in the introduction, this tactile impressions can be used in a variety of application such as object recognition using tactile exploration and grasp stability evaluation [8].

5.2.2. Comparing tactile readings with the real sensors

A task of grasping and picking up a cube was selected as the test scenario. The idea was to perform the same task using this robot and compare the results with the ones obtained by executing the same actions in the simulator. In order to accomplish this, a high level controller was implemented using an abstraction architecture presented in [45], which allows to switch between real and simulated hardware transparent from the controller point of view.

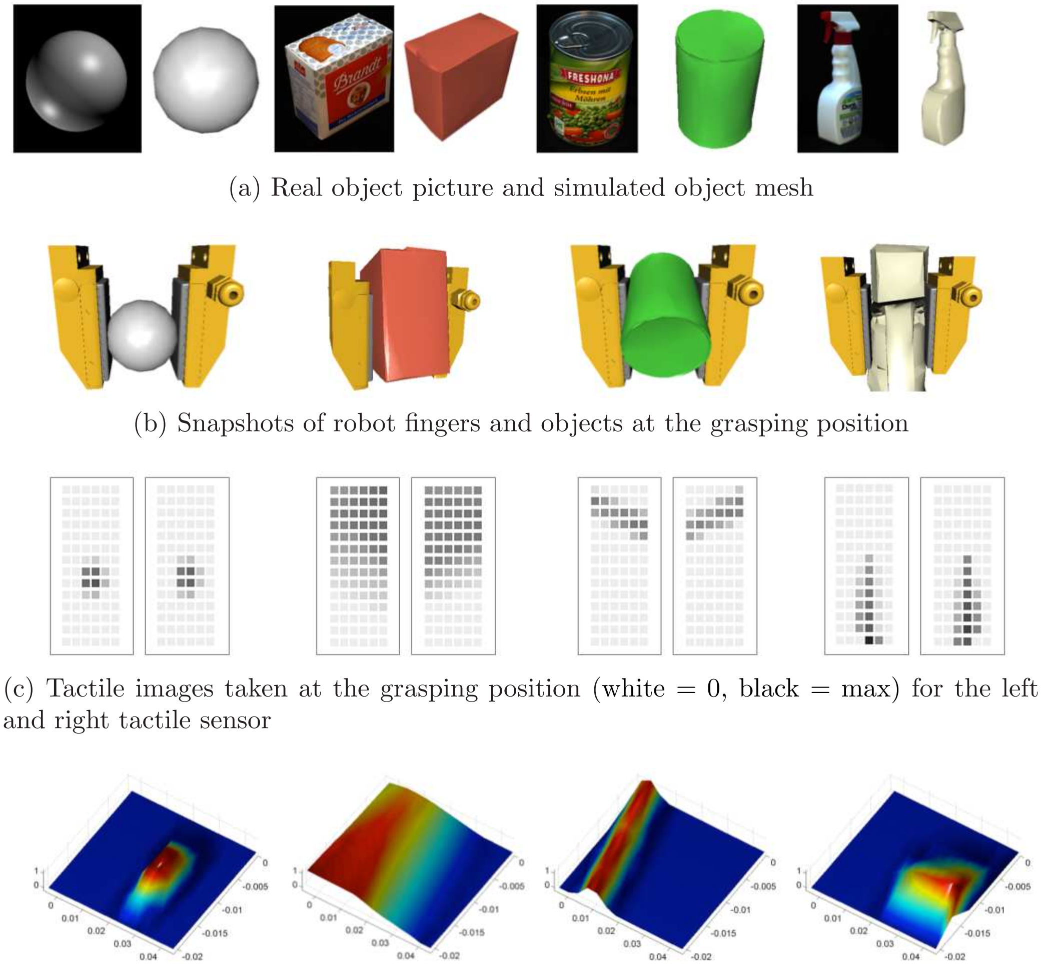
The tactile sensor feedback was used to control the grasping force and to determine the stability of the grasp. The experiment was defined as an abstract action consisting of five primitive actions: approach, grasp, lift, move down and release.

The controller turns this abstract information into the gripper specific primitives and transitions. It then drives the Schunk actuator using velocity control until the first contact with the tactile sensors is detected. After the initial touch the controller switches to force control by setting the maximum current of the Schunk gripper based on the tactile sensor feedback. The tactile value sums are used as the reference for the force control. This particular case shows the function of the tactile sensor in combination with the robot controller.

Given the abstraction architecture's ability to be embodiment independent, the same controller was used to control the real robot as well as the simulated case.

An image sequence of the real robot performing the work cycle with the tactile images produced by each sensor, can be seen in Fig. 11. The experiment was performed with the tactile sensor covering 30 percent of its area when touching the cube.

The actions were executed as expected but closer investigation of the tactile values revealed some problems. When grasping the cube, even pressure was applied to each tactile sensor that should return, as a result, very similar tactile images. However, significant differences between the individual tactile elements can be seen in the bottom row of Fig. 11 where one sensor tactile



(a) Real object picture and simulated object mesh

(b) Snapshots of robot fingers and objects at the grasping position

(c) Tactile images taken at the grasping position (white = 0, black = max) for the left and right tactile sensor

(d) Normal force readings in Newtons at each texel position measured in the grasp position for the right tactile sensor.

Figure 10: Testing results of grasping four different objects.

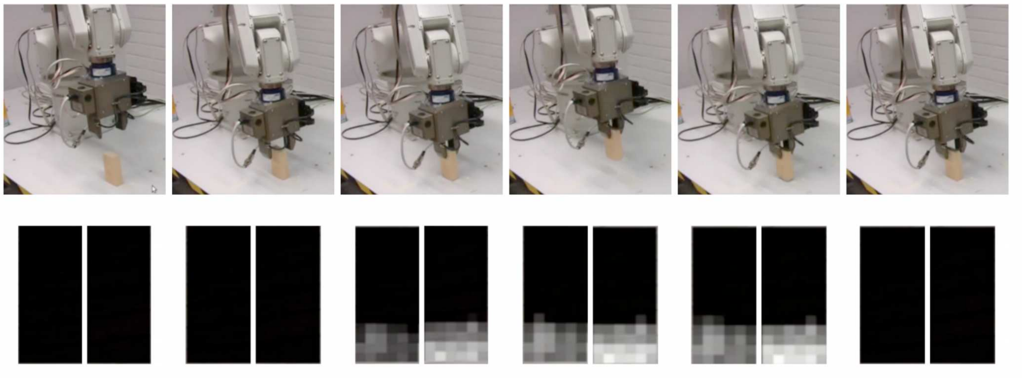


Figure 11: Real robot performing the chosen work cycle. On the top row, pictures of the robot performing the task on each stage. On the bottom, the tactile images generated by left and right sensors.

image is substantially lighter than the other in most cases, when touching the cube. Therefore the sum of the tactile values on opposite sides are not the same even when they should be.

The simulated robot performed the same work cycle as the real one. Images of the robot on each work cycle phase are shown in Fig. 12 with the corresponding tactile images at the bottom.

Closer inspection of the tactile values reveals that the sensors perform exactly as expected. Under even pressure, the individual tactile element values are the same in the area touching the cube. The individual texels in the sensor are not visible in the figure since the values are identical and therefore no color difference is visible. Opposite sensors also return similar tactile values.

The simulation trials show that the simulated tactile sensor element behaves physically correct. The simulation model shows consistent results whereas the real tactile sensor results vary on each work cycle. This is due to the fact that in simulation there are no manufacturing flaws or problems from wearing. The system consistently performs the same way under the same conditions. In addition, the detection tolerances from individual elements do not pose a problem.

When using the tactile imprints to evaluate the stability of a grasp, the simulated tactile could be used to teach a controller what a stable grasp

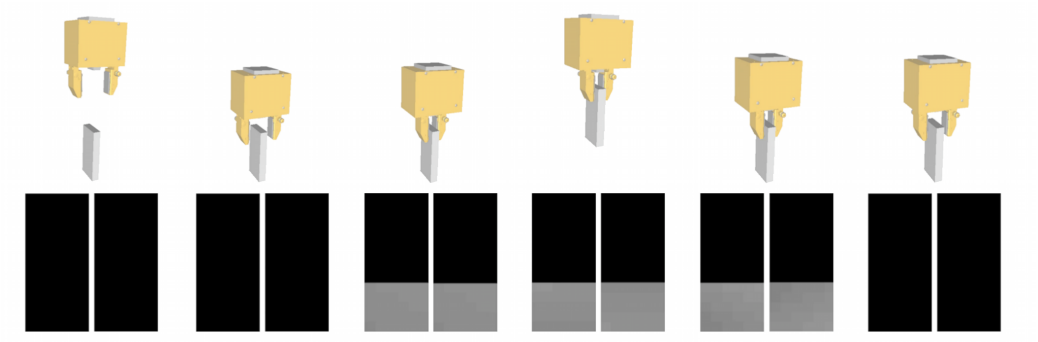


Figure 12: Simulated robot performing the chosen work cycle. On the top row, pictures of the simulated robot performing the task on each stage. On the bottom, the tactile images generated by left and right sensors. (individual texels not visible due to close to identical values)

looks like. The sensors currently available, will fail to detect a stable grasp based on the simulated images due to difference between the simulated and the real sensor feedback given the imperfections between the real individual sensors. The simulated tactile could be used to teach the controller what an ideal stable imprint would look like, but the real sensor would not display an identical image. An algorithm for allowing deviations from the ideal imprint would have to be developed.

There are some features such as resolution, noise, hysteresis, creep and aging that characterize a tactile sensor [46] and that should be considered in simulation in order to get a more similar response to the real sensors readings. The resolution in simulation can be changed freely, so it can match the spatial and temporal resolution of the real tactile sensor which are hardware dependent. From the real sensor's results it can also be seen that the tactile values suffer from noise in the results whereas, at the moment, the simulated sensor reports the changes in the force directly without any interferences. The hysteresis in the real tactile can also be significant due to the material covering the tactile elements. Finally, the real sensor material also causes some creep in the results as the foam cover resistance changes over time even under constant pressure. This change settles after some time but there is always some creep even after an extended period. The cover material also causes aging to be a problem when using the real tactile. All these features of the real tactile sensor can be added to the tactile sensor model. The difficulty is that the variance from sensor to sensor can be quite considerable as shown in Fig. 11 as well as in [37].

Further experiments [47] using the proposed tactile sensor model on commercial hands such as the Schunk 3-finger hand have been performed, showing that it is possible to create a dynamic simulation of a complete robot system with it.

5.3. Mimicking the human finger with a tactile sensor

An experiment with the human hand was conducted to test the versatility of the tactile model as well as to show the direction of the future work. The tactile sensor were created using geometries of human fingers (human hand model courtesy of the Biomechanics and Ergonomics Group, Universitat Jaume I) and then a cylinder shaped object was placed in between the index finger and the thumb (see Fig. 13). In the image it can be seen that the finger senses the object in a correct manner. The purpose of the experiment is to show that any shape can be used to form the tactile sensor. This is not an extensive test as such, but gives ideas for various uses of the simulated tactile sensor for the future. The freedom of form of the tactile sensor element can be extremely useful in modeling different tactile applications.

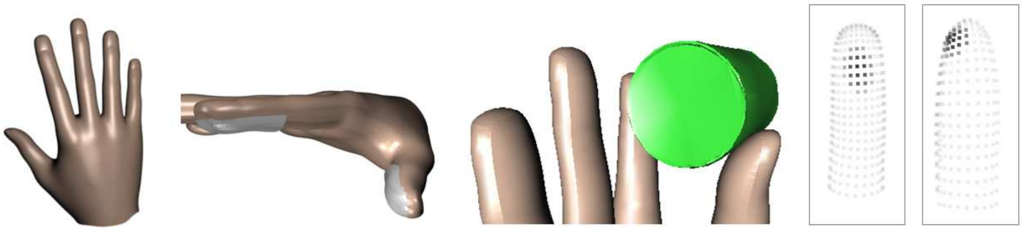


Figure 13: Human finger tactile holding a cylinder shaped object. It can be seeing from left to right: the human hand geometry, the tactile geometries for the index and thumb, the hand grasping the cylinder and the tactile images for each finger.

Real tactile sensors are not accurate enough to model the human touch just yet. This sensor model can give researchers a head start on researching human touch given that the simulated sensors do not suffer from the same restrictions as the actual real tactile sensors do.

6. Conclusions and Future Works

In this study, a simulation model of a tactile sensor was presented. The simulation model is based on soft contact modelling with a full friction description. The sensor was tested using the most common use cases of tactile sensors in robotic grasping. The simulated tactile sensor performed all the tests including stable grasping without any errors and can be used as a tactile sensor model. The model can be updated to behave in the same manner as a specified type of tactile sensor such as one from Weiss Robotics. This would entail modifying the stiffness as well as adding delay and load spreading that are due to the covering material and other electrical properties such as noise.

The experimental results of the tactile sensor model show good performance in being able to produce tactile feedback. Problems arise when trying to calibrate the tactile model to correspond exactly to a certain real tactile sensor. This is due to the variations in the real tactile values which makes the process extremely difficult.

However, using this tactile sensor model can enable researchers to do experiments that should be theoretically possible but, due to the current limitations in the existing hardware, are still difficult.

The collision detection method for solving the contact points is currently a brute force method. Improvements on this area can greatly improve the overall computation times of the tactile sensor model. The collision response method is computationally effective as can be seen by the equations. The calculations are straight forward and the dimensions of the equations of motion matrices will not be increased given that the penalty forces are applied to the dynamic model as external forces. Future work includes improving the collision detection times and improving computational efficiency.

7. Acknowledgements

We would like to thank Javier Felip for his contribution on the experiments to determine the friction parameters of a real sensor. The research leading to these results has received funding from the European Community's Seventh Framework Programme under grant agreement n° 215821 and Fundació Caixa-Castelló and the Universitat Jaume I through the project P1-1B2011-25.

References

[1] M. H. Lee, H. R. Nicholls, Tactile sensing for mechatronics : a state of the art survey, Mechatronics 9 (1) (1999) 1–31.

[2] R. Dahiya, G. Metta, M. Valle, G. Sandini, Tactile sensing - from humans to humanoids, IEEE Transactions on Robotics 26 (1) (2010) 1 –20. doi:10.1109/TRO.2009.2033627.

[3] J. Tegin, J. Wikander, Tactile sensing in intelligent robotic manipulation a review, An International Journal on Industrial Robot 32 (1) (2005) 64–70. doi:10.1108/01439910510573318.

[4] J. Felip, A. Morales, Robust sensor-based grasp primitive for a three-finger robot hand, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009. IROS 2009., 2009, pp. 1811 –1816. doi:10.1109/IROS.2009.5354760.

[5] D. Kragic, S. Crinier, D. Brunn, H. Christensen, Vision and tactile sensing for real world tasks, in: IEEE International Conference on Robotics and Automation, Vol. 1, 2003, pp. 1545 – 1550 vol.1.

[6] M. Prats, P. J. Sanz, A. P. Pobil, Reliable non-prehensile door opening through the combination of vision, tactile and force feedback, Autonomous Robots 29 (2) (2010) 201–218. doi:http://dx.doi.org/10.1007/s10514-010-9192-1.

[7] K. Hsiao, S. Chitta, M. Ciocarlie, E. G. Jones, Contact-reactive grasping of objects with partial shape information, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2010. IROS 2010., 2010.

[8] Y. Bekiroglu, J. Laaksonen, J. Jrgensen, V. Kyrki, D. Kragic, Learning grasp stability based on haptic data, in: In Proceedings of the RSS 2010 Workshop: Representations for object grasping and manipulation in single and dual arm tasks, 2010.

[9] A. M. Okamura, M. R. Cutkosky, Feature Detection for Haptic Exploration with Robotic Fingers, The International Journal of Robotics Research 20 (12) (2001) 925–938. doi:10.1177/02783640122068191.

[10] J. Tegin, J. Wikander, Simulating tactile sensors in robotic grasping, in: Proceedings of The Third Swedish Workshop on Autonomous Robotics, Vol. 1, Stockholm, Sweden, 2005.

[11] S. Hasegawa, M. Sato, Real-time rigid body simulation for haptic interactions based on contact volume of polygonal objects, Computer Graphics Forum 23 (3) (2004) 529–538.

[12] J. A. Joergensen, L.-P. Ellekilde, H. G. Petersen, Robworksim - an open simulator for sensor based grasping, Robotics (ISR), 2010 41st International Symposium on and 2010 6th German Conference on Robotics (ROBOTIK) (2010) 1 –8.

[13] M. Shimojo, Mechanical filtering effect of elastic cover for tactile sensor, IEEE Transactions on Robotics and Automation 13 (1) (1997) 128–132.

[14] R. Diankov, Automated construction of robotic manipulation programs, Ph.D. thesis, Carnegie Mellon University, Robotics Institute (August 2010).

[15] D. Baraff, Analytical methods for dynamic simulation of non-penetrating rigid bodies, SIGGRAPH Comput. Graph. 23 (3) (1989) 223–232. doi:http://doi.acm.org/10.1145/74334.74356.

[16] B. Mirtich, Impulse-based dynamic simulation of rigid body systems, Ph.D. thesis, University of California, Berkeley (1996).

[17] P. Kraus, V. Kumar, Compliant contact models for rigid body collisions, in: Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on, Vol. 2, 1997, pp. 1382–1387 vol.2. doi:10.1109/ROBOT.1997.614330.

[18] M. Moore, J. Wilhelms, Collision detection and response for computer animationr3, in: SIGGRAPH '88: Proceedings of the 15th annual conference on Computer graphics and interactive techniques, ACM, New York, NY, USA, 1988, pp. 289–298. doi:http://doi.acm.org/10.1145/54852.378528.

[19] K. Hunt, F. Crossley, Coefficient of restitution interpreted as damping in vibroimpact, Transactions of the ASME Journal of Applied Mechanicss Series E (42) (1975) 440–445.

[20] Y. Khulief, A. Shabana, A continuous force model for the impact analysis of flexible multibody systems, Mechanism and Machine Theory 22 (3) (1987) 213 – 224.

[21] K. Johnson, Contact mechanics, in: Cambridge University Press, University Press, 1985, p. 452.

[22] B. Brogliato, A. ten Dam, L. Paoli, F. Genot, M. Abadie, Numerical simulation of finite dimensional multibody nonsmooth mechanical systems, Applied Mechanics Reviews 55 (2) (2002) 107–150.

[23] W. Drumwright, D. Shell, An evaluation of methods for modeling contact in multibody simulation, in: IEEE International Conference on Robotics and Automation, 2011.

[24] M. Shinya, Stabilizing explicit methods in spring-mass simulation, Computer Graphics International Conference 0 (2004) 528–531.

[25] D. Prattichizzo, J. Trinkle, Grasping, in: B. Siciliano, O. Kathib (Eds.), Handbook on Robotics, Springer, 2008, pp. 671–700.

[26] H. Olsson, K. Astrom, C. de Wit, M. Gafvert, P. Lischinsky, Friction models and friction compensation, European Journal of Control 4 (3) (1998) 176–195.

[27] C. Canudas de Wit, H. Olsson, K. Astrom, P. Lischinsky, A new model for control of systems with friction, IEEE Transactions on Automatic Control 40 (3) (1995) 419–425. doi:10.1109/9.376053.

[28] M. Gfvert, Comparison of two dynamic friction models, in: In proceedings of the 1997 IEEE International Conference on Control Applications, 1997, pp. 368–391.

[29] B. A. Bliman, M. Sorine, Easy-to-use realistic dry friction models for automatic control, in: In proceedings of 3rd European Control Conference, 1995, pp. 3788–3794.

[30] A. Miller, P. Allen, Graspit!: A versatile simulator for robotic grasping, IEEE Robotics & Automation Magazine 11 (4) (2004) 110–122.

[31] ODE, Open dynamics engine, `http://www.ode.org/` (Oct. 2011).
URL `http://www.ode.org/`

[32] Bullet, Game physics simulation, `http://www.bulletphysics.org/` (Oct. 2011).
URL `http://www.bulletphysics.org/`

[33] J. Tegin, B. Iliev, A. Skoglund, D. Kragic, J. Wikander, Real life grasping using an under-actuated robot hand - simulation and experiments, in: Advanced Robotics, 2009. ICAR 2009. International Conference on, 2009, pp. 1 –8.

[34] H. Scharfe, N. Hendrich, J. Zhang, Hybrid physics simulation of multi-fingered hands for dexterous in-hand manipulation, in: Robotics and Automation (ICRA), 2012 IEEE International Conference on, 2012, pp. 3777 –3783. doi:10.1109/ICRA.2012.6225156.

[35] A. Ghafoor, J. S. Dai, J. Duffy, Stiffness modeling of the soft-finger contact in robotic grasping, Journal of Mechanical Design 126 (4) (2004) 464–656.

[36] V. A. Ho, S. Hirai, Three-dimensional modeling and simulation of the sliding motion of a soft fingertip with friction, focusing on stick-slip transition, in: IEEE International Conference on Robotics and Automation, 2011, pp. 5233 –5239.

[37] Z. Pezzementi, E. Jantho, L. Estrade, G. Hager, Characterization and simulation of tactile sensors, in: Haptics Symposium, 2010 IEEE, 2010, pp. 199 –205. doi:10.1109/HAPTIC.2010.5444654.

[38] B. Armstrong-Helouvry, P. Dupont, C. de Wit., A survey of models, analysis tools and compensation methods for the control of machines with friction, Automatica 30 (7) (1994) 1083–1138.

[39] K. Weiss, H. Worn, The working principle of resistive tactile sensor cells, in: Mechatronics and Automation, 2005 IEEE International Conference, Vol. 1, 2005, pp. 471–476.

[40] K. Kerpa, O. Weiss, H. Worn, Tactile sensor system for an anthropomorphic robotic hand, in: Proc. IEEE Int. Conf. Manipulation Grasping, 2004.

[41] K. Kerpa, O. Weiss, H. Worn, Development of a flexible tactile sensor system for a humanoid robot, in: IEEE/RSJ International Conference

on Intelligent Robots and Systems, 2003. IROS 2003., Vol. 1, 2003, pp. 1–6 Vol. 1.

[42] R. Kelly, J. Llamas, R. Campa, A measurement procedure for viscous and coulomb friction, Instrumentation and Measurement, IEEE Transactions on 49 (4) (2000) 857 –861. doi:10.1109/19.863938.

[43] OpenGRASP, A simulation toolkit for grasping and dexterous manipulation, `http://opengrasp.sourceforge.net/` (Oct. 2011).
URL `http://opengrasp.sourceforge.net`

[44] B. León, S. Ulbrich, R. Diankov, G. Puche, M. Przybylski, A. Morales, T. Asfour, S. Moisio, J. Bohg, J. Kuffner, R. Dillmann, Opengrasp: A toolkit for robot grasping simulation, in: N. Ando, S. Balakirsky, T. Hemker, M. Reggiani, O. von Stryk (Eds.), Simulation, Modeling, and Programming for Autonomous Robots, Vol. 6472 of Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 2010, pp. 109–120.

[45] J. Laaksonen, J. Felip, A. Morales, V. Kyrki, Embodiment independent manipulation through action abstraction, in: IEEE International Conference on Robotics and Automation, 2010, pp. 2113 –2118. doi:10.1109/ROBOT.2010.5509153.

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

[47] B. Leon, J. Felip, H. Marti, A. Morales, Simulation of robot dynamics for grasping and manipulation tasks, 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids).