Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

MASTERTHESIS

Tactile based grasping with the biomimetic sensors BioTac and the Shadow Dexterous Hand

submitted by

Yannick Jonetzko

University of Hamburg

Department of Informatics

TAMS - Technical Aspects of Multimodal Systems

Curse of studies: Informatics

Matriculation-No.: 6739486

Primary supervisor: Prof. Dr. Jianwei Zhang

Secondary supervisor: Dr. Norman Hendrich

Abstract

The human sense of touch is one of the most important senses that is experienced in everyday life. Mechanoreceptors are all over the skin and allow to gather important information like temperature, robustness or structure of touched surfaces. This information is necessary for grasping and object manipulation, like opening a bottle. As this sense is hugely beneficial for humans, duplication on robots can show further increasing advantages.
This work presents the tactile sensor BioTac. A sensor is attached at each finger of the anthropomorphic robot hand Shadow Dexterous Hand. With these tactile sensors, the applied force on an object should be controlled to prevent deformation or even destroying of objects. To control these forces, it is necessary to gather significant information from the sensor data and calculate the contact point and the applied force. To generate the information out of the raw sensor values, it is crucial to analyze and preprocess the data. Preprocessing includes the elimination of temperature dependencies, noise filtering and normalization. Different calibration processes should improve the accuracy of the contact location and force estimation.

Zusammenfassung

Der Tastsinn ist einer der wichtigsten Sinne des Menschen. Mithilfe der Mechanorezeptoren auf der Haut werden wichtige Informationen wie Temperatur, Festigkeit oder Struktur der berührten Oberfläche gesammelt. Diese Informationen sind nicht nur essentiell, um Dinge zu greifen sondern auch, um sie zu manipulieren wie das Öffnen einer Flasche. Auch in der Robotik möchte man sich diesen Vorteil zunutze machen.
In dieser Arbeit wird der taktile Sensor BioTac vorgestellt und genauer betrachtet. Diese Art Sensor ist an der antropomorphen Roboterhand Shadow Dexterous Hand montiert. Mithilfe dieser taktilen Sensoren soll es möglich gemacht werden, die von der Roboterhand auf ein Objekt angewendete Kraft zu kontrollieren, um so unerwünschte Verformungen oder sogar die Zerstörung des Objekts zu vermeiden. Dafür ist es notwendig, die angewendete Kraft sowie den zugehörigen Kontaktpunkt zu kennen. Um diese aus den rohen Daten des Sensors zu gewinnen, werden die Daten analysiert und aufbereitet. Diese Aufbereitung besteht aus der Eliminierung der Temperaturabhängigkeit, der Filterung von Rauschen und der Normalisierung der rohen Sensordaten. Mit verschiedenen Kalibrierungsverfahren wird versucht, den Sensor zu kalibrieren und somit die Genauigkeit des Kontaktpunktes sowie der Kraft zu verbessern.

Contents

List of Figures

1 Introduction

In the domain of service robots, researchers and programmers provide their robots with different kinds of tasks to support in their everyday abilities, such as cleaning a room, filling the dishwasher or cooking. Their main knowledge and ability for tasks is achieved through either hard-coding or learning, for example by demonstration of a human. In most cases, the robots environmental conditions stay the same. If changes are made, then only with minor changes. However, problems still occur when placing the robot in a completely unknown environment, where it is expected to perform the same tasks.

The robot setup used in this work is a PR2 with a five finger Shadow motor hand mounted to the right arm, pictured in figure 1.1. The PR2 is in the original setup equipped with two two finger grippers, one at each arm. The original gripper can be seen in figure 1.1 at the end of the left arm. The PR2 is currently one of the most sophisticated service robots, even though it was released in 2009. The main purpose for this robot was to function in the human environment. Many interesting tasks like assisting in the preparation of pancakes [1] were implemented. The setup with the Shadow Dexterous Hand is currently the only PR2 with this gripper, which allows more complex and precise grasps than the original parallel gripper.

This thesis starts with the motivation 1.1 and the goals of this thesis 1.2. In the second chapter, the state of the art is presented. This chapter looks at the definition of grasping in the field of robotics in 2.1, tactile sensing and the BioTac sensor in 2.2. Furthermore, it explains anthropomorphic grippers and the Shadow Dexterous Hand in section 2.3. The last point of the state of the art is ROS (Robot Operating System) in 2.5. In chapter number 3 the sensor data of the tactile sensor BioTac is analyzed, preprocessed and calibrated in order to estimate the point of contact in section 3.4 as well as the normal force estimation in section 3.5. These two estimations are presented in the tactile based grasping chapter 4. Here a simple way of grasping using two fingers is presented which could be the foundation of complex tactile based grasps.

1.1 Motivation

As humans, our daily tasks involve work with our hands, grasping any kind of objects whether they are hard, soft, fragile, cold or warm. For robots, these simple tasks, like grasping for a glass of water, are still a big problem. Without any feedback from sensors, plastic cups, for example, are nearly impossible to pick up. Currently, many grippers use little to no feedback because sensors are missing or too expensive to integrate. Looking at fragile objects, humans are capable, through their tactile sense in their fingertips, of grasping e.g. a plastic cup without deforming or destroying it, even if their eyes were blindfolded. This ability, with no given information except a region of the position of the object, is desirable for robots. Tactile information will not only

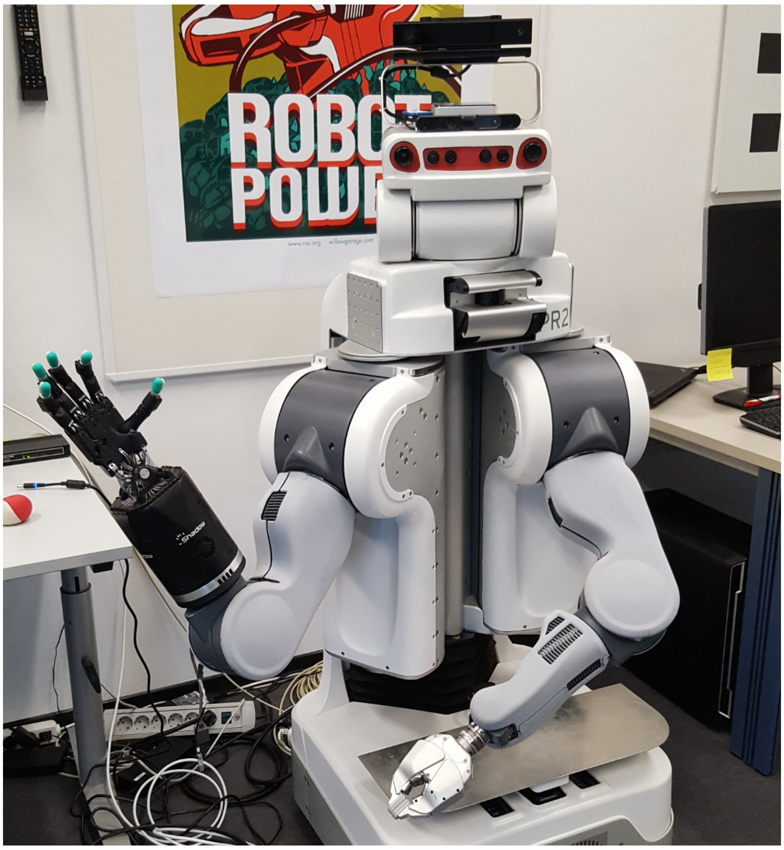


Figure 1.1: The robot platform used in this thesis is a PR2 service robot founded in 2009. This robot was built to aim and support in the everyday human environment. Additionally this robot has a custom modification, the Shadow Dexterous Hand mounted to the right arm. The anthropomorphic gripper is equipped with five BioTac tactile sensors. The main part of this work is the analyses of these tactile sensors and the use for tactile based grasping. The longterm goal is the integration of a tactile based grasp with the whole robot setup.

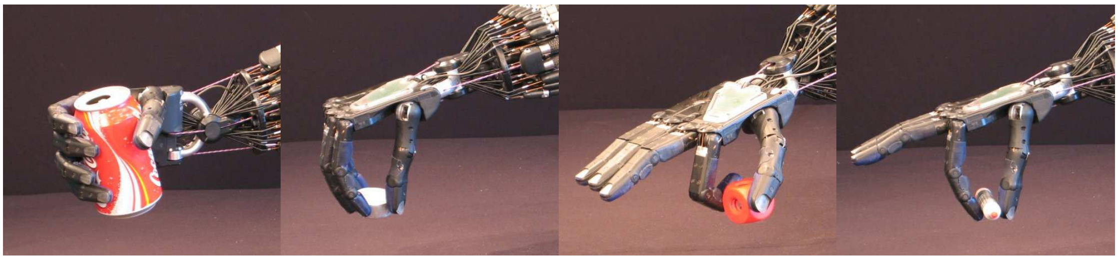
advantage robot's grasping ability towards various kinds of objects, but can also, for example, stabilize control and improve location checks. Imagine a robot opening a bottle, touching the bottle cap instead of the neck. Therefore, gathering tactile feedback is a small but significant step towards the goal of robotic grasping.
The human environment consists of many small and fragile objects, requiring ambitious tasks like plugging a USB or a power cord on a daily basis. For service robots to fulfill these tasks, tactile sensing and the anthropomorphic gripper itself are an advantage. As the human hand is very versatile, it can effortlessly pick up different sized objects like water bottles, a pen or a nail. It would be nice if these advantages could also be used with service robots.

1.2 Thesis goals

The overall goal of this thesis is to perform tactile based grasping with the Shadow Dexterous Hand and the PR2. To achieve this goal, various milestones can be sought. This work is limited to the Shadow Hand and the attached BioTac sensors. One goal is the analyses and understanding of the tactile sensor BioTac and the possibilities of using it within tactile based grasps on robots. Another goal is to analyze the important sensor data for grasping and reduce appearing problems. Even though multiple difficulties need to be expected, the end goal is to firmly hold an object, through the anthropomorphic gripper and the tactile sensors.

2 State of the art

Grasping with anthropomorphic robot hands has become an increasing research topic, with several publications every year. The first section of this chapter 2.1 will explore how anthropomorphic grippers are used for grasping, which reaches from planned grasping of known or unknown objects, over human teleoperated grasps, to learning by demonstration. Section 2.2 examines the human sense of touch and introduces the use of tactile sensors as well as the used BioTac sensor. In 2.3 the anthropomorphic gripper Shadow Dexterous Hand will be introduced. This robot hand is used to grasp objects including tactile feedback from the attached BioTac sensors. The used operating system ROS will be presented in the fourth section 2.5 including the structure of the Shadow Hand code for ROS.



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

Figure 2.1: Four common types of a human grasp are shown. Different sized and shaped objects can be grasped with at least one of them [2]. Figure (a) shows an object enclosed by a power grasp. Here, as many contact points as possible are applied. Figure (b) shows an all finger precision grasp and how smaller objects could be grasped and hold firmly. In figure (c), a two finger pinch grasp is applied. The advantage of this grasp is a large contact face while still being very precise. Figure (d) shows a very precise grasp. For the grasp types in figures (b), (c) and (d), it is useful to integrate a tactile feedback of the fingertips.

2.1 Grasping

Grasping is a problem as old as robots and something that requires special tools. For humans, this tool is the hand, which allows us to work, live, and function on an everyday basis. Animals rely, for example, on paws or claws, which support their living and survival, but is not comparable to the dexterous grasping of a human. For humans, it is the hand that allows us to grasp things and even animals often grasp things e.g. with paws, but in the most cases not as dexterous like a

human. Robots, on the contrary, do not have any specific tools. Different designs were developed to distinguish between grippers that mimic the human anatomy and grippers that fulfill practical requirements like simplicity, low cost or required tasks like grasping only one specific object [3].
In this section, only the way to a grasp is considered, not the improvement of grasp stabilities like the change of contact points or manipulative motions. Such a grasp can be reached in different ways, the most common approaches are either telemanipulation, where the gripper is controlled remotely by hand, teaching by demonstration, where skills are learned from the demonstrator solving different tasks, or by solving the grasp problem in an analytical way. Analytical ways usually happen through analyzing human grasps or computing grasps. As the focus of this work is tactile grasping. All approaches could profit from the given tactile information, at least by evaluating the grasp after it is applied.
An interesting aspect, besides the way of how to grasp successfully in the field of tactile sensing, is the grasp type itself. The grasp type depends on the way the fingers or the whole hand is positioned on the object. There are different ways to divide grasps into types and how to classify them. Some common grasps are pictured in figure 2.1, the importance of tactile sensing differs for each grasp. As the focus of this work is on the tactile feedback in the fingertips, the power grasp shown in (a) can not benefit from this information as much as (b), (c) and (d). The next section gives an overview of how to generate grasps within the different common approaches, the first one being telemanipulation 2.1.1, learning by demonstration 2.1.2 and analytical approaches 2.1.3.

2.1.1 Telemanipulation

One way to grasp objects is a remote-controlled robotic hand that tracks the gestures of humankind. For this to occur for robots, devices like a data glove or optical tracking are necessary. The teleoperated grasping is less interesting for the field of autonomous service robots yet worth mentioning. An important field of this technology is the remote control in inaccessible or hazard environments [4]. The idea of teleoperation, including tactile feedback, goes back to a publication from 1992, which describes different technologies like an exoskeleton or vibrating gloves [5].

2.1.2 Learning by demonstration

A more interesting way in the field of service robotics is learning by demonstration. Service robots should fulfill tasks that a human does every day. The goal is teaching robots such movements that they could perform them autonomously afterwards. Advantages of learning by demonstration is that people with no robotic, science or even computer experience can demonstrate robots various tasks [6].
Learning by demonstration can be differentiated in two parts. One part is to separate observations of a test person to perform a grasp or a manipulation task that tracks the hand movements either of the hand movement or the object [7]. The other part is similar to telemanipulation, where the robot is controlled remotely and the movements are tracked at the robot hand itself. The advantage, in this case, is that the test person can directly react to the behavior of the robotic

hand [8]. Therefore, the tactile feedback can be recorded and could improve the applied learned grasp afterwards.

2.1.3 Analytical approach

A third strategy to generate a grasp is the analytical approach. The common ways of generating grasps can be roughly defined in two groups, one where the size, shape, and location of the object is known, the other, where this information is not given.
One of the most famous frameworks for the analytical calculation of grasps is *GraspIt!* [9]. GraspIt! calculates contact points on known objects. The evaluation of a generated grasp is done with the calculated wrench space before the grasp is applied. The wrench space is spanned between the calculated contact points and includes the applied forces and torques at the contact points. A way to integrate the tactile feedback is to evaluate applied grasps afterwards to check if the calculation was successful.
An approach with unknown objects is made through taken information out of point clouds [10]. Successful grasps could be generated through 3D information about the front view of possible objects and the known kinematics of the robot manipulator and gripper.

2.2 Tactile sensing

For most living creatures on earth, tactile sensing is of fundamental meaning to interact and react with the environment. The so-called mechanoreceptors are responsible to give tactile feedback of touched surfaces, like temperature and surface properties. With a disturbed tactile sense, for example with chilled hands, even putting on a t-shirt, is a tough task. For humans, the use of tactile sensing can be differentiated in three different kinds of activity: manipulating, exploration and response [11]. For manipulation tasks, like opening a bottle, the tactile sense provides useful information about the position of the finger on the bottle and the cap. In the field of exploration, different materials can often be set apart better with fingers than with an optical observation, for example, synthetic leather and real leather. Through this sense, humans get information of temperature and surface properties that can not be gained by view. The response task is important to react to external stimuli through accidentally touching a heated hotplate for example. People with peripheral neuropathy can injure themselves badly in such situations.
For robots, the same three activities can be applied. In comparison to humans, robots are equipped with considerably less tactile sensors. In most cases, grippers just have sensors at the fingertips and rarely all over the hand and much less on the rest of the robot. This makes it much more difficult to get enough information to tackle those three tasks. The interesting aspects for manipulating with grippers are force control, contact locations, and stability assessment. Contact locations can be predicted by kinematics but not as accurate as contacts from tactile information. Similar to the human sense of touch, surface textures, hardness and thermal properties can be gathered for exploration with tactile sensors. The response task is important to detect external stimuli as well as making and breaking contacts.
In this section, the emphasis is on sensors which work like mechanoreceptors. The tactile sensors BioTac, used in this work, belong to this category. Tactile information can be gained with

other sensor types like proprioceptive, kinematic or force. With proprioceptive or kinematic sensors, the inner state of e.g. joints can be measured, which allows determining the position and orientation of contacts. Through force sensors, external applied forces can be measured. A combination of the known kinematics and the measured force allows computing the contact location [11].
The first part of this section describes the sense of touch of human hands (2.2.1) and a list of important sensor types (2.2.2). Last but not least, BioTac sensors are being described (2.2.3).

2.2.1 The human sense of touch

To understand tactile sensing, it is necessary to take a look at the human sense of touch and how the cutaneous mechanoreceptors are related to the sense [12].

The human hand is covered with 17.000 tactile units of four different types: two fast adapting types, FA I and FA II and two slowly adapting types, SA I and SA II. Fast means the moment of making the contact, slow the holding of the contact.

The four different tactile units on the inner surface of the hand differ in functional properties.

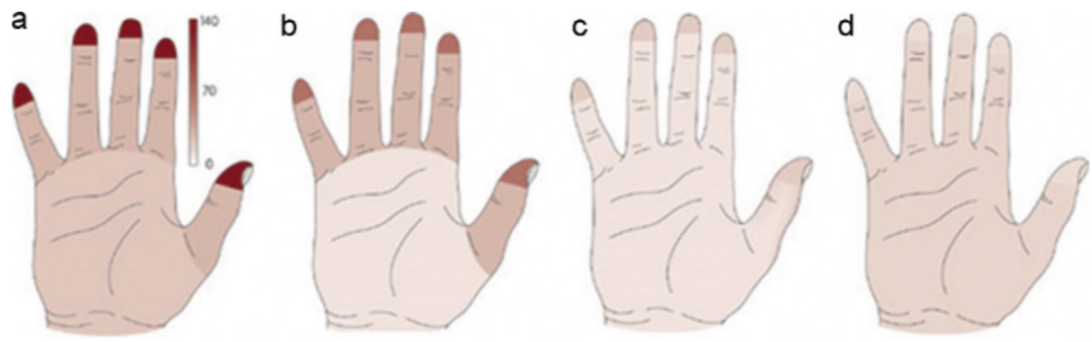


Figure 2.2: The distribution of the four different tactile units in the human skin on the hand. Figure (a) shows the distribution of the fast adapting FA I type for small contacts, figure (b) shows the fast adapting FA II type for large contacts, figure (c) shows the distribution of the slow adapting SA I type for small contacts and figure (d) shows the slow adapting SA II type for large contacts [12].

These properties are sensitive to static and dynamic contacts, as well as the structure of the contacted surface. They are also divided in the density within different regions around the hand. These densities are pictured in figure 2.2. The FA I receptors make up 43% of all mechanoreceptors on the inner hand surface. The density of the FA I tactile units is very high at the fingertips, lower at the rest of the finger and only roughly at the palm, which is shown in (a). With these very small receptors (in the way of diameter), impulses are recognized. The density of FA II receptors is shown in figure (b). The distribution of the units is similar to the first one, however, it is not that dense. The overall amount of these receptors is 13%. In figure (c), the density of SAI tactile units is shown with an overall amount of 25%. The density is similar on the finger without the tips and the palm. Through this sensor, small continued contacts can be perceived.

The last type, SA II is shown in (d). This is the only type where the density is higher at the rest of the finger than on the fingertip and makes 19% of the overall amount of the 17.000 mechanoreceptors.
Through these receptors, it is possible to distinguish distances between two contact points, the average being 1.6 mm at the fingertip. For the rest of the finger, the average resolution is 2.5 times higher whereas for the palm it is 5 times higher.

2.2.2 Sensor types

Different parts of robots must withstand different amounts of load. For example, a robot's foot has to carry the whole robot and therefore needs to be very robust. On the contrary, a tactile sensor should give feedback about very small amounts of forces. As a general rule, either a sensor is very robust but has a low resolution, or the resolution is high yet the amount of load decreases. A typical requirement for tactile sensors is a sensitivity of 1 to 2 mm^2.

Capacitive pressure sensor array For a capacitive sensor array, two layers of conductive material are placed on top of each other with a dielectric layer in between. One layer consists of electrode rows, the other of electrode columns. Compressing the dielectric layer in between with an applied contact changes the capacitance. The position of the contact can be determined by the electrode rows and columns. The applied force strength can be determined by the changed capacitance with the equation $C \approx \varepsilon A/d$, where d is the spacing between the two layers. A is the contacted area and ε is a dielectric constant which is dependent of the non-conductive material in between. This kind of sensor is often used in touchscreens [11]. Such sensors, among others, were used to cover large areas of robot surfaces [13].

Piezoresistive pressure sensor array The material is piezoresistive if the electrical resistance changes under pressure or tractive effort. This property is used for piezoresistive tactile sensors. In most cases, a piezoresistive inc is printed in patterns on a conductive rubber or bulk molded. In each case, a conductive supplement is necessary to create the piezoresistive property. In the first approach, conductive rubber columns and rows were connected with piezoresistive materials [14]. The sensor data tends to show huge drifts and hysteresis which was reduced but never really removed. Nevertheless, this technology could be employed in applications where accuracy is not that important [11].

Skin deflection sensing Besides the tactile information, a sensor with deformable skin improves the stability of a grasp, reduces shocks and protects the sensitive sensors under the skin. One early approach within this method was the use of an array of magnets and measuring the local magnetic field with hall sensors [15]. The first developed sensor was similar to the BioTacs, which are used in this work. This was an impedance tomographic tactile sensor. The sensor was used to detect shapes and the radius of curvature of objects. This sensor has a neoprene skin filled with distilled water. The rigid core is covered with an 8x5 array of electrodes. With this setup, a signal was measured that was proportional to the distance to the skin [16].

Other array sensors Another way to get tactile information is the use of optical sensors. The advantage of these sensors is the immunity of electromagnetic interferences. The idea of an optical base method is to place a camera inside the sensor and observe the pattern printed on the contacting surface. The material in between has to be transparent. Another way is to use an emitter of light and a detector that measures the reflected light from the surface. This idea could also be applied to acoustic or ultrasonic sensors [11].

2.2.3 BioTac

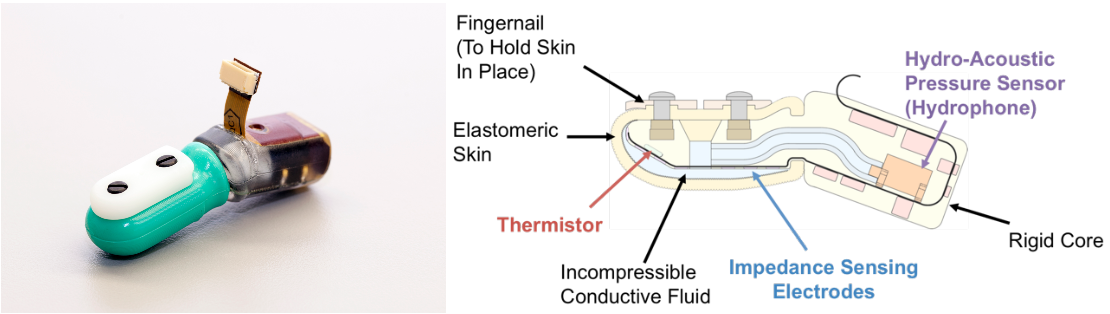


Figure 2.3: The tactile sensor BioTac: this is one of the most sophisticated sensors currently commercially available. It is designed after the model of the human fingertip. On the right side the cross section of the BioTac with its sensors is shown [17].

The BioTac [17] is a biologically inspired sensor which imitates the human sense of touch through mechanical stimulation and temperature. Measurements of the mechanical stimulation are for example forces, vibrations or the location of contact. The BioTac sensor differs from other tactile sensors in providing all the information in one device. Other common devices supply only information about contact location and force. The BioTac used in this work, which is pictured in figure 2.3 is about the size of a human fingertip. The BioTac belongs to the group of *skin deflection sensing*. All sensors lay within a rigid core surrounded by a protective elastic skin that is filled with a conductive liquid. The surrounding skin has a fingerprint-like surface to increase the friction and produce more vibrations when sliding over a surface.
Besides the use of the BioTac in the field of robotics, the sensors are also used for prosthesis. The BioTacs are integrated in a system with automatic reflexes like the human fingers. In this way, the user is able to grasp objects without the need of active force control.
Another currently advertised usage of the BioTacs is the evaluation of materials. The sensor is used in a fixed control element to repeat the same movements on different materials to define its quality. With this setup test samples can be created and the so-called spider plots are used to classify them. Such a plot is pictured in figure 2.4. Through these plots the sensitivity of the sensors gets visible [18].

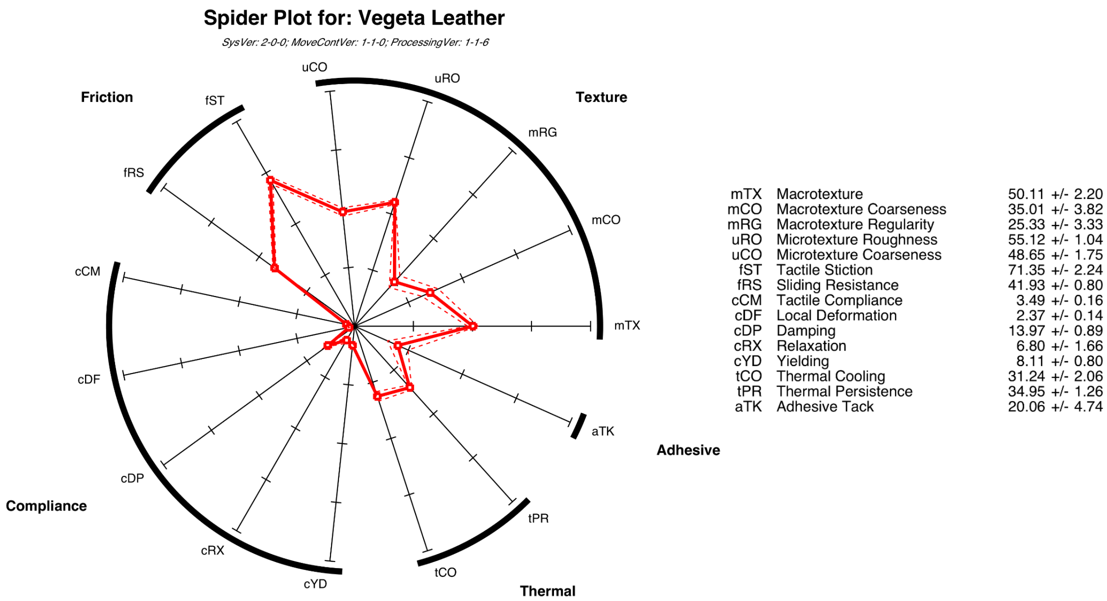


Figure 2.4: A spider plot showing evaluation and quality control of a touched material with the BioTac. The quality control is currently a large part of the usage of this sensor in the industry [19].

Brief history of the BioTacs

In the middle of 2007, four scientists from the University of Southern California and Umeå University submitted the first paper: *Biomimetic Tactile Sensor Array* [20]. In this paper the authors introduced an early version of the BioTac. A prototype of the core is shown in figure 2.5. 2008 the SynTouch Inc [18] company was founded which commercially produces and sells the sensors. The purpose of this sensor was initially intended for anthropomorphic robot hands and for prosthetics. Later on, the sensors were used to make quality control for different materials.
In 2013, SynTouch introduced the NumaTac sensor, it is a feature-reduced and low-cost version of the original that BioTac developed for mass production. This sensor provides a dynamic pressure signal that is similar to the dynamic fluid pressure of the BioTac. With this signal, making and breaking contacts can be easily detected. The sensor can be manufactured in different sizes, not only to cover fingertips but the whole hand or a whole robot arm [21].
In the middle of 2015, SynTouch announced the BioTac Single Phalanx or short BioTac SP. The sensor comprises the same sensory modalities as it's predecessor but in a miniaturized and more robust way. It also preserves the distal articulations of anthropomorphic robot hands. The BioTac SP is the smallest tactile sensor capable of imitating the whole range of sensory information like a human finger [22].
The recent development of SynTouch from 2017 is the BioTac Toccare, where they use the BioTac as well as the BioTac SP for quality control of textures and materials. A quality scale is shown in figure 2.4 [23].

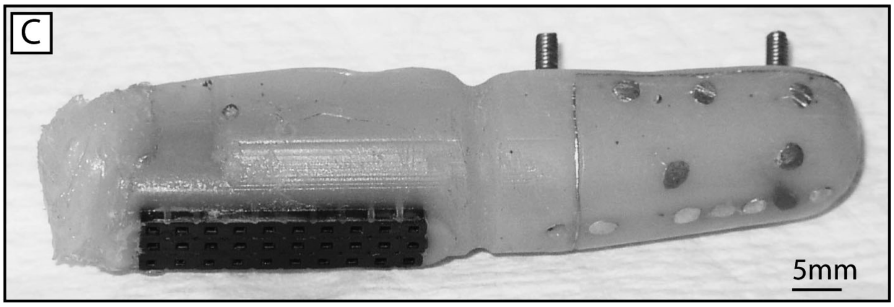


Figure 2.5: Early prototype of the BioTac core [20].

2.2.4 Other tactile sensors

Tekscan Grip System

The Tekscan Grip System, which is pictured in figure 2.6, is a tactile sensor built to be attached to the human hand and to measure static and dynamic pressures from grasping objects [24, 25]. The sensor contains 18 different sensing regions which can be placed all over the hand. One big advantage of this sensor is the paper-thin thickness and the universal size. These properties also allow using the sensor on an anthropomorphic robot hand. The force is measured as a change in electrical resistance between rows and columns of the conductive path on the sensor. The intersecting points are called sensels. Each sensel is electrically isolated which allows the direct measurement of the contact point or area. This is a huge advantage to the BioTac, where the electrodes are protected under the elastic skin which can't be mapped directly to a contact point.

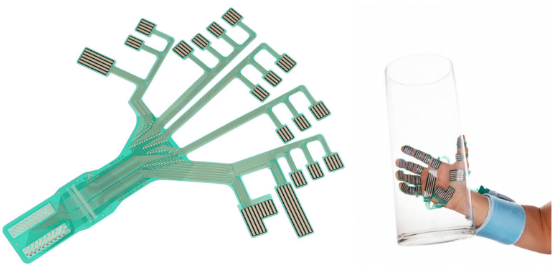


Figure 2.6: The Tekscan Grip System tactile sensor was built to attach 18 different scaled sensors all over a human hand or glove. Through the crossed conductive paths, it is possible to measure static and dynamic pressures and precise contact points or the area [25].

TacTip

The TacTip sensor is a biologically inspired tactile sensor built after the cutaneous model of the human fingertip. The sensor can be seen in figure 2.7. Its tip is a flexible half sphere which is empty. A camera inside tracks markers which are printed on the inner side of the rubber skin in

a specific pattern, see figure 2.7. This sensor is classified as an optical tactile sensor. The sensor is capable of detecting surface deflections in a direct, very sensitive and detailed way [26].
The disadvantage of this sensor against the BioTac is that it is not capable of sensing temperature or forces. However, it is able to detect shapes very accurately and due to it's flexible and large skin, it encompasses a bigger area of objects that assists in the stability of grasps and manipulation tasks. The TacTip sensors scale (40 mm diameter of the tactile finger pad) is larger than the BioTac. Furthermore, it is unsuitable for integration on a human-sized anthropomorphic gripper.

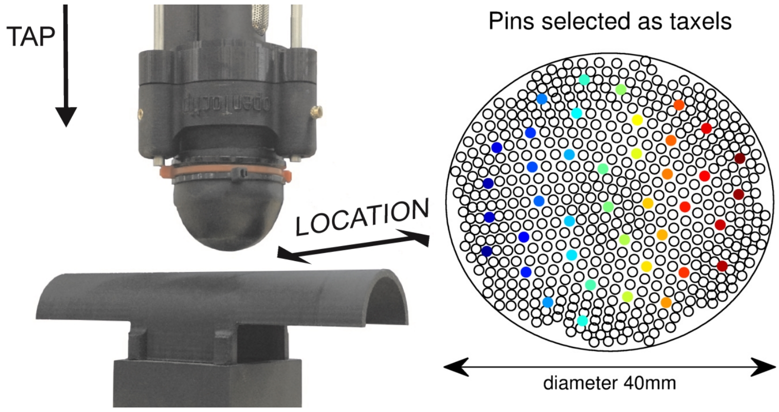


Figure 2.7: The optical tactile sensor TacTip, on the left side, and the inner pattern which is observed by a camera inside the sensor to detect surface deflections [27]. The sensor itself is larger than the BioTac sensors and not suitable for the use on anthropomorphic grippers. In the moment, due to the ability of precise contact location and deformation of the surface a smaller version would be interesting for human-like hands.

2.3 Anthropomorphic Gripper

Human hands have evolved in over hundreds of thousands of years, not only to grasp things but also to manipulate them in a dexterous way. With this hand, a human is capable of performing acrobatic manipulation, like a pencil rolling, sliding motions or the control of small tools. Hands evolved to handle a huge range of different objects, regardless of the size or the hardness. This evolutionary concept and dexterity are used for robot hands. The first developed anthropomorphic gripper was in the late 1970s by Okada. The robot hand was tendon-driven and had three fingers with the same size and number of phalanxes as a human hand [28].
To call a gripper anthropomorphic, it has to mimic the human hand partly or totally in the way of size, shape, movement possibilities or others. The term anthropomorphic does not directly accompany with dexterity, as for human-like robot hands, the dexterity could be quite different. On the contrary, a dexterous gripper is not necessarily human-like. The currently available or developed anthropomorphic grippers are all still dexterous [11].

In the following section 2.3.1, the Shadow Dexterous Hand used in this work is presented including the kinematics. In the last section 2.3.2, other currently available grippers are presented.

2.3.1 Shadow Dexterous Hand

The Shadow Dexterous Hand used in this work is motor-controlled and equipped with 5 fingers. Each of these fingers has a BioTac tactile sensor attached which replaces the last phalanx and the controllability of the last joint. The Dexterous Hand was built after the model of a human hand. It has the same size and nearly the same possibilities of motion. A picture of the gripper can be seen in figure 2.8.

Figure 2.8: The five fingered motor-driven Shadow Dexterous Hand with attached BioTac sensors on all five fingers. This gripper is used in this work to apply tactile based grasping.

Without the attached tactile sensors, the gripper has 24 Degrees of Freedom (DoF) which can be seen in figure 2.9. Each finger has four DoF, one in the last joint, one in the second and two in the knuckles. The first three rotation axes are aligned. The fourth one is orthogonal and allows a lateral movement of the finger. The little finger has one extra DoF at the side of the palm. The thumb of the Shadow Hand is quite different from the fingers, with the first two joints having aligned rotation axises in the closing direction of the finger. The second joint has one DoF extra for a lateral movement. In the last joint, the knuckles, there are also two DoF, one in the lateral direction and one aligned with the finger. Summed up, it makes 22 DoF with two in the wrist is 24. As mentioned with the attached BioTacs, the last joint of each finger is stiff and the hand has only 19 DoF left.
The Shadow Robot Company developed different versions of this anthropomorphic gripper. The Shadow Dexterous Hand is also available with air muscle controlled joints and also in a downgraded version with only four fingers, called Hand-Lite. Each of these versions is available with a custom number of attached BioTacs. Shadow also provides a left handed version [29, 30].

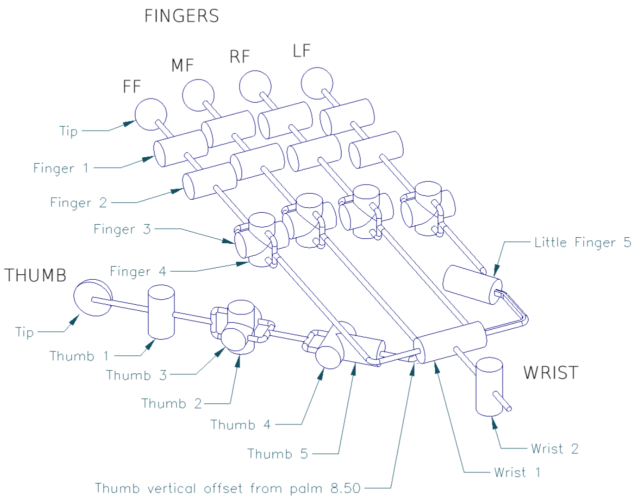


Figure 2.9: The kinematic chain of the five fingered Shadow Dexterous Hand. All rotational joints are pictured as cylinders. The overall number of DoF for this hand is 24 with 20 directly controllable joints. For each attached BioTac, the hand looses one degree, in this case the gripper has 19 DoF left [29]. On the right side the Shadow Hand is compared with a human to show the compliance.

2.3.2 Other anthropomorphic grippers

DLR Hand II

The *Deutsches Zentrum für Luft- und Raumfahrt (DLR)* published many anthropomorphic grippers, however they were all a bit bigger than an average human hand. These grippers were mainly built for space traveling, so their main requirement was the robustness. With that size and robustness, the grippers were heavier than other human-like grippers. A picture of the DLR Hand II can be seen in figure 2.10. On the left side, this gripper is 1.5 times bigger than the average human hand. The gripper has 13 DoF, three degrees in each finger and one in the palm for better dexterity. An advantage on the Shadow Hand is that all motors and sensors are inside the hand and not in an additional forearm [31].

Low cost 3D printed gripper

Dexterous anthropomorphic robot hands are often very expensive which is the reason why there are several attempts of low cost 3D printed grippers. One hand, published in [32], has nearly the same size and possible motions like the Shadow Dexterous Hand with 20 DoF and consequently also the size of a human hand. The gripper can be seen in figure 2.10. Here, the hand is tendon-driven and controlled by a pneumatic control unit. The big advantage of this hand is that all components can quickly be assembled and exchanged, which allows modification of all single parts. The hand is also equipped with the Tekskin tactile sensor all over the gripper.

Figure 2.10: The left hand is the DLR Hand II, a very robust anthropomorphic robot hand built for the use in space. The gripper is 1.5 times bigger than a human hand [33]. On the right sight, a low cost 3D printed anthropomorphic hand is shown. The hand has 20 DoF and nearly the same possibilities of motion than the Shadow Hand. The 3D printed parts of the hand could be easily assembled and exchanged [32].

2.4 Related work

The BioTac is one of the most sophisticated tactile sensors since its release in 2008. Several publications about the sensor were published afterwards. In [34] the raw sensor data is preprocessed in different stages. The first stage in this paper is noise filtering followed by drift compensation and normalization. In [35] the authors compared the ability of detecting microvibrations like slippage, as well as small transient events like a water drop on the fingertip of the BioTac sensor with the human sensitivity. Within different publications, the point of contact was determined in different ways. [36] and [34] calculate the contact analytically, [37] determine the location by learning different states of the sensor. Another application with this sensor is the determination of forces, this subject is handled in [37, 38, 39].

Also the curvature of contacted objects could be estimated from the sensor data, this is possible due to the deformable skin around the BioTac and is covered in [40, 37]. In [41] the authors identify objects by their compliance, texture and thermal properties using Bayesian exploration and were able to distinguish between 10 different objects with a 99% success rate. Also in [42], a tactile multi-modal (vibration and thermal) material identification approach was established, based on recursive Bayesian estimation.

In different publications, the tactile data is used to improve grasping. For example [43] used the tactile feedback to determine when to stop the closing motion of their gripper. In this case they used a Shadow Dexterous Hand, with three BioTac sensors attached to the thumb, the first and the middle finger. In [44] grasp adaptation was applied. Here tactile information is used to detect instabilities of a grasp. In this approach, the authors learned a policy to detect the mentioned instabilities. The sensor was used to detect perturbations. In [45] the authors used the BioTac sensor information to accomplish a handover of an object grasped with a Shadow Hand with attached BioTac sensors. The authors distinguished between perturbation forces and forces

meant to end in a handover.
Besides the robotic, the tactile sensors are also used for prosthesis. In [38], the authors evaluated the force, vibration and thermal information from the BioTac sensor on prosthetic limbs and it's benefits against a prosthesis without tactile feedback.
In 2011, the Shadow Hand was used to manipulate deformable objects, in this case a sheet paper. The paper deformation was tracked by markers printed on the paper. The flat lying sheet of paper was grasped by using visual guided bulging and pinching of the paper [46]. Since the 24 DoF of the Shadow Hand, recorded grasps are very complex. Grasp synergies for different precision grasps were generated in this paper [8]. The complexity of such human supervised grasps could be reduced to 2-6 DoF.

2.5 ROS - Robot Operating System

A huge problem in the field of robotics is the high amount of different hardware. This makes the reuse of software nontrivial and time consuming. This begins on the level of hardware drivers over scheduled controlling and resource conflicts up to high level collision free motion planning. This results in a wide range of currently used software. An overview to the time before ROS provides [47]. Most existing frameworks were developed to match one specific purpose, which was established in the design process and made platform-agnostic difficult.
ROS was designed to get a framework independent from the robot hardware and is the product of a large-scale integrated vision of developers. The first idea was to build a framework for a specific set of challenges, yet the resulting architecture design makes it possible to extend the software to different tasks[48].
Up until today, the core ROS system is released as a distribution similar to Linux that gets releases every year [49].

2.5.1 Brief History

The first architectures underlying ROS were created at the Stanford University in the mid-2000s. Projects like the Stanford AI Robot (STAIR) [50] and the Personal Robots (PR) program [51], handling the integration of embodied Artificial Intelligence, developed prototypes of dynamic and flexible software systems. Later on in 2007 these software concepts were extended by Willow Garage [52]. They provided significant resources and created well-tested implementations to accomplish the development of this software and the creation of fundamental packages. The whole architecture was released under the BSD open source license which led to a wide use in the robotics community.
In the beginning, many different institutes used ROS for their own individual robots. The open source license allows maintaining own versions of the software and developers are free to make their repository public. Among others, this leads to a wide robot-agnostic use of ROS.

Today, the whole ROS community has grown to over tens of thousands users, spread all over the world. Not only research institutes use it, but also industrial companies or private people at home [53].

2.5.2 Design idea

The main goals of the ROS framework can be summarized as:

- Peer-to-peer
- Tools-based
- Multi-lingual
- Thin
- Free and Open-Source

Until the time it was developed, no other framework had such a set of criteria [48].

Peer-to-peer

In a large-scale robot setup like a service robot, commonly a number of different hosts connected with Ethernet or a wireless LAN profits if not all messages are shared through the network. Frameworks with a central server often result in high traffic and a slow communication between the hosts and the server. This problem will be avoided through a peer-to-peer topology. This architecture still needs a central lookup mechanism for communication. In ROS, this is called the *master*. This central service can run on any host and provides the routing of connection requests and broadcast information.

Multi-lingual

Different people have different preferences and different programming languages have different strengths. Initially the languages C++, Python, Octave and Lisp were supported, today there are also other alternatives like Java. However, actually used are just C++ and Python.
The peer-to-peer communication in ROS is negotiated and configured in XML-RPC, where implementations already exist for different languages. To support the exchange of information between the languages, ROS supports a language-neutral interface definition language (IDL). With this interface the developer is able to implement simple and complex messages for communication. These messages will automatically be expanded to C++, Python, Lisp or Octave files which handle the communication and prevent errors.

Tools-based

To manage the complexity of ROS, different tools were implemented to support the developing of new components. Some examples for these tools are the visualization of the component tree, the peer-to-peer communication topology or navigating the package tree.

Thin

To provide the reuse of complex code fragments, algorithms and structures should be stored in libraries which makes it easier to export and import the provided functionality. This structure also allows using external code like drivers or other open source projects like OpenCV [54] for image processing.

Free and Open-Source

ROS itself is distributed under the terms of the BSD license. With this license every developer is allowed to use the code also in commercial or non-commercial ways. The complete source code of ROS is publicly available, which allows everyone to improve and develop at all levels of the framework.

2.5.3 File system

This section is the first part where the actual developing of a ROS package or service is described. Creating a new project needs a structure in the program code and in the dependencies. First of all, a ROS distribution is needed which is usually installed on the system. This distribution is the foundation and provides the most common libraries and resources needed to develop with ROS [49].
The following description summarizes the basic concepts of file system in ROS:

Package A package is the smallest unit in a ROS project as it organizes the software, such as runtime processes called nodes, libraries, configuration files or message and service types or others. A package is the smallest unit which can be released to a ROS distribution. Every package has at least a manifest and a CMake file.

Meta Package A Meta Package is used to organize a set of related packages. A Meta Package itself usually doesn't contain any source code and is not needed to develop a project. A Meta package has the same structure as a package with a *package.xml* and the *CMakeFile*, which includes additional information marking the package as a Meta Package.

Manifest The manifest of a package is named *package.xml* file. It includes information about the package itself, such as author, license, description, maintainer and the most important part: the dependencies to other packages, both for building as well as at runtime.

Make file Since the ROS distribution *Groovy*, the build-system used for compiling the source code is called catkin. Catkin needs a CMakeFile which declares build information like dependencies to libraries or executable nodes.

Source code The source code or the nodes are stored in a *src* folder for C++ or in a *scripts* folder for Python files.

Messages and services Message and service descriptions are stored in the appropriate folders. The meaning and the use of this files are described in the next sections.

2.5.4 Computation graph and communication

The peer-to-peer network of ROS consists of different functionalities to communicate between the nodes like *topics* or *services*. The *master* manages the communication, for example through establishing the communication. Figure 2.11 shows a simplified diagram of the communication graph. In the description below, the single concepts are presented [49].

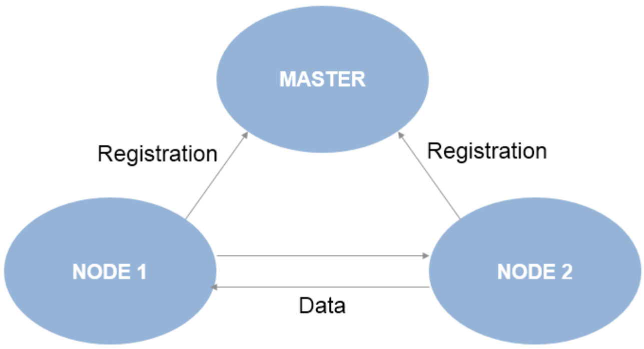


Figure 2.11: This simplified graph pictures the communication between the nodes in the computation graph [55]. Each node registers all its published topics and provided services at the master and requests all subscription topics. Any new connection is established through the master.

Nodes Within a node, the computation proceeds. Different nodes can be started and stopped independently from each other. One node can, for example, evaluate the sensor data given by a laser scanner and provide them to other nodes. A node can't be started without a master. If a node was started it registers itself at the master and informs about the topics and services it provides and listens to, see figure 2.11.

Master The ROS master is the overall manager of the nodes and communication. In one computation graph there could be only one master. It is possible to have a graph with more than one master, this is called *multimaster*. The master provides a lookup to the rest of the graph so nodes are able to find each other.

Topic Topics are a way to communicate between nodes in a n-to-n relation. Every node can publish information and all other nodes can listen to it, if they want to. When a node is started, all topics this node provides are registered to the master. If another node wants to listen to that topic, it raises a request to the master to subscribe to the topic. If the topic is published from another node, the connection will be established. The communication proceeds now between these two nodes directly and not through the master, pictured by the data connection in figure 2.11. Topics are used in a one way asynchronous communication, for example to spread sensor data. The publishing node does not know how many nodes are listening to the topic and does not expect any response. The information shared over topics is encoded in message files, these files are programming language independent. Custom message types can be stored in the *msg* folder of the package.

Service Services work in a similar way as topics. However, the difference is that the node providing a service establishes the information only on request in a 1-to-1 connection. This communication is synchronous and blocking, meaning that if node A requests a service from node B, A waits for the response from B and only then continues with its process. For services, the connection is established over the master. The communication itself is accomplished just between the two nodes. The information exchanged with a service is determined in service messages that are stored in the *srv* folder of a package. This service message consists of one message for the request and one for the response.

Action An action is similar to a service, with the exception that it is non-blocking. Actions are usually used if the expected feedback can take a while, for example, a motion of the robot. The action server node provides continuous feedback about the current state of the requested goal. The connection is established similar to services and topics. Furthermore, custom messages can be stored in an *action* folder in the package.

Parameter server Nodes can store parameters on the parameter server. Other nodes can lookup these data structures by just communicating with the master.

2.5.5 Command-line tools

The ROS command-line tools help to compile and to run the developed code. Except for building the nodes, the commands are independent of the current position in the file system [53].

Buildsystem catkin Catkin is the official build system in ROS. It relieves the original build system rosbuild. Catkin is a consolidation of CMake macros and python scripts, which allows using *find package* makros [56]. Catkin generates executables, interfaces, scripts and libraries, especially for C++ files.

Roscore The roscore is equivalent to the master. The roscore is the first thing that has to be started to run a robot, with the simple command:

```
user@hostname:$ roscore
```

Without the core, no nodes can be started. To start the roscore, the environment variable *$ROS_MASTER_URI* needs to be set. This variable is a string of the form *http://hostname:11311* where the hostname is the IP address of the host the core will run. 11311 is the port used for the communication between different hosts. Changing the port allows running different cores in the same network.

Rosrun All nodes are typically executable programs, so it is possible to search around the file systems and start the nodes. Rosrun allows by the ROS file structure to search in specific packages for executables with the command-line:

```
user@hostname:$ rosrun PACKAGE EXECUTABLE [ARGS]
```

Rosrun starts exact one node at a time as long as the roscore is started before.

Roslaunch The command-line tool for running launch files is quite similar to the rosrun command used for nodes:

```
user@hostname:$ roslaunch PACKAGE LAUNCH_FILE
```

Again, the package including the file is indicated and the name of the launch file. Launch files are stored in a *launch* folder in a ROS package. Launch files are written in XML. It allows conflating different nodes and specify their arguments. It is also possible to put parameters on the parameter server and include other launch files. Launch files also start a roscore if no one is already running.

2.5.6 ROS integration Shadow Hand

To get an overview of the necessary packages for the Shadow Dexterous Hand and an impression of the complexity of such projects, the most important shadow hand nodes and topics are shown. Figure 2.12 shows a small section of all nodes and topics after launching the Shadow code. The whole graph is too large and includes many redundant topics. The square boxes are topics and the ellipses are nodes, the edges between nodes and topics indicate the subscriptions and publications of the nodes. If an edge pointing to the node, the node is subscribed to this topic. The outgoing edges indicate published topics.
The center of this graph is the *realtime_loop* node. This node manages, among others, the controllers. A controller is used to move a specific joint, each joint has it's own controller. To move one joint, a message with information about the movement can be published to the */sh_rh_ffj3_position_controller/command* topic. Then, the *realtime_loop* handles the movement. In this graph, only the topics for one joint is shown. In the real graph each joint has it's own controller topics. The *realtime_loop* also publishes all current joint positions on the */joint_state* topic. On this topic, the */robot_state_publisher* is subscribed. This node knows about all kinematics in the hand and creates a transform tree, which includes information about the current spatially relations between the joints. This information is published on the */tf* and the */tf_static* topics. Another node is the */teach_mode_node* which allows switching between different control modes e.g. position control and effort control.
Most nodes publish on the */rosout* topic. Rosout provides the console logging mechanism, like printing outputs on the terminal.
The last important topic in this work is the */rh/tactile* topic, the *realtime_loop*, which publishes the raw sensor values from the five BioTacs to this topic. This information is used later in the */tams_biotac_node* which is implemented in this work.

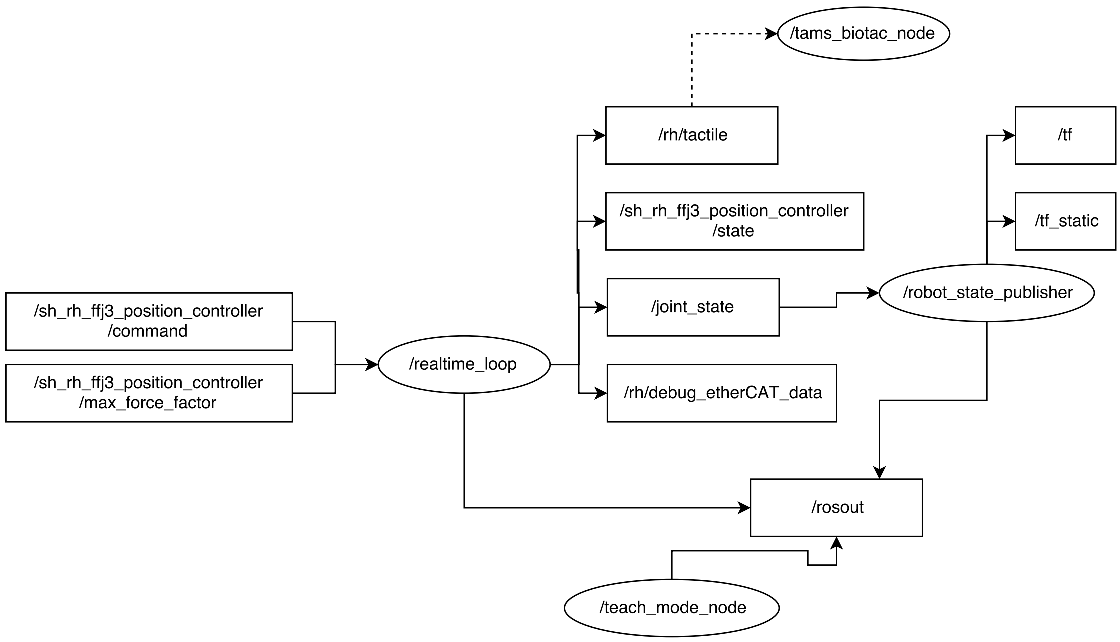


Figure 2.12: Important nodes (ellipses) and topics (squares) after a normal launch of the Shadow Hand code is shown. Topic subscriptions are represented as incoming edges, publications as outgoing. The */realtime_loop* manages all joint controllers used to move each joint and provides information about the state of the sensors and actuators, among others, on the */joint_state* and */rh/tactile* topic. The joints states are then processed by the */robot_state_publisher* and the transform tree is published. The tactile information is later used in the *tams_biotac_node*, represented as a doted connection.

3 Tactile sensing with the BioTac

In the first section (3.1) of this chapter, the different sensor types built in the BioTacs are introduced with their resolutions and equations to get physical units. The next section (3.2) looks at the data analyses and preparation. Problems and difficulties are mentioned like the changing temperature during a cold start. Through a temperature compensation, a filter and a normalization, the different problems will be handled. For these steps, the BioTacs didn't move or contact anything. The next step is to analyze the data for useful information such as point of contact [36, 39], radius of curvature of the contacted surface [37, 40], material properties of textures [42] or forces and torques [36, 37].
Since the target is to have a stable grasp with controlled applied forces, including tactile feedback, the interesting topics are the point of contact as well as the mentioned applied force at this contact. To measure the accuracy of any calculations with the sensor data, it is necessary to have reference values. A setup to measure such ground truth data is introduced, to determine the real contact point and the applied force. These values are independent of the tactile sensor.
As the force estimation might depend on the point of contact but not vice versa, the contact location will be determined first (3.4). Here, the weaknesses of the approach from [36] and [39] will be shown and an improved approach will be introduced. In the last section of this chapter (3.5), the normal force will be estimated, followed by a visualization of the sensor data during an applied tangential force.

3.1 Sensor data

The BioTac is a fingertip-sized tactile sensor built to mimic the human sense of touch. It is able to detect contacts, temperature and surface properties of touched objects. The sensors are placed in a rigid core surrounded by a flexible silicone skin filled with 1M Sodium Bromide dissolved in a PEG-200/Water mixture [20, 17].

The BioTac contains three different sensors, which measure changes when contacting any surface. The first one measures the pressure of the liquid in the skin. Furthermore, a high-pass filtered version of this data is provided, which allows measuring microvibrations. The pressure sensor is placed at the end of the rigid core, on the opposit of the fingertip. On the fingertip, the second sensor is placed, which measures the temperature of the BioTac. This value is also high pass filtered and gives information about the thermal flux. The last type of the sensor are 19 electrodes placed over the rigid core to detect impedance changes on different locations on the BioTac surface. All these sensor data depend on the temperature and the volume of the conductive fluid, which leads to data drifting. Due to this problem, the tactile sensor performs well in detecting changes but not in measuring actual physical units [17].

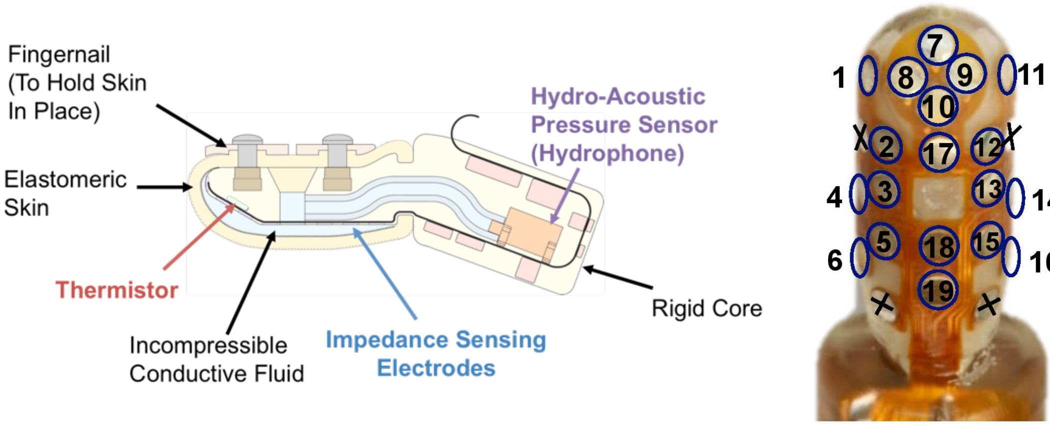


Figure 3.1: The cross section on the left side shows the technical structure of the BioTac. The pressure sensor is attached within the rigid core. The thermistor and electrodes are attached to the fingertip which is protected by an elastic skin filled with conductive fluid [17]. The electrode positions on the core are shown in the right figure.

Sensory Modality	Symbol	Range	Resolution	Frequency Response
Impedance	E_N	0 - 3.3 V	3.2 mV	0 - 100 Hz
Fluid Pressure	P_{DC}	0 - 100 kPa	36.5 Pa	0 - 1040 Hz
Microvibration	P_{AC}	+/-0.76 kPa	0.37 Pa	10 - 1040 Hz
Temperature	T_{DC}	0 - 75 C	0.1 C	0 - 22.6 Hz
Thermal Flux	T_{AC}	0 - 1 C/s	0.001 C/s	0.45 - 22.6 Hz

Table 3.1: BioTac sensory transducer sampling details [17].

3.1.1 Electrodes

The BioTac contains 19 electrodes aligned on the rigid core, which is under the flexible skin of the fingertip. The impedance sensing electrodes are shown in the right image of figure 3.1. In table 3.2, the Cartesian coordinates are listed. Besides the 19 electrodes, there are 4 excitation electrodes, whereby the impedance of each electrode can be measured in a voltage divider with reference to a $10\,k\Omega$ resistor. The excitation electrodes send a pulse with $3.3\,V$ through the conductive fluid. This pulse is measured at each electrode for one sample. The measured voltage depends on the amount of liquid between each electrode and the excitation electrodes. This is called the impedance. If a pressure is applied to the skin over an electrode, the fluid spreads over the skin volume and the impedance at this electrode increases. The measured voltage (V_n) has a resolution of 12-bit which makes up a range of 0 - 4095 for the electrode value E_n. With the equation from the voltage divider, the impedance (Z_n) can be determined:

$$Z_n = (\frac{3.3V}{V_n} - 1)10k\Omega = (\frac{4095bits}{E_n} - 1)10k\Omega \tag{3.1}$$

In figure 3.2, the raw sensor data of the electrode values of the first finger can be seen in the

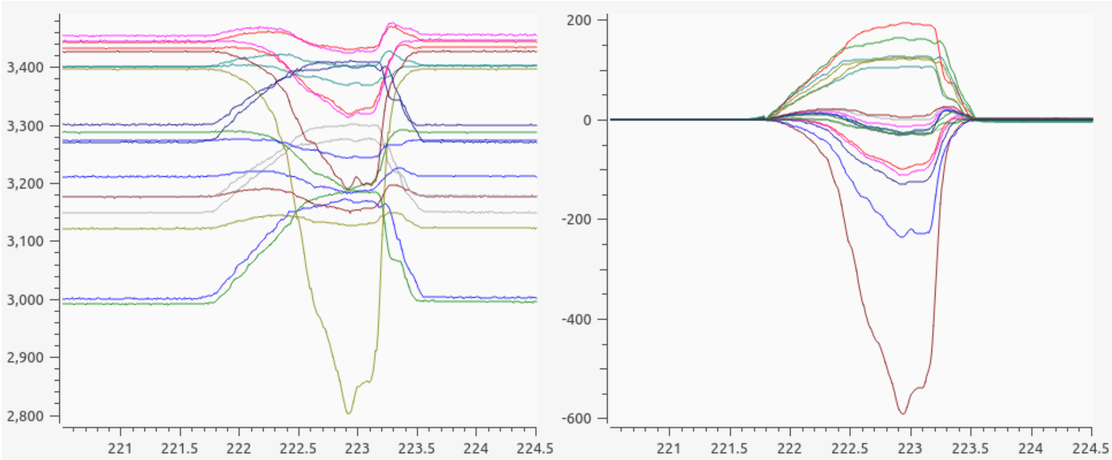


Figure 3.2: The left plot shows the 19 raw electrode values of the first finger during an applied pressure. In the right plot, the values are filtered and normalized. The spreading behavior of the electrode values is the result of the bulging BioTac skin.

left plot. In the first 1.5 seconds, the BioTac is in a rest state. Applying some force to the surface leads to the mentioned spreading of the fluid and the decreasing values of the electrodes under the contact. All other electrode values increase due to the rising fluid level. This effect is better visible in the right plot. In the rest state, the sensor data are tared to 0. In this case, the values at the applied contact will be negative.

3.1.2 Absolute pressure

The overall pressure of the fluid in between the elastic skin and the rigid core of the BioTac is measured with a piezo-resistive pressure transducer. This transducer is a Hydro-Acoustic pressure sensor (Hydrophone) that is placed on the opposit side of the fingertip protected in the rigid core (figure 3.1). The measured value of the pressure is in the range of 0 - 100 kPa with an offset to prevent negative saturation. To produce the pdc value, the pressure is also amplified with a gain of 10 and a low-pass filter at 1040 Hz. With a 12-bit resolution, the fluid pressure can be estimated by the equation below:

$$\mathit{fluid\ pressure} = (P_{DC} - \mathit{offset})\ 0.0365 \frac{kPa}{bit} \tag{3.2}$$

The *offset* value has to be determined when the BioTac is not in contact with external objects, and is used to tare this state. The absolute pressure values of all five BioTac sensors can be seen in figure 3.3. Due to the different amount of liquid in the flexible hull, the rest state values are quite different for the raw sensor data (left diagram). Also the magnitude of the pdc value is different for the same amount of force applied to the same contact point on the different sensors. This can be seen in the right plot with the filtered and normalized sensor data.

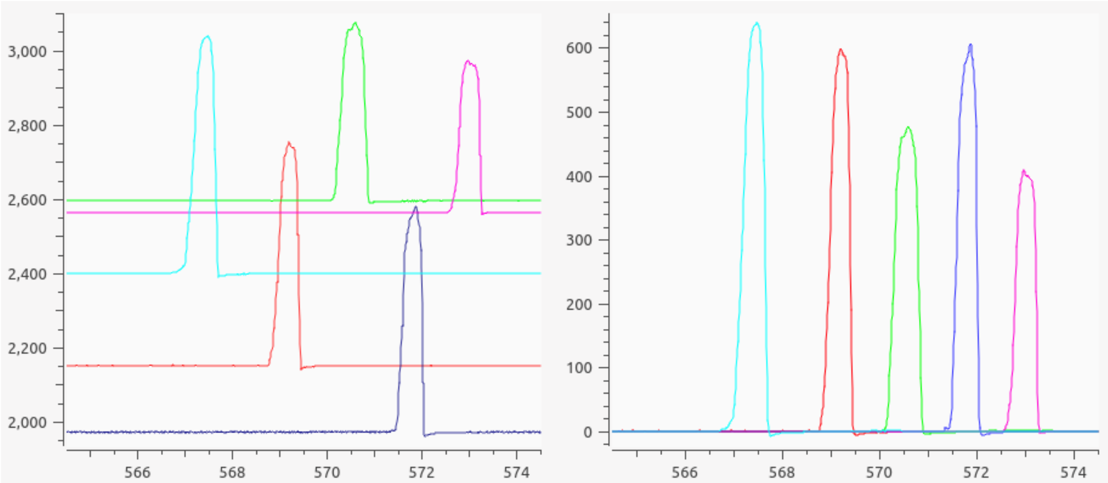


Figure 3.3: The pdc value of all five BioTacs during rest state and contacts. The left diagram shows the raw sensor data, the right diagram shows the filtered and normalized data.

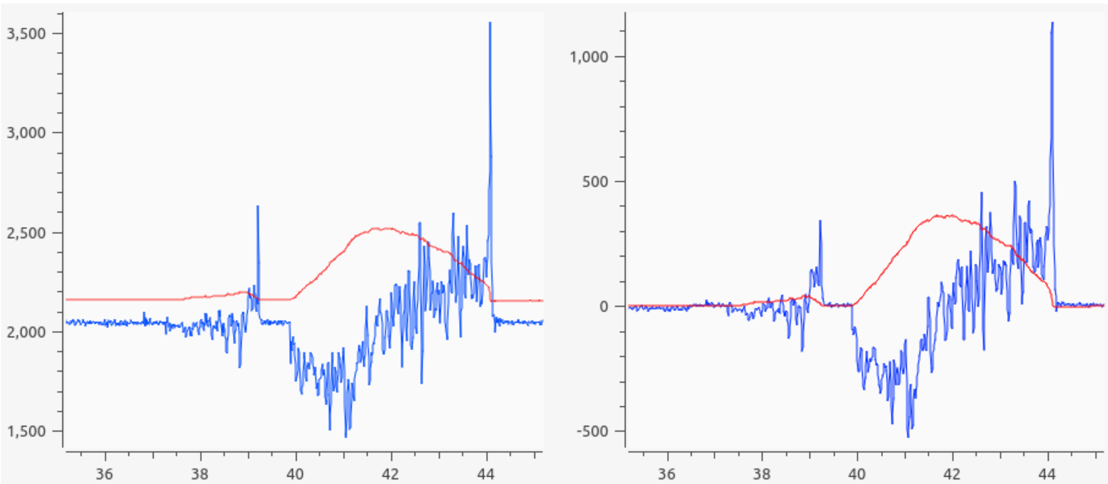


Figure 3.4: The pac sensor data of a BioTac during a sliding contact over the surface. On the left side the raw sensor data, on the right side the filtered and normalized data. In both diagrams, the pac value is plotted in blue, the pdc value is plotted in red.

3.1.3 Dynamic pressure

To measure microvibrations, the DC (pdc) signal gets an additional gain of 99.1 and is filtered between 10 - 1040 Hz. This gets the high-resolution AC pressure signal. The signal is similar to the pdc signal sampled with a 12-bit resolution and can be computed with the following equation:

$$\textit{dynamic pressure} = (P_{AC} - \textit{offset})\ 0.37 \frac{Pa}{bit} \tag{3.3}$$

The *offset* is used in the same way as in the dynamic pressure equation. Through the pac signal, the BioTac is capable of detecting vibrations as small as a few nanometers [35]. Figure 3.4 shows the pac and the pdc value during a sliding contact on the surface. In the graph between 38 s and 39 s a very soft contact is shown. The BioTac is touched without applying any active pressure, a contact can still be detected through the pac signal, whereas the pdc value has almost no deflection. In the next seconds, a sliding with a higher amount of pressure is applied which can be seen at the pdc value.

3.1.4 Temperature

With a thermistor voltage divider at the tip of the BioTac pictured in figure 3.1, the temperature can be measured with reference to a 30 kΩ resistor and a 10 V supply. The thermistor has a resistance of $0.6444^{4025^\circ K/T}\,\Omega$ specified by SynTouch in the product manual [17]. The signal is low pass filtered at 22.6 Hz to prevent aliasing. The following equation estimates the temperature:

$$temperature = \frac{4025}{ln(\frac{155183-46555\frac{T_{DC}}{4095bits}}{\frac{T_{DC}}{4095bits}})}\,^\circ C - 273.15\,^\circ C \tag{3.4}$$

The temperature in the BioTacs is increased with heaters to reach 31°C at the fingertip with an ambient temperature of 25°C. The temperature at the heaters is approximately around 85°C [57]. The thermistor itself inside the BioTac measures an equilibrated temperature of around 45°C. In figure 3.5, the tdc value is converted to degrees Celsius with equation 3.4. Three contacts are performed, the red curve is a cold cup with 15°C. The measured temperature needs a long time of over two minutes to stabilize. In comparison with an ambient temperature cup (green curve) there is a huge difference in the measured temperature. In both cases, the temperature falls and stabilizes. For the third contact, the BioTac is touched with a human finger measured with 32°C (blue line). First, the curve decreases, yet then rises over the idle temperature of the BioTac after 60 seconds and falls back to the idle state after the contact.

3.1.5 Thermal flux

Due to the heated device, the thermal absorptivity of contacted materials can be measured. Similar to the dynamic pressure, the dynamic temperature is band-pass filtered with 0.25 - 22.6 Hz and gets an additional gain of 98. The dynamic temperature is sampled with a resolution of 12-bit between 0 and 3.3 V and can be computed with the following equation:

$$dynamic\ temperature = \frac{-41.07}{ln(\frac{155183-46555\frac{T_{AC}}{4095bits}}{\frac{T_{AC}}{4095bits}})}\,^\circ C \tag{3.5}$$

In difference to the temperature value, the thermal flux provides fast information about the touched surface. This information can be used to distinguish between textures [41].

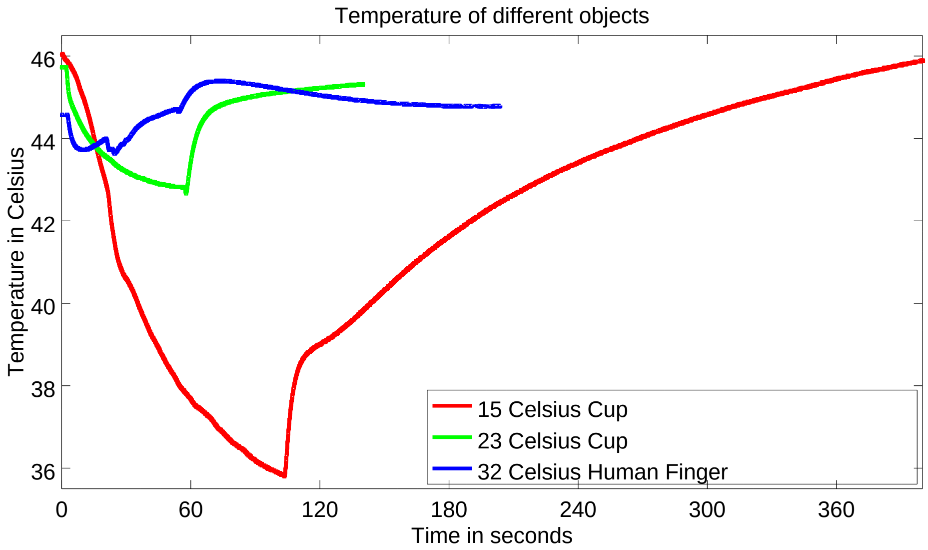


Figure 3.5: In the diagram, 3 surfaces with different temperatures were touched. The temperature is given in degree Celsius, calculated with equation 3.4. The temperature is measured inside the heated BioTac, which results in other values than the touched object on the sensor surface. The temperature for a warm object decreases first but then rises above the stabilized value. After releasing the contact, the measured temperature falls back to the initial value. For both cups the temperature falls and stabilizes after time, for the cold cup much longer than for the room temperature one.

3.2 Data analysis and preparation

In left plots of figures 3.2, 3.3, 3.4, and 3.5 the sensor data published on the *tactile* topic is plotted. On the first view, the data looks quite suitable, e.g. in figure 3.3 making and breaking contacts is easily detectable. When looking closer to the Y-axis and to the rest states, the values are quite different. On the one hand, all rest states are on a different level, however, on the other side the range is between 0 and 4095. This is the 12-bit resolution provided by the BioTacs. In order to get real unit values, it is necessary to normalize and calibrate the data. Another problem is the drifting effect of the sensor data influenced by the temperature (3.2.1). As written in the product manual of the BioTacs *"Similar to human fingertips, the BioTac is better at providing information about changes than absolute values."*, it requires some preparation of the data. This data will go through different stages to get suitable information out of the raw data. First, the data will be filtered for noise with the Exponentially Weighted Moving Average (3.2.2), which was inspired by [34]. The drifting problem for the electrode values is handled within the *subtract temperature* process. To deal with the different rest states of the values, due to the different amount of fluid in the fingertips (sec. 3.1), the raw data will be shifted to 0 and adjusted to eliminate the different rest state levels.

This will be done continuously in contrast to [34], where the data were just shifted to 0 at the beginning, which leads to huge problems when the BioTacs are not at operating temperature.

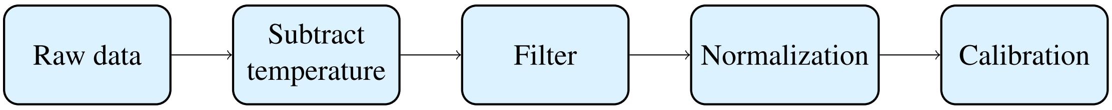


Figure 3.6: The preprocessing pipeline used to improve the sensor data. The temperature is compensated in the first stage, the second stage filters the noise. In the third part the data is tared in non-contact states and calibrated in the last stage.

This part will be called the normalization (3.2.3). The values cannot be normalized between 0 and 1 or -1 and 1 in this section since the minimum and maximum values are not known, they differ for every finger. This problem is faced in the last processing part, the calibration (3.4.4). This is a bit more complicated. The calibration cannot be made with the raw data alone because it needs some more data analysis beforehand.

3.2.1 Subtract temperature

The BioTac data is highly affected by the sensors' temperature. The first diagram in figure 3.7 shows the first 45 minutes of the temperature value from all five BioTacs on the Shadow Hand in degrees Celsius. The temperature of the thermistor at the beginning of a cold start is around 34 °C. After 45 minutes it increases to around 45°C ± 1°C for the different BioTacs.

Taking a look at the pdc and the electrode values during the same timespan, one can see that in the middle and the lower diagram of figure 3.7 depicts that the electrode values correlate with the temperature. The pdc values do not correlate with the temperature in the same way as the electrodes.

For the electrodes it is possible to compensate the temperature by using the following equation:

$$temp\_e_i' = e_i - tdc \cdot \alpha_i \tag{3.6}$$

With $temp\_e_i'$ as the new temperature independent electrode value, tdc is the raw temperature and α_i is the calibration value for each electrode. To get α_i, the recorded data shown in figure 3.7 was analyzed. The calibration value which leads to the smallest change in e_i' is used to calibrate the electrode at position i. The resulting α values for all five BioTacs are pictured in figure 3.8. Conspicuous in this graphic is that most values are mirrored on the vertical axis. In figure 3.9 equation 3.6 is applied to the electrodes during a cold start up, after the first 5 minutes the values are stabilized and temperature independent.

3.2.2 Filter

To reduce unwanted noise, the sensor data is partially filtered. For the microvibrations, as well as the thermal flux, it makes no sense to apply a filter at this point, since all changes should stay recognizable. In this work neither microvibrations nor the thermal flux will be used, maybe a noise filter can be applied at another point. The following filter will be applied to the electrode

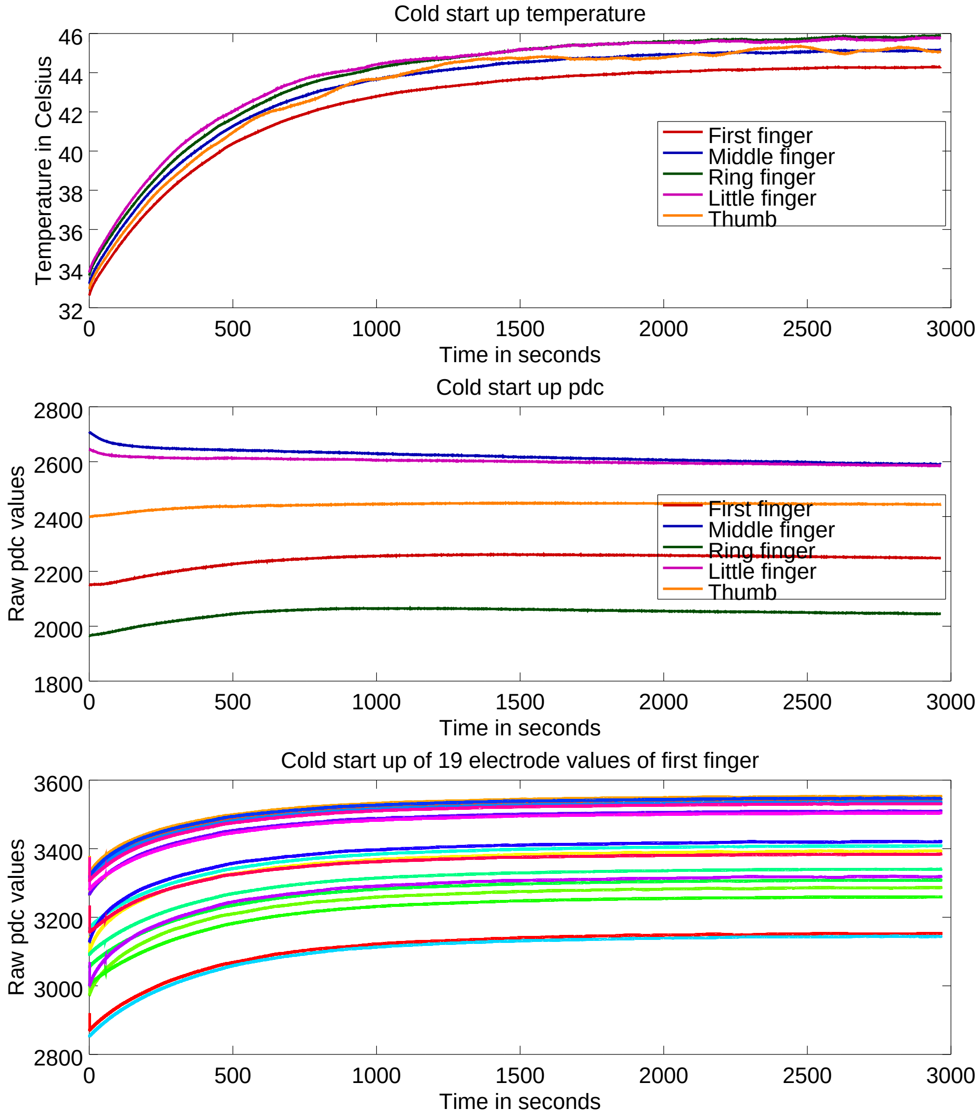
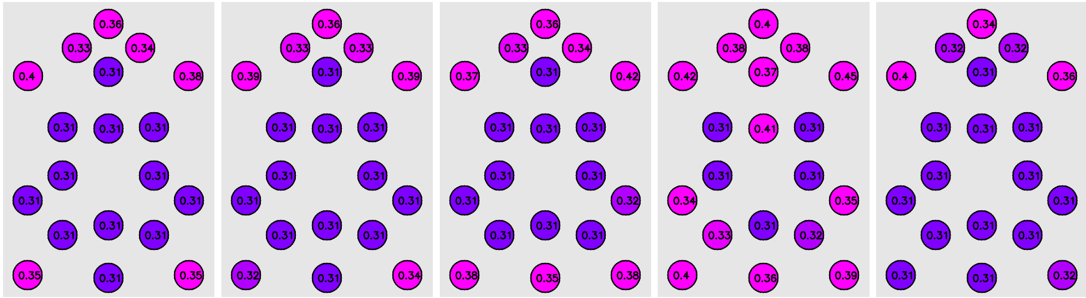


Figure 3.7: The sensor data used in the three diagrams is recorded for 45 minutes from a cold start to show the temperature dependency. The top graph shows the thermistor temperature. The middle graph shows the pdc values of all five fingers. The bottom graph shows the 19 electrode values of the first finger.

values, the pdc value, and the temperature to reduce unwanted noise:

$$EMA_n = \alpha \cdot x_n + (1-\alpha) \cdot EMA_{n-1} \tag{3.7}$$

This noise filter is called Exponentially Moving Average (EMA). This equation has the advantage that it does not need an amount of collected data to filter as it directly filters the first values. The EMA uses two values, the new incoming sensor data x_n and the old filtered data EMA_{n-1}.



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

Figure 3.8: The calibration values α_i used to subtract the temperature from the electrodes for all five fingers. (a) is the first finger, (b) the middle finger, (c) the ring finger, (d) the little finger and (e) the thumb. The electrodes on the fingertip and on the site of the finger are more influenced by the temperature than the rest of the sensor. The brighter the color of the electrode is, the higher is the α value at this position.

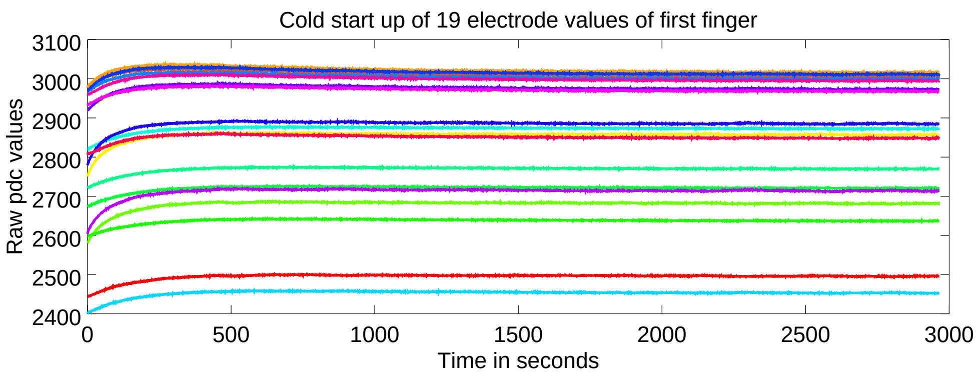


Figure 3.9: The temperature compensation is applied to the electrodes of the first finger during a cold start. After the first 5 minutes the values have stabilized and are temperature independent.

These two values are partially combined and create the current filtered data. How much the old value influences the current filtered data can be adjusted with α, a low α leads to a high influence of the old data that increases the smoothing of the data curves. However, this will also slow the reaction of data changes. For $n = 0$ the filtered data is $EMA_0 = x_0$.

3.2.3 Normalization

The sensor values of each BioTac are different. This concerns the idle state as well as a contacted external surface. The different idle values are clearly visible in figure 3.3 for the pdc values of all five BioTacs or in figure 3.2 for the 19 electrode values within one sensor. With the normalization, the idle state should be aligned, for the different amplitudes a calibration is necessary. The phenomena with the different rest state values appear due to the different amount of liquid in the fingertips. The manufacturer suggests a specific amount of liquid but it is quite difficult to insert the right volume. In the past, the liquid tended to leak out during tight or long grasps. This problem is mitigated with clamps around the leaking opening which can be seen in figure 3.10. The second problem being value drifting. This is already scaled-down by the temperature subtraction and is also handled by the normalization. To eliminate this problem, continuous calibration is necessary.

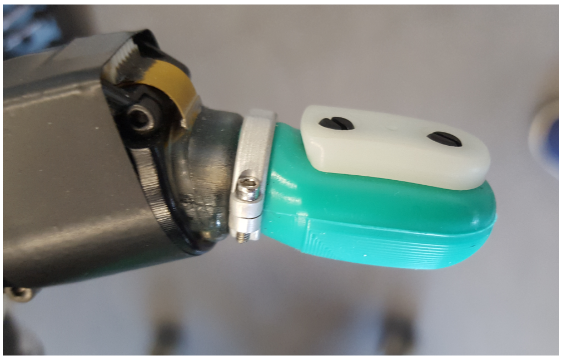


Figure 3.10: The BioTac with a clamp around the leaking area which prevents leaking liquid out of the silicone skin. The change in liquid resulted in a change of the pressure, which influences the sensor values.

The main idea of the normalization approach:

1 Shift all values to 0 when starting the normalization.

2 Detect idle state of individual BioTac, then shift to 0 again.

To realize this approach, some assumptions need to be made. First of all, a reference state is necessary to shift the values to 0, this state has to be a rest state which is used as an offset. This reference state or offset is subtracted from the raw sensor values and leads to zeroing of all values. To prevent the drifting, the reference state needs to be updated.

There are six different situations which could possibly appear and need to be handled:

1. Starting the normalization. For this situation, two different cases need to be handled. On the one side, the BioTac is in an idle state, the offset will be calculated correctly and everything is fine. On the other side, there is a contact or the hand is moving. In both cases, the calculation can fail and a false offset would be used. This will be detected and handled in the 5th situation.

2. The BioTac is in a rest or idle state. In this situation the offset can be updated to prevent the drifting phenomena. To detect the idle state it is necessary to know how the sensors behave in a resting state and non resting state. It is important now to distinguish between

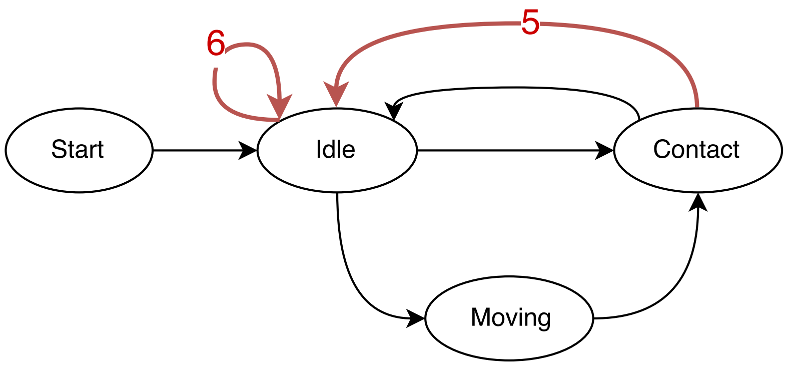


Figure 3.11: The state diagram of the normalization algorithm. The black paths represent normal state transitions. The red paths show transitions from a falsely assumed state to the correct state. Assuming a contact in an idle state leads to a stuck in the contact state. To get back to the correct state, transition 5 has to be made by hand or through the context e.g. after the release of a grasp. For transition 6 the assumed state is the idle state, after releasing a contact the pdc value will be negative and the values will be normalized, the assumed idle state is correct again.

value drifting and real changes due to an external contact. Considering only the pdc value first leads to a simple and naive approach: comparing each pdc value with it's successor and if the difference is below a threshold, the offset will be reset. The problem with this approach is that the tactile information is published with 100 Hz and even rapid impacts often result only in small changes of the value and the offset will always be reset. To improve this method the pdc value will be observed in time intervals. If the interval is selected appropriately this will work for rapid changes. With gently contacts the algorithm will still fail. Increasing the time interval will not solve the problem and could lead to other problems. To improve the method more than two pdc values will be considered. Averaging over a small amount of pdc values still recorded in intervals can improve the gently contact problem and prevent resetting at the wrong moment, e.g. at a short rapid impact. To avoid continuous resetting, the buffer of pdc values will be cleared after a new offset calculation and a new history of pdc values has to be collected. The longer the observed time span is the higher the threshold must be, which specifies the difference between drift and external impact. So the time span should be chosen with care. If the average of the pdc history is negative, the offset will also be reset.
A second way to improve the detection is to consider the electrodes. Again the interesting part is to distinguish between drift and impact. Characteristic for external changes is an increasing range of the values, with the drifting effect the range remains constant. If the pdc condition is satisfied, the range of the electrodes is computed and compared with a threshold. Only if both conditions are satisfied the offset is reset.
The adjustable parameters of the algorithm are the thresholds. One for the pdc, one for the electrodes and the interval together with the amount of observed pdc values. The result of the continuous offset calculation can be seen in figure 3.2.

Other ways to optimize the algorithm would be including the remaining sensor data: The pac value could be used to detect making and breaking contacts just as the temperature. Without a contact it is unlikely that the temperature will change rapidly, even less the thermal flux value.

3. The BioTac has a contact. During a contact, the offset for normalizing the sensor values cannot be updated. In the second situation, an idle state is detected, or in other words, if no idle state is detected the BioTac is either in a contact or the hand is moving (see the fourth situation). Making a contact always leads to an increasing pdc value and an increasing range of the electrode values, which breaks the idle condition. During the contact the drifting of the electrodes also takes place, especially when the contacted object has another temperature than the BioTac itself. Breaking the contact can lead to two different cases: Either the values hold the idle condition, then everything is fine and the offset will be adjusted after a few moments. Or the values drifted too much and the normalization still assumes there is a contact. Then situation six occurs.

4. Moving the hand around. When moving the hand, the sensor values behave similar to a very soft contact. An example is shown in figure 3.12. The sensor data changes due to the extrinsic forces acting on the liquid in the fingertips, like the changing gravity and the centrifugal force. In contrast to a contact the amplitudes of the sensor values are distinctly lower. A moving hand is comparable to a short, soft contact which normally leads to a correct idle state afterwards and therefore a working normalization. Nevertheless, it might happen that the computed normalization offset is not correct anymore and the rest state cannot be detected, again situation six occurs.

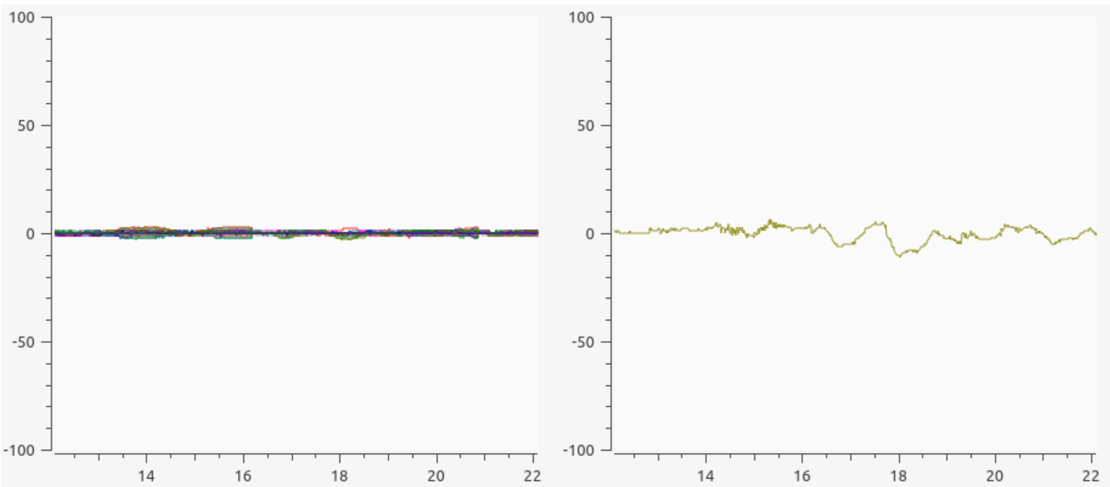


Figure 3.12: The figure shows the filtered and normalized sensor data of one BioTac while the hand is moving around. In the left diagram, the 19 electrode values are plotted, on the right side the pdc value.

5. True negative pictured in 3.11 as 5, assuming a contact in an idle state. This situation

might appear after long contacts, the values may drift over the threshold and the offset will not be reset. This phenomena is difficult to prevent. One possible solution is to use a ROS service to manually reset the offset after grasps. In the worst case, this service has to be called by a human.

6. False positives pictured in 3.11 as 6, falsely assuming an idle state. In this case, all values will be normalized even if there is a contact. This state is really unlikely and could only appear in a very slow and soft contact. Even if such a contact will appear, the values will be normalized after releasing the contact due to a negative pdc value.

3.3 Ground truth

With the BioTac sensor, a lot of information can be gathered. Since the sensor doesn't provide physical units directly, it is necessary to calculate them out of the prepared raw values. In the following sections, the point of contact, as well as the normal force, will be estimated. To improve and evaluate the accuracy of this information the ground truth for the contact location and the applied force has to be found.
First, the experiment setup will be described, in the second part of this chapter the process of data recording will be described.

3.3.1 Experiment setup

To measure the applied force on the surface of the BioTac, an ATI Nano17 force torque sensor is used (*1* in figure 3.13 and 3.3.1). To apply the force to a small area on the surface, a thin 3D printed cylinder is mounted on top of the sensor. This cylinder has a diameter of 4 mm and a spherical tip, which contacts with the BioTac. The idea is to use the cylinder to access only one electrode at a time. Of course, it is not possible considering just one electrode, but with such a small contact area the value of the underlying electrode will be much higher than the surrounding ones.
For the determination of the location of the contact point, two apriltags are used. These tags were created to estimate their positions and orientations in the scene by observing them with a camera [58]. One apriltag is fixed to the force torque sensor (*2* in figure 3.13), the other one is fixed to the back of the BioTac sensor (*3* in figure 3.13). With a simple webcam from above the scene (*4* in figure 3.13), the transform between these two tags can be computed. The relations between tag one and force sensor as well as tag two and BioTac are measured by hand and implemented as a fixed transform.
This setup allows recording the real force and contact point and compares them to the BioTac sensor data.

Hardware

ATI Nano17-E The ATI Nano17 is currently the smallest commercially available 6-axis force torque sensor. It has a height of 14.5 mm and a diameter of 17 mm and weights 9.07 gram. The sensor has a resolution up to 0.0031 N [59].

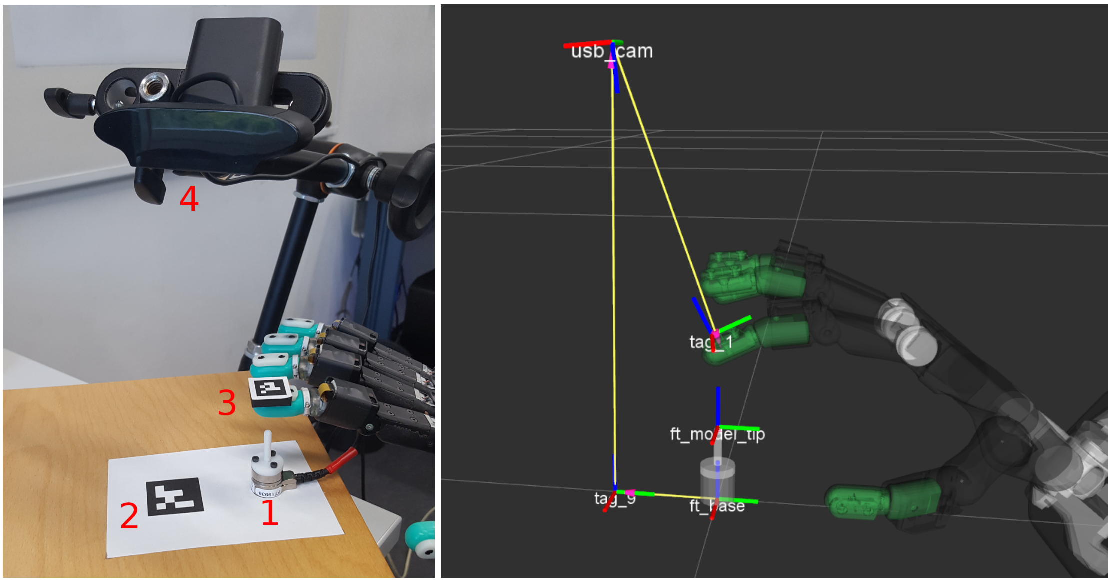


Figure 3.13: Ground truth setup: Force applied to the BioTac is measured with a ATI Nano17-E sensor (1). The point of contact is determined with one apriltag fixed to the force torque sensor (2) and one fixed to the BioTac sensor (3) observed by a Logitech Webcam C930e (4). The transform tree of the setup is visualized in rviz on the right side.

Apriltag Apriltags use a 2D barcode similar to QR-Codes to estimate 6-DOF position and orientation from a single image. These tags were created to get the ground truth of a controllable scene in an easy and fast way. Up to a rotation of 80 degrees, the algorithm still computes the pose with a very high accuracy. The apriltag detection of the scene is pictured in figure 3.14. For this purpose the tag family 16h5 was used. This type of tags have a low bit resolution and is suitable for small tags [58].

Logitech Webcam C930e This webcam is able to record HD 1080p (1920x1080) videos up to 30 frames-per-second. It has a field-of-view of 90 degrees and a 4x digital zoom. In this setup, the camera is used with a resolution of 1280x720 at a frame rate of 30Hz which results in a high accuracy of the transform estimation between force sensor and BioTac [60].

3.3.2 Data recording

With this setup, data sets can be recorded to test calculations with the data. In the next chapters, the contact point as well as the normal force will be estimated with the sensor data and compared with the ground truth. The raw tactile sensor data will be recorded, as well as the ground truth contact. The advantage of such a data set is, that the preprocessing, with temperature compensation, noise filter and normalization can be applied without the running hardware.

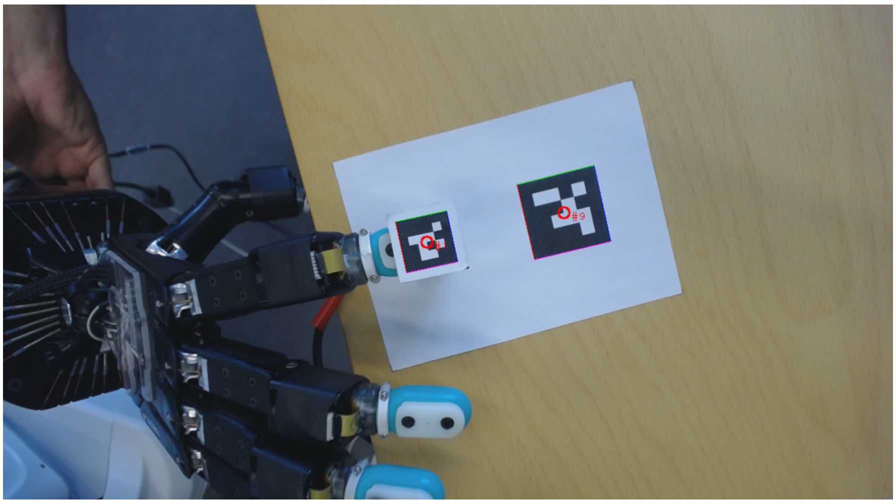


Figure 3.14: The ground truth setup from the view of the webcam. The apriltag algorithm detects position and orientation of the tags. From these poses the contact point can be determined.

To record this data *rosbags* can be used. Rosbags record all declared topics and allow replaying them. Within this recording procedure, the BioTac is pressed multiple times manually on the tip of the force torque sensor in a similar way as pictured in figure 3.14 and 3.13. The applied force is directed in the orthogonal way to the surface to create a normal force pressure. This procedure will be repeated to different locations and different amount of forces.
The recorded data is extracted from the rosbags and saved in csv [61] files to easily analyze.

3.4 Point of contact estimation

There are different approaches to solve the point of contact problem. On the one side, there are analytical ways presented in [36] and [39] on the other side this problem is often solved with learning approaches presented in [37]. In this work, the analytical way will be considered. First, the calculation from [36] and [39] will be presented (3.4.1) and evaluated, an improvement of this formula will be shown (3.4.2). Both methods will be compared to see why there is an improvement. Afterwards, the electrodes will be calibrated with different methods to achieve more accurate results (3.4.4). The calculations will be tested with the different calibration approaches and evaluated afterwards (3.4.5).

3.4.1 Analytical approach

The contact point is calculated with the weighted average of the Cartesian coordinates weighted with the normalized sensor values of the 19 electrodes. Equation 3.8 shows the calculation of the contact position:

$$\langle x_c, y_c, z_c \rangle = \frac{\sum_{i=1}^{19}(|e_{i^*}|^n \langle x_i, y_i, z_i \rangle)}{\sum_{i=1}^{19}(|e_{i^*}|^n)} \tag{3.8}$$

where x_i, y_i and z_i are the Cartesian coordinates of the electrodes with the reference frame depicts in figure 3.15. The coordinates are listed in table 3.2. The electrode positions are weighted with the absolute value of the normalized sensor value e_{i^*} exponentiated with n which affects the influence of the higher values. Low values of n involve weaker electrodes in the contact point calculation and make it imprecise, while high values of n lead to jumping of the position between the electrodes due to the high influence. In [36] the authors found out that 2 is a good value for n.

	Coordinates ($\overrightarrow{R_e}$)			Normal Vector ($\overrightarrow{N_e}$)		
	x (mm)	y (mm)	z (mm)	X (n_x)	Y (n_y)	Z(n_z)
E1	0.993	-4.855	-1.116	0.196	-0.956	-0.220
E2	-2.700	-3.513	-3.670	0.000	-0.692	-0.722
E3	-6.200	-3.513	-3.670	0.000	-0.692	-0.722
E4	-8.000	-4.956	-1.116	0.000	-0.976	-0.220
E5	-10.500	-3.513	-3.670	0.000	-0.692	-0.722
E6	-13.400	-4.956	-1.116	0.000	-0.976	-0.220
E7	4.763	0.000	-2.330	0.500	0.000	-0.866
E8	3.031	-1.950	-3.330	0.500	0.000	-0.866
E9	3.031	1.950	-3.330	0.500	0.000	-0.866
E10	1.299	0.000	-4.330	0.500	0.000	-0.866
E11	0.993	4.855	-1.116	0.196	0.956	-0.220
E12	-2.700	3.513	-3.670	0.000	0.692	-0.722
E13	-6.200	3.513	-3.670	0.000	0.692	-0.722
E14	-8.000	4.956	-1.116	0.000	0.976	-0.220
E15	-10.500	3.513	-3.670	0.000	0.692	-0.722
E16	-13.400	4.956	-1.116	0.000	0.976	-0.220
E17	-2.800	0.000	-5.080	0.000	0.000	-1.000
E18	-9.800	0.000	-5.080	0.000	0.000	-1.000
E19	-13.600	0.000	-5.080	0.000	0.000	-1.000
X1	-3.700	-4.956	-1.116	0.000	-0.976	-0.220
X2	-15.900	-3.589	-3.595	0.000	-0.706	-0.708
X3	-3.700	4.956	-1.116	0.000	0.976	-0.220
X4	-15.900	3.589	-3.595	0.000	0.706	-0.708

Table 3.2: Electrode orientation and location in 3D coordinates [36] in relation to the frame pictured in figure 3.15.

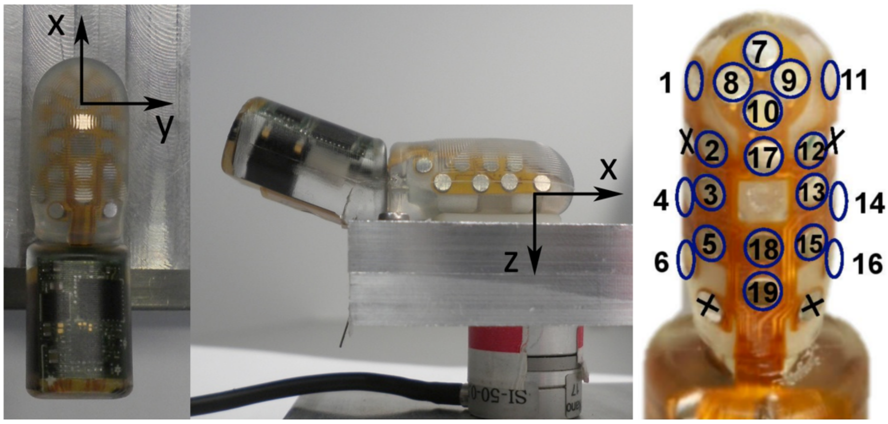


Figure 3.15: The BioTac sensor with transparent skin and visible electrode position. The origin of coordinate system used in table 3.2 is pictured [36]. On the right side the ordering of the electrodes is shown [62].

The outcome of equation 3.8 x_c, y_c and z_c is the contact point in relation to the Cartesian position of the electrodes. In the next step the contact points are mapped to the surface of the BioTac to get the real contact. This will be done with the following two equations:

$$\langle x_{c'}, y_{c'}, z_{c'} \rangle = \langle x_c, \frac{r \cdot y_c}{\sqrt{y_c^2 + z_c^2}}, \frac{r \cdot z_c}{\sqrt{y_c^2 + z_c^2}} \rangle \tag{3.9}$$

$$\langle x_{c'}, y_{c'}, z_{c'} \rangle = \langle x_c, y_c, z_c \rangle \frac{r}{\sqrt{x_c^2 + y_c^2 + z_c^2}} \tag{3.10}$$

For the surface mapping it is necessary to distinguish between two areas of the tactile sensor. The first part is the cylindrical finger area and can be computed with equation 3.9. The vector from the origin, which is the x-axis in this case, to the contact point in reference to the Cartesian electrode positions, will be extended to the radius r of the cylinder. In equation 3.10 the origin is the same as the reference frame pictured in figure 3.15. These equations were used in [36] as well as in [39]. Both equations have a minor mistake: the missing radius variable in the cylindrical part of the calculation. Since they are just considered the contact point in 2D, this mistake is maybe not conspicuous.

In figure 3.16 2000 example contact points are plotted, the green dots are the contact points estimated from the ground truth setup. The blue ones are calculated by equation 3.8 and mapped to the surface.

3.4.2 Improved analytical approach

To improve the analytical approach it is necessary to know and understand the weaknesses of the algorithm. The underlying idea of the approach is to average the Cartesian coordinates of the

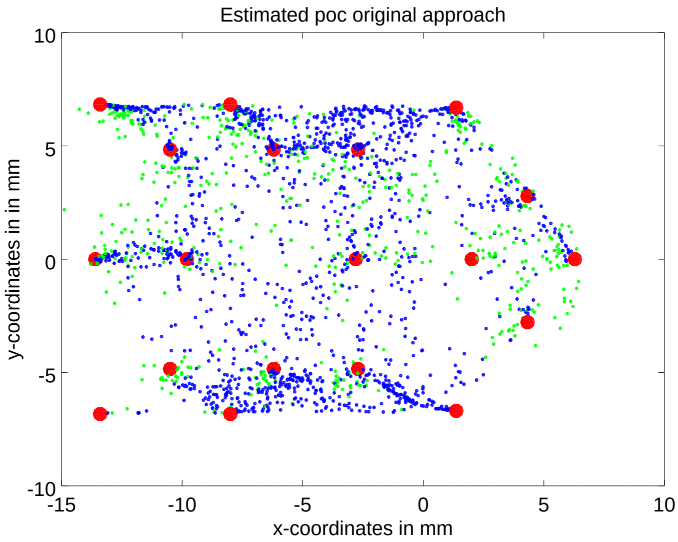


Figure 3.16: This plot shows 2000 contact points of the data set. The Cartesian positions of the electrodes mapped to the surface are pictured in red. The green dots are the true contacts on the BioTac surface, the blue ones are calculated with the point of contact equation from [36] and [39].

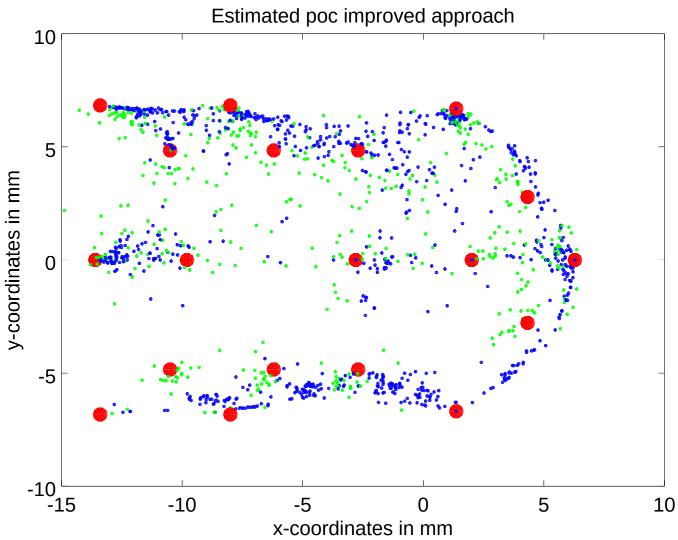


Figure 3.17: This plot shows 2000 contact points of the data set. The Cartesian positions of the electrodes mapped to the surface are pictured in red. The green dots are the true contacts on the BioTac surface, the blue ones are calculated with the improved point of contact equation introduced in this work.

electrodes. Without weighting, the resulting position will always be at the center. With weighting, the position will move to the electrodes with the highest value. One problem with this idea is that all electrodes need to have the same range of response values to an applied force. In reality the range is quite different. Some electrodes respond with a much higher value than others even if the same force is used, which means that these electrodes will have a higher influence on the averaged weighted position.
To deal with this problem, the exponent n was introduced, this is more or less a working solution. A better way is to calibrate the sensor data. For the point of contact estimation it is not necessary to calibrate or map the electrode values to real forces, but to calibrate them against each other or to normalize them between -1 and 1. This is involved with a bigger effort and will be regarded in section 3.4.4.

Another, more important weakness of this approach is the use of absolute values of the electrodes. This takes the difference between positive and negative impedance away. Or with other words: there is no difference between an applied force and a bulging of the surface equivalent to a force pointing away from the sensor. All impedances are considered as applied forces pointing towards the BioTac core.
Picturing the effect of positive and negative impedances: the average position is attracted by negative and pushed by positive impedances. Assuming, all electrodes have negative impedances, means all electrodes attract the contact points, which leads to inaccuracy and blurring of the contact points.

With the improved analytical approach, the second weakness will be examined. The idea is to ignore the positive impedance, since it is not possible to include negative weights in a weighted average equation.
This results in the following equation:

$$\langle x_c, y_c, z_c \rangle = \frac{\sum_{i=1}^{19}(|e_{i^*}|^n \langle x_i, y_i, z_i \rangle)}{\sum_{i=1}^{19}(|e_{i^*}|^n)} \quad with \;\; e_{i^*} = \begin{cases} e_{i^*} & for\ e_{i^*} < 0 \\ 0 & for\ e_{i^*} > 0 \end{cases} \tag{3.11}$$

This is the same equation as the original analytical approach with the additional condition that all positive weights will be evaluated as 0. With this condition the positive impedance of the bulging skin has no effect on the resulting contact point.

In figure 3.17 the ground truth contacts (green) and the estimated contacts (blue) are plotted with reference to the Cartesian electrode positions mapped to the surface. There are huge differences between figure 3.16 and 3.17, salient is that the estimated contacts from the original approach are much more spread over the surface. This behavior mirrors the assertion in the beginning of this section, all electrodes influence the average position which leads to this behavior. The estimated contacts of the improved approach are more bundled to different locations.

3.4.3 Comparison of both approaches

To compare the original and the introduced improved algorithm for contact location, both algorithms are tested with a test data set. This data set includes 87000 test contacts recorded with the setup described in section 3.3. These contacts are distributed all over the BioTac surface.
The algorithms are compared,by first overall accuracy, and distribution of error of the BioTac surface.

Accuracy

As mentioned, in both papers the authors used $n = 2$ as exponent [36, 39]. In the diagram pictured in figure 3.18 the average distance error is plotted for different exponents n in a range of 0.1 to 5.0.

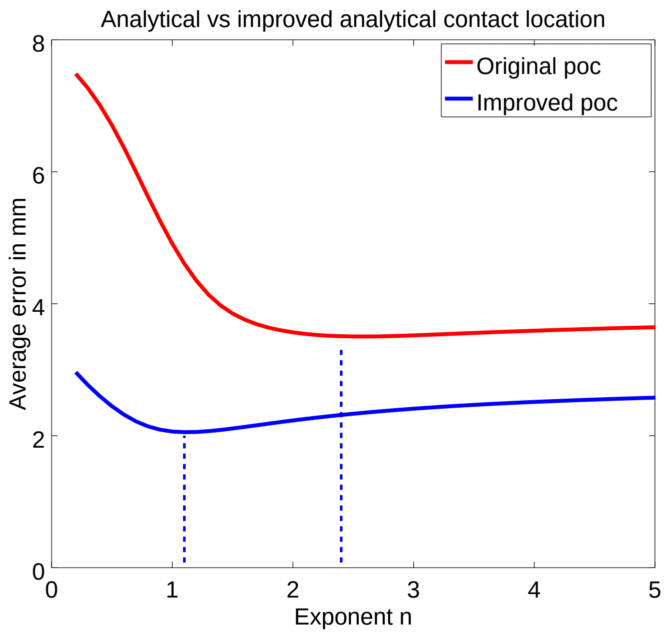


Figure 3.18: In the diagram the accuracy difference between the original analytical point of contact approach (red line) and the improved analytical approach (blue line) is shown. Both curves show the average error of the distance between the ground truth contact point and the computed contact point. The y axis gives the used exponent n for the equations. For the original approach the best exponent is $n = 2.4$ with the corresponding average error of 3.42 mm. For the improved approach the best exponent is $n = 1.1$ with the corresponding average error of 2.01 mm.

Both curves have a similar behavior, both start with a high error for a low n. Then decrease to a minimum followed by a slow increase. A primary aspect is that the blue line has a smaller average error. Taking a look at both minima, the accuracy of the improved algorithm is around

1.4 mm better than the original approach. In the next part, the error distribution of both algorithms will be observed.

Error distribution on the BioTac surface

The last section showed that the average distance error of the improved contact location estimation is way lower than the original one. In this part, the distribution of low and high distance errors on the BioTac surface is shown.
Figure 3.19 shows the difference of both approaches, the top plots are generated with the original approach, the bottom plots with the improved approach. The left diagrams pictures how high the error is on different locations on the BioTac surface. The right histograms show the occurrence of the distance error. For the corresponding plot as well as the color scale used. Both color scales have the same range to make them comparable.

For the original approach the error is particularly high at the fingertip and a bit noisy for the center of the finger. At these parts the error increases up to 13 mm which is nearly the half of the surface length. Such high errors do not appear for the improved algorithm.
Taking a look at the error histogram of the original approach the high amount of errors with over 6 mm is distinct. The reason for these high errors is shown in figure 3.20. In this figure, the sum of all 19 normalized electrode values is computed and plotted in a histogram to show the occurrence of a distance error lower than 4 mm on the left side and higher than 4 mm on the right side. The key aspect is the electrode sum for high errors, it is shifted to the positive side which means that the electrodes with a positive impedance have a high influence on the location. Computed contact points with a negative sum are quite more accurate due to the higher influence of electrodes with negative impedance. As a reminder, the positive impedance is a result of the spreading liquid in the BioTac skin, the liquid flows away from the contact point and produces positive impedance at electrodes far away from the contact location.
For the improved algorithm, the positive impedance is ignored and improves the accuracy of the computed contact location.

3.4.4 Calibration

To calibrate the electrodes and improve the contact point accuracy, three different calibration methods will used. The goal is to improve the influence of each electrode in a way that all electrodes equally influence the contact point equation. Currently, the values of some electrodes are higher for the same amount of force applied than others and thus a higher affect the location. The first two methods calibrate each electrode with all applied contact forces near the specific electrode. For the first method, the absolute force emitted from the force torque sensor is used. The second method uses just the proportional force distributed over all electrodes with a negative impedance. The third method doesn't use the ground truth setup. For this method the minimum values emitted from the electrodes are used to normalize the values between -1 and 1.

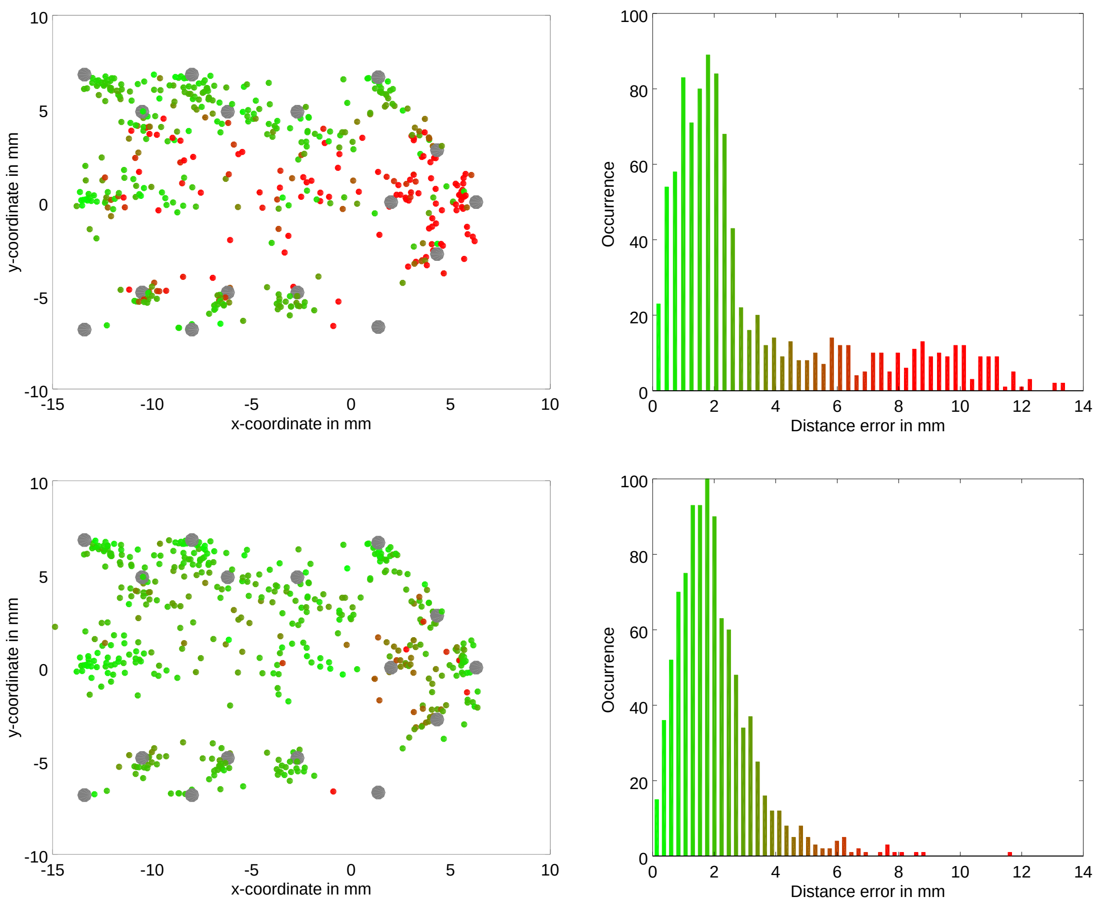


Figure 3.19: The diagrams show the position error of the contact location estimation on the surface of the sensor. The top diagrams use the original contact location algorithm, the bottom ones use the introduced improved approach. The calculated positions are compared with the true contact point to get the error distance. The right histograms show die occurrence of distance errors, as well as the color scale used in the left diagrams to show the height of the error. The Cartesian electrode positions are plotted as reference in gray. As shown, the error is higher for the original calculation, particularly high is the error on the finger tip for the sensor.

Calibrate electrodes with absolute force

For the first calibration method the transfer functions for each electrode will be generated with the electrode value in ratio to the absolute force applied to the BioTac. Each sample of the dataset is assigned to the nearest electrode on the surface. In the upper row of figure 3.21, the normalized data from electrodes 3, 14 and 16 are plotted against the absolute force measured by the force torque sensor in the direction of the z-axis.
The blue dots are single electrode value force pairs. Individual press iterations are clearly visible,

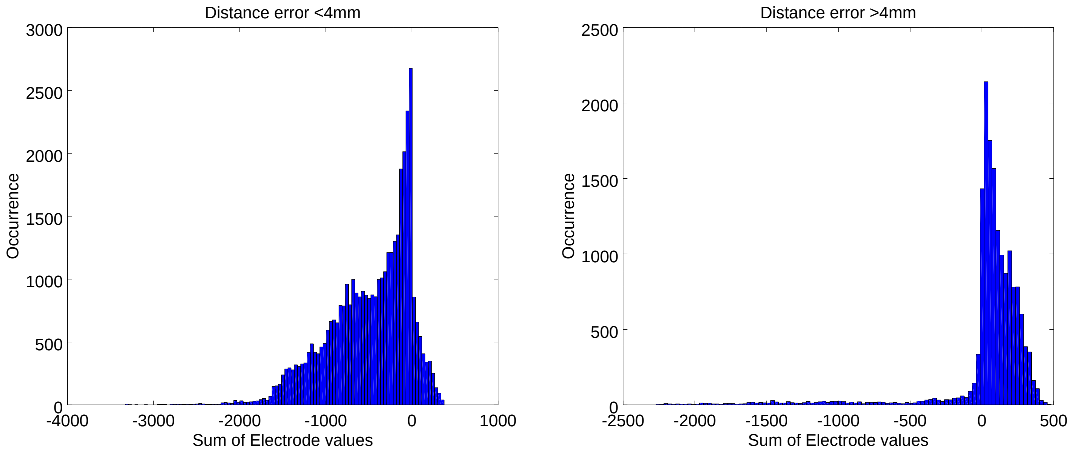


Figure 3.20: In both diagrams the 19 normalized electrode values are summed up for all samples of the data set. The left histogram shows the occurrence of sums for computed distance errors lower then 4 mm, the right histogram for errors greater then 4 mm. In both cases the original contact location approach is used. With this plots the weakness of the approach can be show, with a lot of positive impedance, the accuracy of the contact point increases.

at least for electrode 14 and 16 as lined points. There is a huge difference for the electrode values in different pressure iterations which could definitely lead to problems with the calibration. The red line is the transfer function calculated with a third degree polynomial interpolation. This function is provided by octave.

Calibrate electrodes with partial force

This calibration method uses the distributed absolute force over the electrodes with negative impedance. The electrode values are plotted against the proportional force of this electrode computed with equation 3.12, shown on the bottom diagrams figure 3.21. The main aspect is that the individual curves are less spread then in the first method, this is best seen for electrodes 14 and 16. Again the transfer function is calculated with polynomial interpolation. Compared to the first method the transfer function looks quite similar. If the use of proportional force is better than the absolute force, will show the evaluation later in this section.

$$F_{i\_partial} = \frac{e_i^* \cdot F_{absolute}}{\sum_{j=1}^{19} e_i^*} \ with \ \ e_{i^*} = \begin{cases} e_{i^*} & for \ e_{i^*} < 0 \\ 0 & for \ e_{i^*} > 0 \end{cases} \tag{3.12}$$

With e_i^* as the normalized electrode value and $F_{i\_partial}$ as the computed proportional force at electrode i.

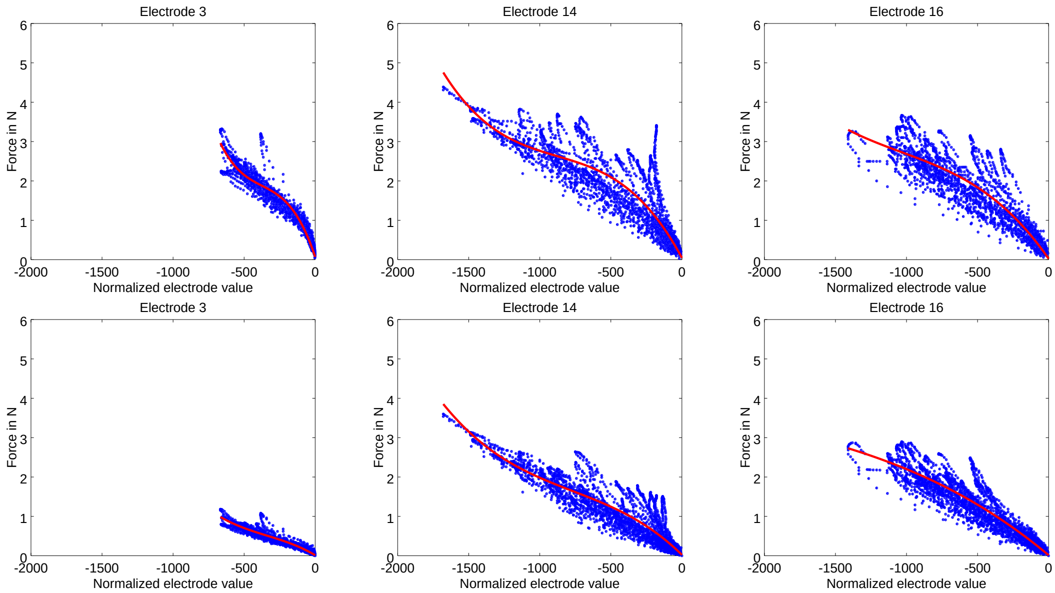


Figure 3.21: These plots show the transfer function for electrodes 3, 14 and 16 (red line). The functions are calculated with a third degree polynomial interpolation between the normalized electrode data at each electrode and the force measured from the ATI force torque sensor. In the upper diagrams the absolute force is used to calculate the transfer function, in the lower diagrams the force is proportionally split on all electrodes with negative impedance and only this partial force is used for the calibration.

Normalize electrode data range between -1 and 1

For the third calibration method, the normalization between the values -1 and 1, the minimum values of the electrodes are searched. Therefore the BioTac is pressed by hand several times all over the surface, till the core can be felt.
To normalize the sensor values of the electrodes between -1 and 1 it is sufficient to normalize them between the values 0 and -1. Since the absolute value of the minimum of an electrode is always higher than the maximum, the negative values will be in a range of 0 and -1 and the positive values in a range of 0 and 1 but will never reach 1. The following equation can be used:

$$e_{i\_norm} = \frac{e_i^*}{e_{i\_min}^*} \tag{3.13}$$

With $e_{i_{n}orm}$ as normalized value between 1 and -1. The accuracy of this method will be evaluated later in this section.

3.4.5 Evaluation

The results of the different calibration methods are shown in figure 3.23. In (a), the electrodes were calibrated with the absolute force (see 3.4.4). In (b), the partial force of the electrode

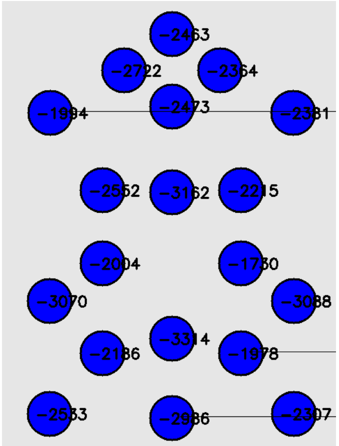
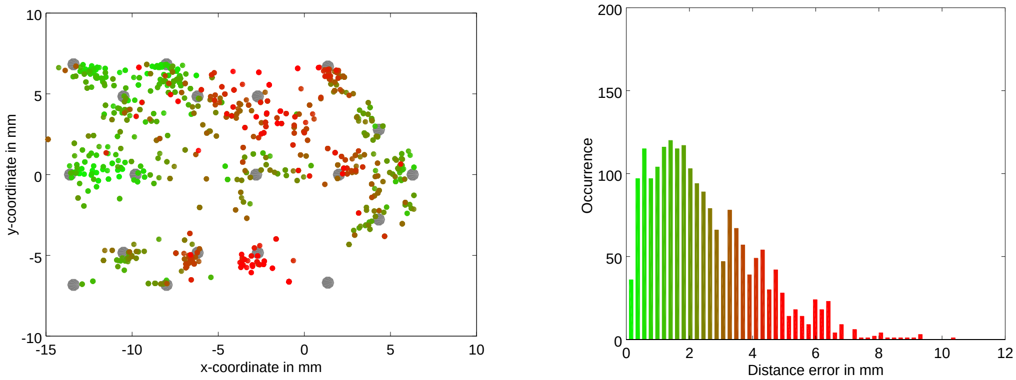
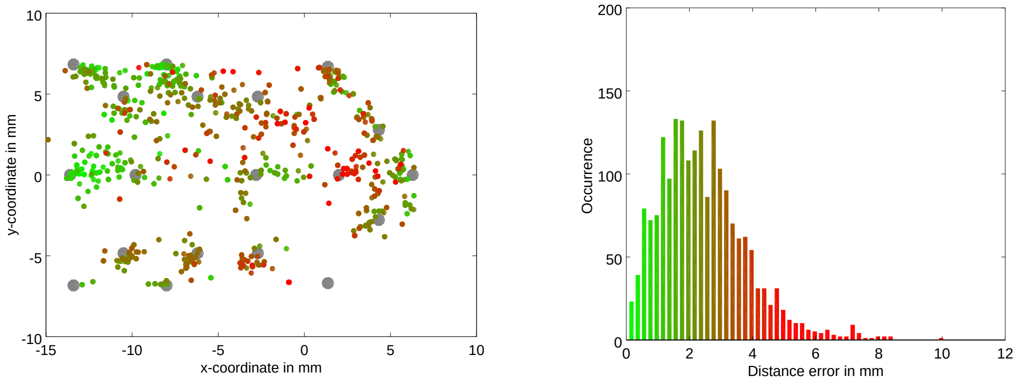


Figure 3.22: The picture shows the 19 minimum values used for the normalization calibration method of the electrodes.

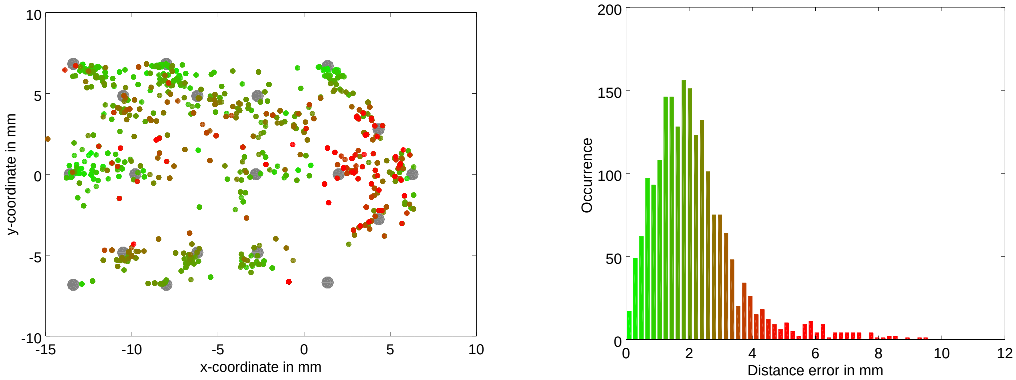
value, proportional to all electrodes with a negative impedance was used (see 3.4.4). The lower diagrams in (c) used the minimum value of each electrode to normalize the values between -1 and 1 (see 3.4.4). The histograms on the right side show the error distribution, whereas left diagrams show the distribution on the surface, colored with the hight of the error. The color scale can be seen in the right diagrams.
Comparing the three histograms on the right side, the last diagram has a way better distribution, with few errors higher than 4 mm. Both other histograms have a higher variance of the error. The distribution on the surface, shown in the left plots, is quite different for the three methods. The absolute force calibration has high errors in the center of the BioTac between -6 mm and 2 mm on the x-axis. The high errors for the normalization calibration mainly concentrate at the fingertip.
The best results achieved the normalization method but compared to the non-calibrated electrode values (see 3.19), the accuracy is almost the same. Reasons, why the force calibration didn't increase the accuracy, might be seen in figure 3.21. The electrode values spread very much. This might be a reason for the worse accuracy.



(a) Absolute force: 2.44 mm average error



(b) Partial force: 2.36 mm average error



(c) Normalization with minimal electrode value: 2.04 mm average error

Figure 3.23: The diagrams show the results of the contact point estimation with calibrated electrodes for the different calibration methods. On the right side the distance error distribution is shown. On the left side the error distribution on the BioTac surface is shown. The color of the bars in the right plot defines the hight of the error on the left side. The normalized calibration achieved the best results from the three methods compared to the not calibrated values (2.01 mm average error), even the normalization results in a worse accuracy.

3.5 Normal force estimation

The BioTac sensor provides two different ways to estimate forces. On the one side, the pdc value gives direct feedback of the fluid pressure inside the silicone skin, on the other side also the electrodes provide feedback of the applied forces on different positions of the BioTac surface. In the first part of this section, the normal force will be observed and calibrated. Afterwards, a possible way to compute tangential forces is shown.

3.5.1 Force response distribution

To get the real normal force applied on the BioTac in Newton, the pdc value will be used. The best case scenario would be, if the pdc value can be transferred with a simple transfer function to the real force independent of the contact location. A test dataset, pictured in figure 3.24, shows that this is not the case. The plot on the left side shows the Cartesian positions of the 19 electrodes. The light and dark green dots are contact points colored through their pdc values with an applied force of 1.4N to 1.5N. The color scale is plotted in the right diagram. The lighter green colors are higher pdc values, darker ones are lower values. As demonstrated, the pdc value is highly dependent on the applied force location.
In [39] all 19 electrodes, the absolute pressure, and the temperature were used to learn the estimation of the normal force within a calibration process. With this approach, the contact location can be ignored, since the normal force is determined from the whole state of the sensor data.

3.5.2 Calibration

The next step to a force estimation is the correlation between the pdc value and the actual force, therefore a transfer function is created. Different locations on the BioTac surface are observed. In this case, the known locations of the electrodes. For each location, the data is recorded for an applied force starting from 0N up to 5N. This dataset is plotted for five locations in figure 3.25. The five chosen locations represent areas around the BioTac surface. Taking a look at the pdc values in figure 3.24, makes it possible to make a rough classification into 5 areas. One is the fingertip, another is the side of the finger as well as the side of the fingertip. The center is divided into the middle and the side of the center.

Taking a look at the different plots in 3.25, for all areas the pdc value is saturated at 2N. For the range between 0N and 2N a linear correlation between pdc value and force is visible in all areas. This correlation can be computed with the transfer function plotted in red. The function is calculated by linear regression.

To cover the whole surface of the BioTac with a transfer between pdc and force, there are different possibilities. One way to do this is to calculate the transfer function for several points around the BioTac surface and create a lookup table. Another way is to just use a few transfer functions and interpolate between them.
In this work, 19 transfer functions were calculated, one at each electrode position. For a given

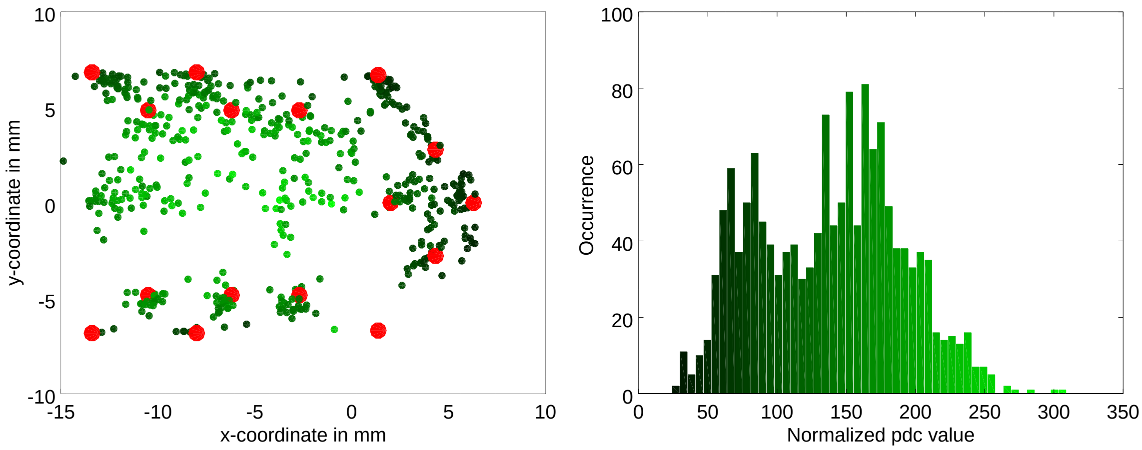


Figure 3.24: Both plots show the pdc value of an applied force between 1.4 N and 1.5 N. The right diagram is a histogram of the occurring pdc values, the interesting part is the color of the bars: The higher pdc values have a lighter green than low pdc values. The same color scale applies to the left plot where the corresponding contact locations of the measured pdc and force value are pictured. The big red dots are the Cartesian coordinates of the 19 electrodes. Salient is that the pdc values for 1.4 N to 1.5 N are much lower on the sensor tip and on the sides. The pdc value depends on the contact location. The distribution of the highest bars in the right plot depends on the data set and the amount of contact in the different regions.

contact point the actual normal force calculated with the following distance-based interpolation [63]:

$$F_N = \begin{cases} \frac{\sum_{i=1}^{3} \frac{1}{d_i} \cdot trans_i(P_{DC})}{\sum_{i=1}^{3} \frac{1}{d_i}} & for\ d_1 > 0 \\ trans_i(P_{DC}) & for\ d_1 = 0 \end{cases} \tag{3.14}$$

With F_N the normal force, d_i the distance to the nearest three electrodes and $trans_i$ the transfer function of the three nearest electrodes. With this function, the normal force is influenced by the nearest three locations. The three transfer functions of these locations are weighted with the distance to the respective contact point. If the contact point is directly on an electrode position, the distance is 0. In this case, the force value is calculated only with the respective transfer function.

With this method, the normal force can be determined for locations all over the surface.

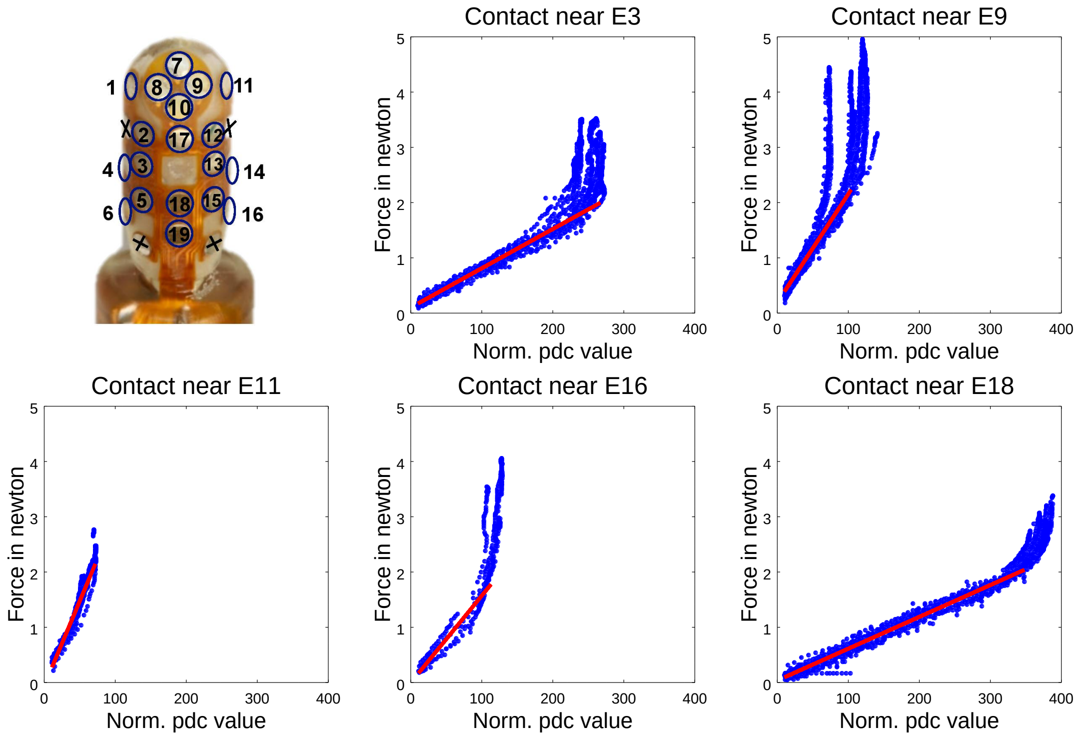


Figure 3.25: The top left picture shows the numbering of the electrodes on the BioTac core [62]. The remaining five diagrams show the force to pdc ratio at different electrode positions. Each of the five diagrams represents an area of the BioTac surface where the pdc to force ratio is similar. This can also be seen in figure 3.24. Electrode 3 represents the area between the center and the side of the fingertip (electrodes 2, 3, 5, 12, 13, 15). Electrode 9 represents the fingertip (electrodes 7, 8, 9, 10). The side of the fingertip is represented by electrode 11 (electrodes 1, 11). The side of the finger by electrode 16 (electrodes 4, 6, 14, 16) and the finger surface center by electrode 18 (electrodes 17, 18, 19). For each electrode, the transfer function is plotted for the range of 0 N to 2 N.

3.5.3 Evaluation

For the evaluation, a dataset of 32700 samples was used. Each sample contains the ground truth force measured by the ATI sensor, the ground truth contact point, the pdc value and the contact location calculated with the analytical analyzes of the electrodes presented in section 3.4. The force is estimated with equation 3.14 including the nearest three electrodes of the contact and their transfer functions. The example of a normal force estimation can be seen in figure 3.26, four contacts are shown here. The ground truth normal force is represented in red, the cyan curve uses the ground truth contact, the green one shows the integration of estimated contact location and normal force.

Both estimations don't fit the real force perfectly and the integrated force estimation achieves

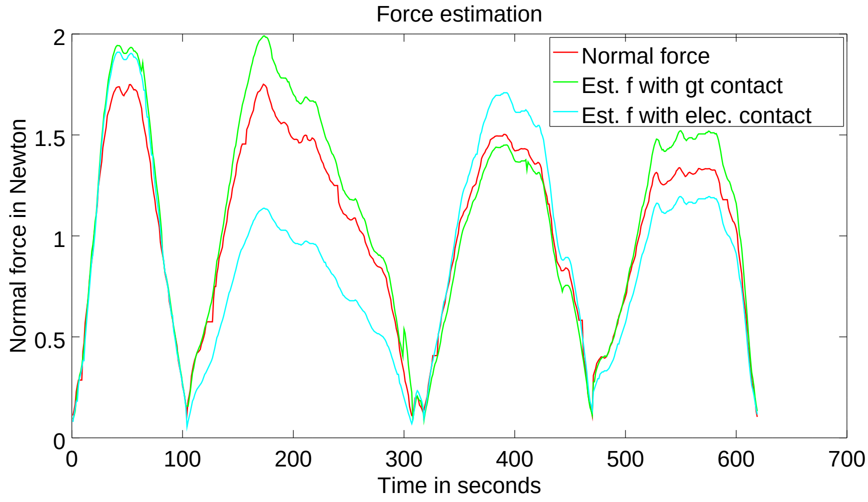


Figure 3.26: The force estimation for four contacts is plotted in this graph. The force measured by the ATI force torque sensor is plotted in red, the cyan curve is the estimated force at the actual position, the green curve used the contact calculation with the electrodes.

nearly the accuracy of the ground truth contact. The results of the 32700 samples are shown in figure 3.27, where the error distribution of both normal force estimations are plotted in a histogram. The left one shows the force estimation using the ground truth contact and reaches a mean error of 0.328 N. The right histogram shows the estimated normal force integrated with the contact estimation with the electrodes, the accuracy is 0.356 N on average, which is nearly the accuracy of the ground truth contact.
In [39] the author achieves an accuracy of 0.056 N by applying a learning approach to the whole sensor data. This analytical approach can't reach these results. Nevertheless, the estimated force can be used for grasping approaches now.

3.5.4 Tangential force

With the most array sensors, tangential forces can't be measured, since the surface doesn't change. Applying tangential forces to such a sensor would only result in a weaker measured normal force. With the BioTacs, the deformable skin allows seeing different results, than just a weaker normal force. The electrode values during an applied tangential force are shown in 3.28. With a fixed image of the electrode values, it is not possible to distinguish between a contact point from an applied normal force, which would lead to the red dot contact point shown in (b) and a tangential force with the contact point at the green dot. To estimate tangential forces, the electrode values should be observed over time.
In figure 3.28 (b), the electrode values are visualized. Here, the real contact point is at the position as the green dot and stays at this location on the surface. The tangential force moves the skin and also the contact to the side with the bulge (green dot). If the point of contact estimation

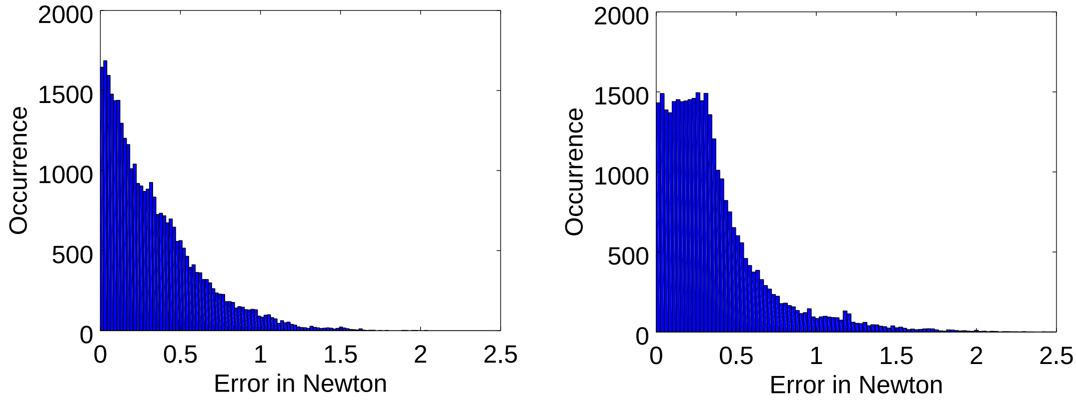


Figure 3.27: Both curves show the error distribution of the estimated normal force, the left diagram used the real contact location, the right one used the contacts calculated with the analyses of the electrodes. The left diagram has a mean error of 0.328 N, the right has a mean error of 0.356 N.

algorithm is used, the estimated contact point is located at the red dot. A tangential force can't be distinguished from a real contact at this location this situation, without knowing the context of the contact. This could be a quite problematic.

3.6 ROS package structure

To get an overview of the code used in this chapter, the ROS package structure is explained. The software is mainly split into two different packages, the *tams_biotac* package shown in figure 3.29 and the *tams_biotac_calibration* package shown in 3.31. Within *tams_biotac*, the preprocessing of the raw data takes place. This software will run during a normal usage with the Shadow Hand. In contrast to *tams_biotac_calibration*, this code will only be executed during the calibration procedure. Other packages used in this work which will not be necessary afterwards, are the *tams_biotac_evaluation* and some octave scripts independent of ROS. The *tams_biotac_evaluation* package contains only nodes to extract data from rosbags to csv files.

3.6.1 tams_biotac

This package provides libraries for the different stages of the preprocessing. To start the whole pipeline, the launch file *biotac.launch* can be used. In this case, each stage will publish its data on an own topic, for example to */rh/tactile_normalized*.

subtract_temperature This is the first stage the raw data passes. Within this phase, the electrodes values are set off against the temperature. The calibration file is read from the *.ros/biotac_info* folder. If this file doesn't exist, the sensor data will be published unfiltered. This part publishes to the */rh/tactile_subtract_temperature* topic.

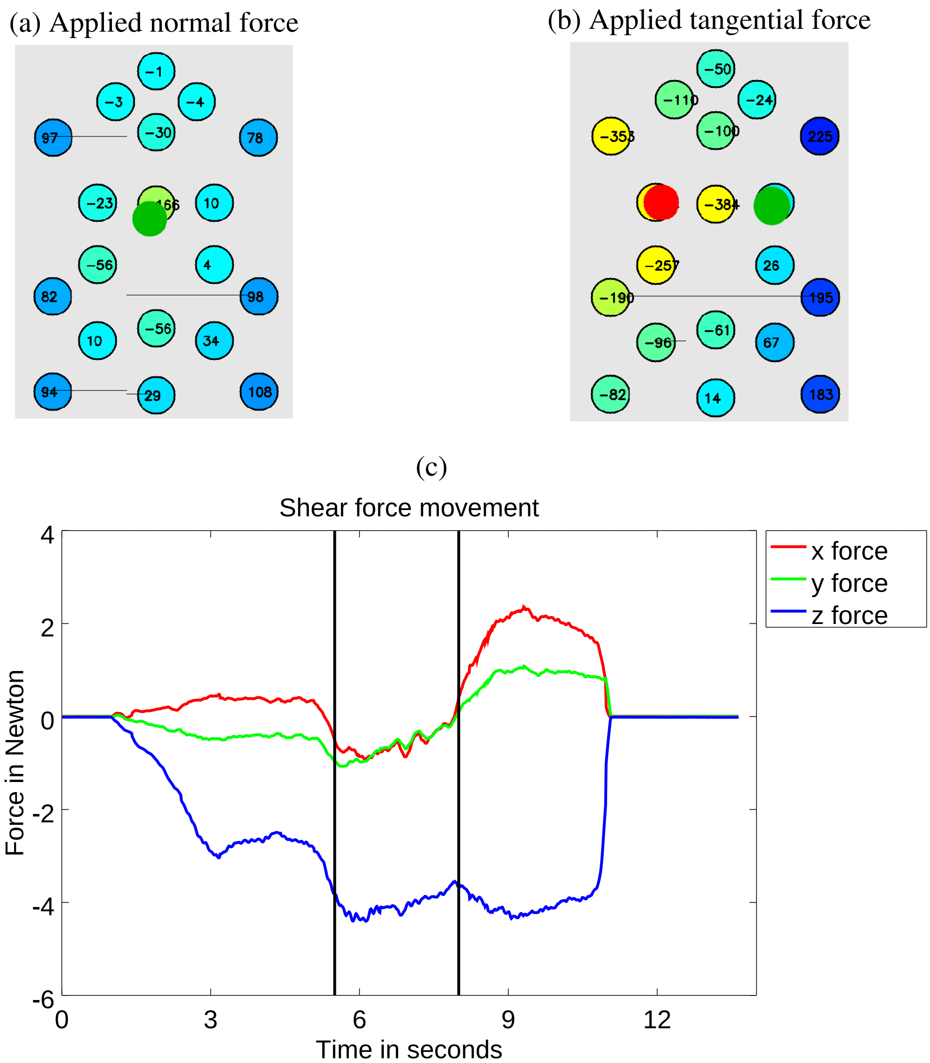


Figure 3.28: (a) The electrodes are visualized during an applied normal force to the surface at the location of the green dot. (b) A tangential force is applied at the same contact point shown in (a), the red dot visualizes the estimated location of the contact point algorithm, the green dot shows the actual contact point. (c) The three forces in x, y and z directions measured by the ATI sensor are plotted. In the first phase a normal force is applied, in the second and third phase, tangential forces in different directions are applied.

filter This library can be used to filter the raw data for noise with the EWA algorithm. This part publishes to the */rh/tactile_filtered* topic.

normalization The normalization shifts all values except for the tdc and the tac values to 0 in an idle state and takes care that the data drifting will not destroy the normalization. This part publishes to the */rh/tactile_normalized* topic.

contact This library can be used to analyze the electrode values to determine the contact location with the improved point of contact algorithm. It also provides the normal force estimation, since the force depends on the location of contact a calibration file with transfer functions is read from *.ros/biotac_info*. The information is published to the */rh/contacts* topic.

visualize_electrodes To visualize the electrode values in rviz, this library can be used. It will publish marker spheres for each electrode on the BioTac surface colored with the current value. This visualization is shown in figure 3.30.

tams_biotac_node This is the only node within the package. It can be used to bring up the whole pipeline.

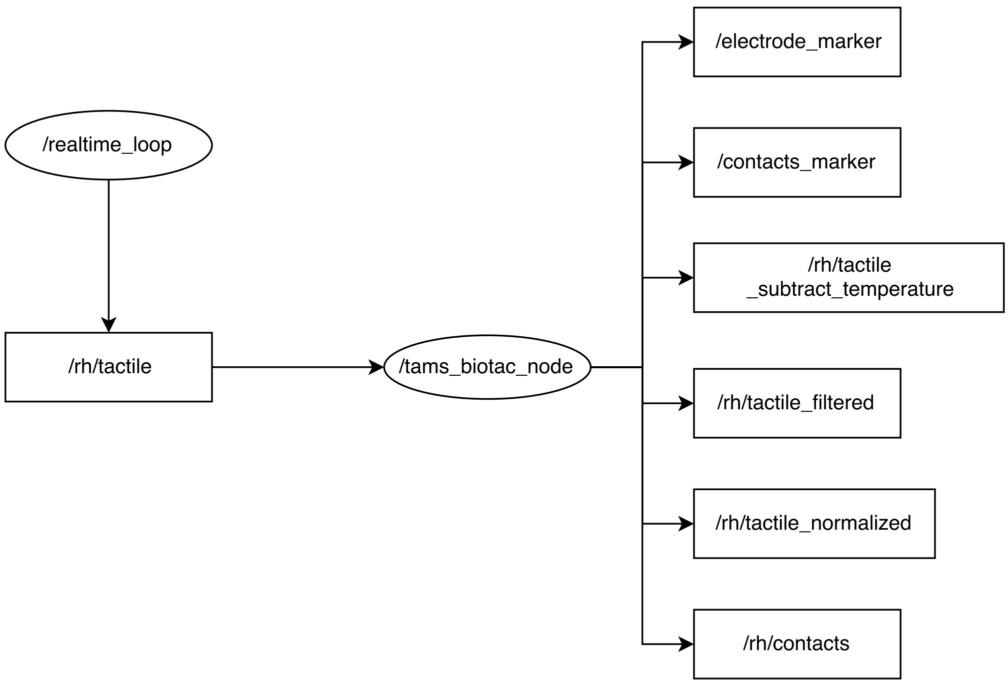


Figure 3.29: The node and topic structure of the *tams_biotac* package. Nodes are represented as ellipses and topics as squares, subscriptions are represented as incoming edges and publications as outgoing edges. The central node is the */tams_biotac_node*, which subscribes to the raw data on the */rh/tactile* topic, which is published from the *realtime_loop* of the Shadow Hand code. The single preprocessing stages are published separately on different topics, named after the stage. Additionally, contacts are published on the */rh/contacts* and the */contacts_marker* to visualize them in rviz as well as the electrodes on the *electrode_marker* topic.

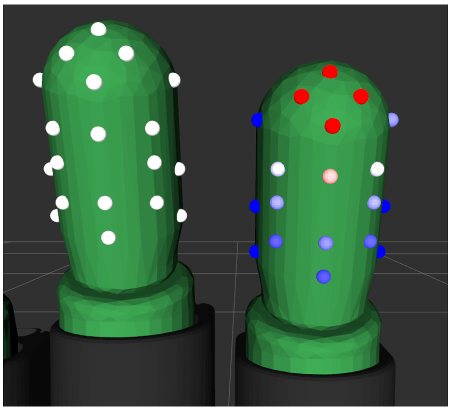


Figure 3.30: The visualization of the electrodes mapped on the BioTac surface. The colors indicate the impedance. If the impedance is negative, the electrodes are colored in red, for positive values the color is blue. If the BioTac is in a non-contact state, the electrodes are white.

3.6.2 tams_biotac_calibration

The *tams_biotac_calibration* is used to calibrate the pdc. The calibration process needs the *wireless_ft.launch* from the *tams_wireless_ft* package which publishes the sensor data of the ATI force torque sensor.

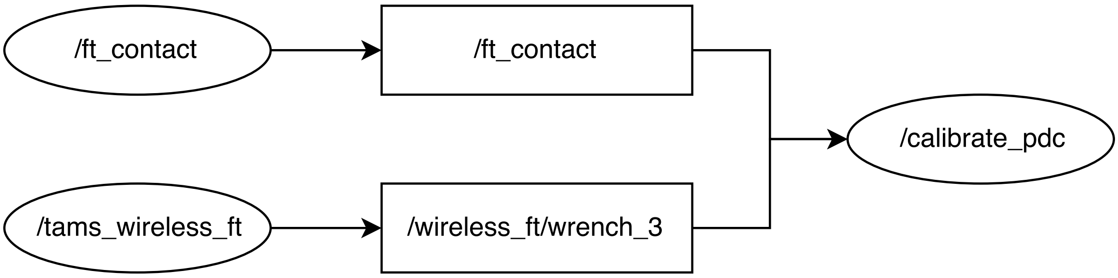


Figure 3.31: The node and topic structure of the */tams_biotac_calibration* package are shown. Nodes are represented as ellipses and topics as squares. Subscriptions are represented as incoming edges and publications as an outgoing edges. The ground truth contact point is published by the */ft_contact* node to the */ft_contact* topic. The force data from the ATI force torque sensor is published to the */wireless_ft/wrench_3* topic. The */calibrate_pdc* node subscribes to the ground truth topics.

ft_contact This node is used to publish the ground truth data. It can be started with the launch file *ft_contact.launch*. This will also start the apriltag detection as well as the webcam driver.

calibrate_pdc This node will record data for the calibration procedure and creates the calibration file in *.ros/biotac_info* after the process.

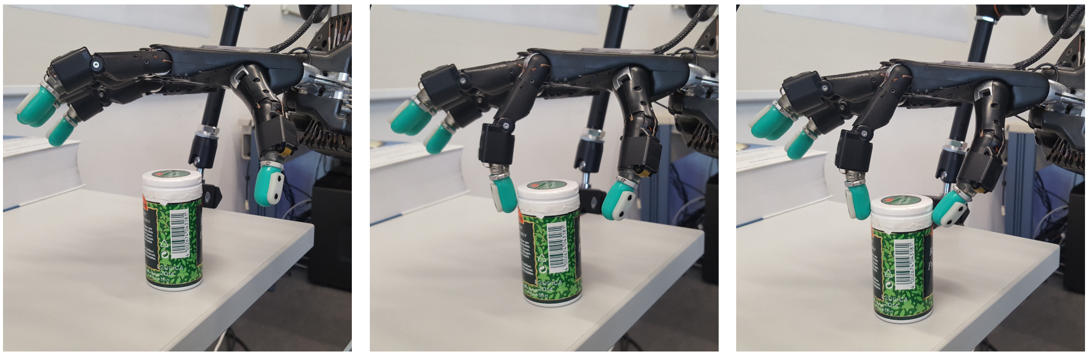
4 Tactile grasping - Proof of Concept

In the previous chapter, the tactile sensor BioTac was analyzed and the sensor data were preprocessed to include it in a grasp. With this preparation, the point of contact, as well as the absolute force, can be estimated now and used as feedback during a grasp. Currently, grippers without tactile feedback usually apply closing movements with a maximum motor force. In this case, fragile or deformable objects might break when the maximum force is not adjusted by hand for the specific object. Another problem is that the gripper does not react to the first contact, which results in pushing the object before a firm grasp is applied.
To include the tactile information in a grasping approach, it is necessary to know how the gripper is controlled and in which state the tactile information should be included. The requirement for a controlled closing or grasping movement is a fast reaction to contacts or environmental changes.

In this chapter, a way to include the tactile information of the BioTacs in a grasping movement controlled with the ROS framework *ROS control* [64] is introduced. First, the proof of concept procedure to include tactile information in grasps is described (4.1). A short introduction in control theory is given in 4.3. Afterwards, ROS control is presented (4.2). The implementation of a controller using tactile feedback is indicated (4.4) and the results of the proof of concept are shown in the last section of this chapter (4.6).

4.1 Proof of concept

To apply the gathered information about the tactile sensor, a simple proof of concept grasp strategy is presented to see the effect of the tactile feedback. In order to fulfill a firm holding of an object, at least two forces need to apply to the object in different directions. In this case, a two finger precision grasp could be used including the thumb and one additional finger. For this example, the first finger is used. To demonstrate the usage and the advantage of the tactile feedback, two fingers are enough.
To control the hand, a modification of the standard Shadow ROS control position controller is used. ROS control is introduced in the next section. The actual modification and implementation of the controller is described.
The strategy of applying the grasp is split into different phases pictured in figure 4.1, the first one is an open hand. To close the first finger, joint *J3*, in the knuckle, is requested to move to a specific position. The same applies to joint *J5* for the thumb at the same time. J5 is the nearest joint to the palm in the thumb. If one of the tactile sensors detects a contact, the respective finger stops it's movement. If both sensors detect a contact, the effort in the controlled joints are adjusted to increase the applied force on the object. To hold the commanded force, a controller handles the effort applied to the joints.



(a) (b) (c)

Figure 4.1: The different stages of the proof of concept grasping approach using tactile sensing are shown. This approach uses the first finger and the thumb and controls the respective joint in the knuckle to close the fingers. In (a) the hand is in an open mode, (b) both fingers close at the same time, in this case, the first finger recognizes a contact and waits for a contact at the thumb to apply a force. In (c) both fingers detect a contact, the force for a firm hold can be applied.

4.2 ROS control

The idea behind the ROS control framework is a robot agnostic way to control, manage and communicate with all actuators built in the robot [64, 65]. ROS control arose from the PR2 controller manager. In 2009 Willow Garage [52] developed the Personal Robot 2 and implemented a controlling structure for the robot. In 2012, the project was continued by hiDOF and PAL robotics, which created a robot agnostic version called *ROS control* based on the old PR2 code. The framework is based on different layers which are pictured in figure 4.2 and described in this section (4.2.1).This layer structure allows reusing code for different hardware and implementing software independant from the hardware. Hence only a hardware interface layer is needed for the specific actuator. The controller itself provides real-time safe code.
Currently, ROS control is used by different third-party frameworks like the navigation stack [66] or moveit [67].

4.2.1 Layers

The segmentation of the process from telling the robot how to move up to the signal the actuator gets, has a lot of advantages. On the one side, it is easy to switch control mode like position and effort control, without implementing a whole new driver.
ROS control is divided into five layers. The lowest layer is the robot or the actuators itself, the hardware interface layer communicates with the robot. The hardware resource interface layer provides interfaces for the controller itself. The controller manager takes care of the loaded and running controllers as well as switching between different controllers. The different controllers can be accessed with ROS interfaces like topics.

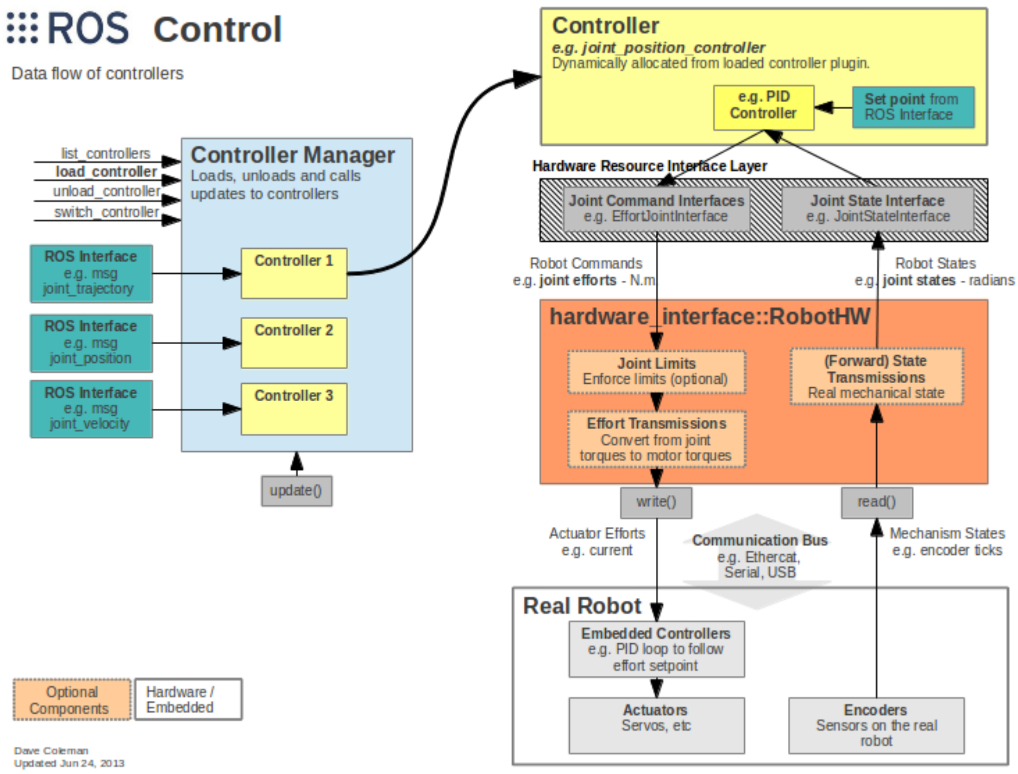


Figure 4.2: Flowchart of ROS control: The ROS control framework is divided into different layers to separate code from hardware and provide an easy way to switch between controller types and reuse the code for different hardware [64].

Real Robot

The *real robot* layer is not directly a layer of ROS control itself, in a way of implementation. It is the one which should be controlled. To communicate with the actuators each robot uses a bus system like Ethercat, Serial or USB to receive instructions. These instructions could be a current, an effort or something else depending on the actuator. In most cases, actuators used for robots support a state feedback like encoder ticks. This layer is hardware dependent.

Hardware Interface

The *hardware interface* dependents on the hardware and needs to be implemented for each actuator, joint or sensor of the robot. This layer abstracts from the robot itself and uses read and write functions to represent the communication to the robot. Some sensors like Imus or force torque sensors are read-only interfaces, whereas others like position or velocity joints are also writable.

For some actuators, a transmission is needed. For example to handle the mapping of the joint space used in the controller to the actuator space of the motor.

Hardware Resource Interface Layer

This layer provides information about the possibilities to communicate with the hardware interface, for example, with an effort command or a position command to control the joints depending on what type the hardware interface layer supports. Furthermore, the other direction and what kind of information the hardware provides is described.

Controller

The *controller* layer is independent of the robot hardware itself. The controller cares about the timed sending of commands to control, for example, the joint. There are a couple of standard controllers which are already implemented like a joint position controller or a joint effort controller. The command type, the controller gives to the hardware interface, depends on the type the interface excepts and could not be set individually within the controller.
Each controller provides a way to receive commands, usually with topics, but also other ROS interfaces could be supported. These commands, like a new position or a velocity, which should be applied to the actuator, are converted to the type accepted from the hardware interface. The Shadow Dexterous Hand accept effort values for example. These effort values are timed and calculated within a real-time loop. One controller can regulate more than one actuator at a time.

Controller Manager

As the name says, the *controller manager* manages the controllers. Each actuator can only be controlled by one controller at a time, yet it is possible to control them in different ways, like position or velocity controlled. This layer provides the loading, unloading, and switching between the different controllers and enforces resource conflicts. The controller manager also calls the important functions of the controller, like the *update* function that calculates the next command for the actuator and publishes the current state. This update function needs to be real-time safe, which is verified by the controller manager.

4.3 Control theory

The control theory considers dynamic systems whose behavior can be influenced by so-called input variables from the outside. Here, a distinction is made between open loop and closed loop control. Control theory is concerned only with closed loop control. The difference is that the closed loop includes feedback from sensors. An example of a closed loop is an oven, here the temperature is set but not controlled if and how it rises. By using a sensor, the rise in temperature can be controlled and improved, this is called a closed loop. The temperature is in this case the *process variable* which is controlled.
For a joint in the Shadow Hand the process variable could be either the position or the effort. To move the finger a *set point* has to be set, the controller controls the motion from the current

state of the process value to the set point. The process value is monitored and compared with the set point, this error indicates the applied action. In the case of the joint, the applied action can differ from acceleration, velocity or effort. If the set point and the process value are the same the controller tries to hold this state, in case of deviations, the controller adjusts [68].

4.4 Implementation

In section 4.2 ROS control was introduced, which is used by Shadow to control the joints of the Shadow Dexterous Hand. They implemented different controllers e.g. a position controller and an effort controller. At runtime, each controllable joint in the hand has it's own controller. These controllers listen to the */controller_name/command* topic. To move the specific joint, publishing a message to this topic will lead to a moving joint. For the position controller, for example, a float value must be published which defines the goal position. The controller receives this number and moves the joint to the commanded position if the value is valid.
This is a simplified description of the functionality of a controller. A more detailed description follows with the actual implemented controller used in this work. The tactile feedback could be integrated with different controllers, here the standard position controller provided by Shadow was used and extended. To show and test the functionality of a tactile based grasp, only joint 3 of the first finger and joint 5 of the thumb concern these adjustments. For the rest of the joints the original Shadow controllers are running. In a future work, the other joints could also use controllers with tactile feedback.

The pseudo code in algorithm 1 is the crucial adjustment in the position controller. Each ROS control controller needs the same methods implemented. The important function is the *update* method. This method is called from the controller manager in each control cycle. Since the Shadow Hand hardware will be controlled with effort values, each control cycle effort values are sent to the hardware. These effort values are calculated with the algorithm in 1. To generate the effort value which is sent to the actuators the algorithm gets the commanded position, the commanded effort and the raw sensor values from the BioTacs.
First of all the tactile information of the first finger and the thumb BioTac is preprocessed with the temperature compensation, the noise filter and the normalization. In line 6 the effort for the commanded position is calculated with a controller. If the tactile sensors don't recognize contacts this effort value will be sent to the motor. If the fingertips contact the object, two different situations are distinguished. Either both fingers have a contact, then the resulting effort is calculated with the commanded force in line 8. If only one sensor detects a contact, this finger stops it's motion and the effort is set to 0 in line 12. For the other finger the effort is still calculated by the position controller in line 6.
At each control cycle, the calculated effort is sent to the motor.

Besides the tactile controller, a simple action server was written. This action server could be used to open and close the finger like a simple parallel gripper. The server accepts a request on the */tams_tactile_grasp_action/goal* topic, with a *GripperCommand* message. This message is, among others, used by moveit and describes an interface to move the grippers in a position with

Algorithm 1: Algorithm to compute the commanded effort including tactile feedback

```
Compute effort (p, f)
Input  : float position p, float force f and tactiles t
Output: float effort e
joint = ["ffj3","thj5"]
for j in joint do
    preprocess(t[j])
end
e = controlPosition(p)
if contact(t[0]) and contact(t[1]) then
    e = controlForce(f)
else
    for j in joint do
        if active_joint == j and contact(t[j]) then
            e = 0
        end
    end
end
return e
```

a specific effort. Valid values for the position are between 0 and 1, with 0 as closed and 1 as open state. The user could just publish a 0 to the goal topic and both fingers will close, the rest is handled by the controller itself.

4.5 ROS Package

tams_sr_tactile_controller

The structure of the nodes and topics which are relevant for the controller are shown in 4.3. The tactile controllers are loaded with the *controller manager* which is processed in the */realtime_loop* node. Each joint has it's own controller, pictured is only one for joint 3 of the first finger. The */realtime_loop* accepts commands and force changes on the respective topics and publishes the current state of each joint. The action server publishes commands to the controller topics.

tams_sr_tactile_controller This node is mainly transfered from Shadows *srh_joint_position_controller*. For each joint an instance of this controller is created to control the respective joint. This node subscribes to the */controller_name/command* topic to receive target positions.

tams_tactile_grasp_action_controller An implementation of a simple action controller that receives a *GripperCommand* message as goal. The position value from the input goal

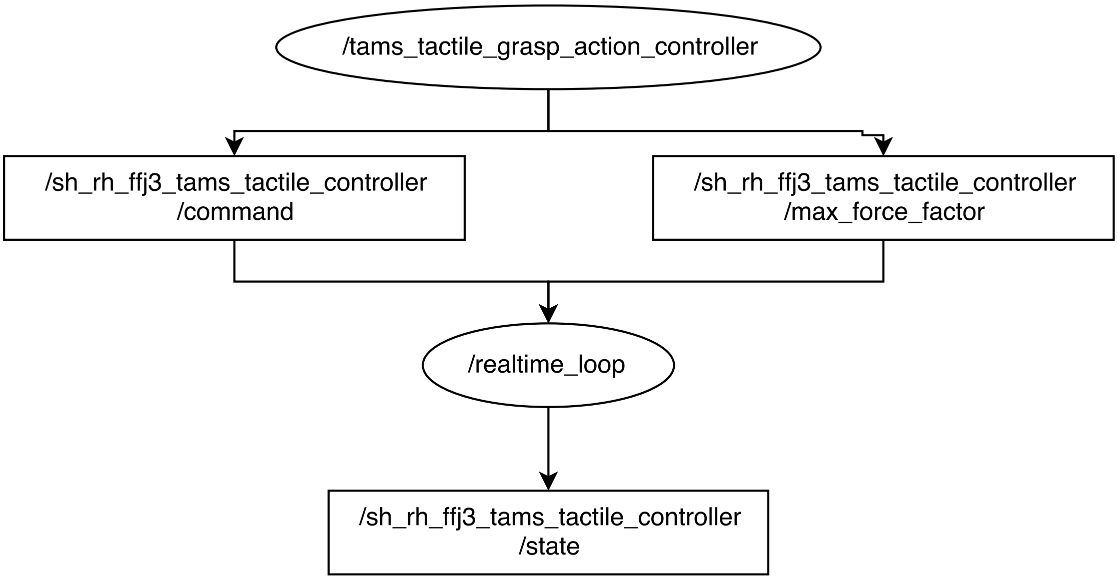


Figure 4.3: The node and topic structure of the tactile controller. Nodes are represented as ellipses and topics as squares, outgoing edges from nodes show publications, incoming represents subscriptions. The actual controller is loaded and processed in the */realtime_loop*, each joint has it's own *command* and *max_force_factor* topic to receive instructions. The action server can be used to send closing and opening movements to the gripper.

accepts values between 0 and 1, to open and close the two fingers like a simple parallel gripper.

change_controller With this script, the original position controller, which are running after the Shadow Hand start up, are stopped and the tactile controller are started.

4.6 Results

The results of a successfully applied grasp are shown in figure 4.1. The values of interest are the position states of the controlled joints in the upper diagram and the pdc values in the lower one. Currently, the pdc value is used as force feedback since the force estimation is not yet integrated within the controller.

The grasp can be distinguished in five phases: In the first one the hand is opened and nothing moves. In this state the pdc values and the joint positions, which are given in radian angles, are constant. The second phase begins after ca. 0.8 seconds, both fingers start to close, the pdc values are still unchanged. After 1.4 seconds the BioTac of the thumb detects a contact, the pdc value increases and the controlled joint 5 stops it's motion. The thumb applies no strong force in this phase but holds the contact to the object. In the next stage, the BioTac of the first finger detects a contact too, both fingers increase the effort now till the desired pdc value is reached,

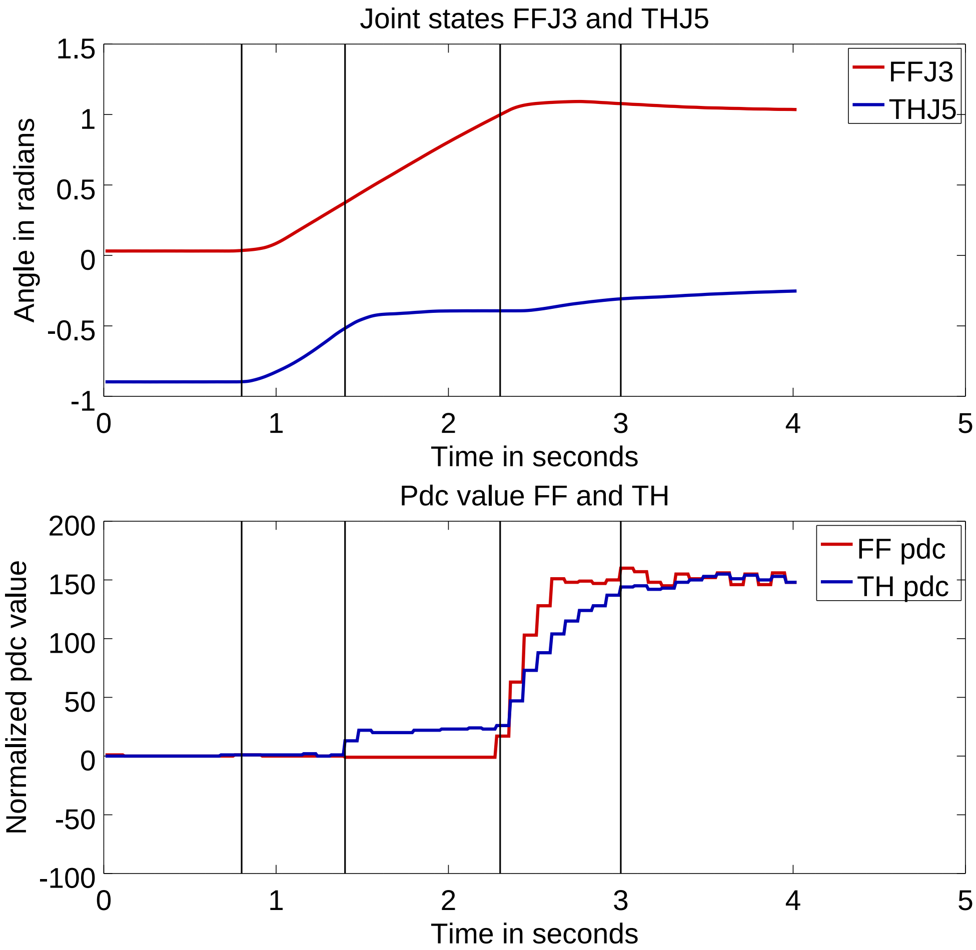


Figure 4.4: The upper diagram shows the joint positions of the controlled joints of the first finger and the thumb. The lower diagram shows the pdc value of the respective BioTacs. Both plots are divided into 5 phases of the grasp. In the first phase no joint is moving, the pdc values are constant. In the second phase both fingers close, the pdc values are still constant. After two seconds the BioTac of the thumb gets a contact and the movement stops. In the fourth phase, the first finger has a contact too, both fingers apply a force now till the desired pdc value of 150 is reached.

in this case, a pdc value of 150 is set as the target. In this phase, the joint position only changes roughly. In the last stage, the pressure is held, only small changes in the pdc value and the position is visible.
For future work the pdc value can be switched with the estimated normal force, the process of the curves will stay similar.

5 Conclusion

In this thesis, the tactile sensor BioTac was observed and described. The analyzes of the raw data result in different problems: On the one side the electrode data is temperature dependent and only stabilizes after at least 45 minutes of warming. This dependency results in data drifting. Another problem is the amount of liquid between the BioTac skin and the core which leads to different pressure in each sensor and consequently different idle states of the raw data.
The temperature dependency of the electrodes was successfully removed and the data drifting was minimized. The different idle states values could be normalized by shifting the values to 0 and continuous state control.
The information of interest for grasping was the normal force and the appropriate contact point on the tactile sensor for the reason that the normal force depends on the contact location. A common approach for the analytical calculation of the contact was presented and an improvement of this algorithm was introduced.
A calibration of the pressure value (pdc) considering the point of contact to an actual force in the physical unit Newton was applied and the accuracy was shown. Also, different calibrations for the point of contact were presented without a significant improvement in the accuracy.
The actual integration of the tactile feedback was shown with a two-fingered grasp considering the force measured with the BioTacs. Closing the fingers didn't lead to a shifting of the object, the force was as recently applied as both fingers detected a contact on the surface. The results were successful and the desired force could be hold on the object.

Future work

For prospective grasps, the tactile feedback could be included on all five fingers and more complex, more dexterous grasps could be applied. The number of affected joints could be adjusted and more natural or human-like movements could be explored.
Furthermore, the tactile information like the contact point can be used for in-hand manipulation or dexterous motions. Also for two-handed tasks, the tactile data could be used, not only with another robot but for cooperative tasks with humans.

Bibliography

[1] M. Beetz, U. Klank, I. Kresse, A. Maldonado, L. Mösenlechner, D. Pangercic, T. Rühr, and M. Tenorth, "Robotic roommates making pancakes," in *Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference on*. IEEE, 2011, pp. 529–536.

[2] F. Rothling, R. Haschke, J. J. Steil, and H. Ritter, "Platform portable anthropomorphic grasping with the bielefeld 20-dof shadow and 9-dof tum hand," in *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*. IEEE, 2007, pp. 2951–2956.

[3] A. Bicchi, "Hands for dexterous manipulation and robust grasping: A difficult road toward simplicity," *IEEE Transactions on robotics and automation*, vol. 16, no. 6, pp. 652–662, 2000.

[4] J. Butterfass, G. Hirzinger, S. Knoch, and H. Liu, "DLR's multisensory articulated hand. I. Hard-and software architecture," in *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, vol. 3. IEEE, 1998, pp. 2081–2086.

[5] K. B. Shimoga, "Finger force and touch feedback issues in dexterous telemanipulation," in *Intelligent Robotic Systems for Space Exploration, 1992. Proceedings. Fourth Annual Conference on*. IEEE, 1992, pp. 159–178.

[6] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and autonomous systems*, vol. 57, no. 5, pp. 469–483, 2009.

[7] N. Hendrich, D. Klimentjew, and J. Zhang, "Multi-sensor based segmentation of human manipulation tasks," in *Multisensor Fusion and Integration for Intelligent Systems (MFI), 2010 IEEE Conference on*. IEEE, 2010, pp. 223–229.

[8] A. Bernardino, M. Henriques, N. Hendrich, and J. Zhang, "Precision grasp synergies for dexterous robotic hands," in *Robotics and Biomimetics (ROBIO), 2013 IEEE International Conference on*. IEEE, 2013, pp. 62–67.

[9] A. T. Miller and P. K. Allen, "GraspIt! a versatile simulator for robotic grasping," *IEEE Robotics & Automation Magazine*, vol. 11, no. 4, pp. 110–122, 2004.

[10] A. Saxena, L. L. Wong, and A. Y. Ng, "Learning Grasp Strategies with Partial Shape Information." in *AAAI*, vol. 3, no. 2, 2008, pp. 1491–1494.

[11] B. Siciliano and O. Khatib, *Springer handbook of robotics*, 2nd ed. Springer, 2016.

[12] Å. B. Vallbo, R. S. Johansson *et al.*, “Properties of cutaneous mechanoreceptors in the human hand related to touch sensation,” *Hum Neurobiol*, vol. 3, no. 1, pp. 3–14, 1984.

[13] H.-K. Lee, S.-I. Chang, and E. Yoon, “A flexible polymer tactile sensor: Fabrication and modular expandability for large area deployment,” *Journal of microelectromechanical systems*, vol. 15, no. 6, pp. 1681–1686, 2006.

[14] R. Russell, “Compliant-skin tactile sensor,” in *Robotics and Automation. Proceedings. 1987 IEEE International Conference on*, vol. 4. IEEE, 1987, pp. 1645–1648.

[15] W. C. Nowlin, “Experimental results on Bayesian algorithms for interpreting compliant tactile sensing data,” in *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*. IEEE, 1991, pp. 378–383.

[16] R. A. Russell and S. Parkinson, “Sensing surface shape by touch,” in *Robotics and Automation, 1993. Proceedings., 1993 IEEE International Conference on*. IEEE, 1993, pp. 423–428.

[17] J. Fishel, G. Lin, and G. Loeb, “BioTac® Product Manual,” *SynTouch LLC, February*, 2015.

[18] “SynTouch,” SynTouch inc, visited on 24.10.2017. [Online]. Available: https://www.syntouchinc.com/

[19] “Tactile Properties Report Leather Sample Analysis,” SynTouch, Inc, visited on 13.11.2017. [Online]. Available: https://www.syntouchinc.com/services/

[20] N. Wettels, V. J. Santos, R. S. Johansson, and G. E. Loeb, “Biomimetic tactile sensor array,” *Advanced Robotics*, vol. 22, no. 8, pp. 829–849, 2008.

[21] B. Matulevich, G. Lin, and J. Fishel, “NumaTac® Product Manual,” *SynTouch LLC, July*, 2015. [Online]. Available: https://www.syntouchinc.com/wp-content/uploads/2017/01/NumaTac-Product-Manual.pdf

[22] B. Matulevich, G. Fishel, Jeremy Lin, and G. Loeb, “BioTac® SP Product Manual,” *SynTouch LLC, October*, 2015. [Online]. Available: https://www.syntouchinc.com/wp-content/uploads/2016/12/BioTac_SP_Product_Manual.pdf

[23] “BioTac Toccare Brochure,” SynTouch inc, visited on 24.10.2017. [Online]. Available: https://www.syntouchinc.com/resources/

[24] K. N. Bachus, A. L. DeMarco, K. T. Judd, D. S. Horwitz, and D. S. Brodke, “Measuring contact area, force, and pressure for bioengineering applications: using Fuji Film and TekScan systems,” *Medical engineering & physics*, vol. 28, no. 5, pp. 483–488, 2006.

[25] “Grip System,” tekscan, visited on 08.11.2017. [Online]. Available: https://www.tekscan.com/products-solutions/systems/grip-system

[26] C. Chorley, C. Melhuish, T. Pipe, and J. Rossiter, “Development of a tactile sensor based on biologically inspired edge encoding,” in *Advanced Robotics, 2009. ICAR 2009. International Conference on*. IEEE, 2009, pp. 1–6.

[27] N. F. Lepora and B. Ward-Cherrier, “Superresolution with an optical tactile sensor,” in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE, 2015, pp. 2686–2691.

[28] T. Okada, “Object-handling system for manual industry,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 2, pp. 79–89, 1979.

[29] *Shadow Dexterous Hand Technical Specification*, 1. 2013. [Online]. Available: http://www.shadowrobot.com/wp-content/uploads/shadow_dexterous_hand_technical_specification_E1_20130101.pdf

[30] “Shadow Robot,” Shadow Robot Company LTD., visited on 7.11.2017. [Online]. Available: https://www.shadowrobot.com/

[31] J. Butterfaß, M. Grebenstein, H. Liu, and G. Hirzinger, “DLR-Hand II: Next generation of a dextrous robot hand,” in *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, vol. 1. IEEE, 2001, pp. 109–114.

[32] Z. Xu, V. Kumar, and E. Todorov, “A low-cost and modular, 20-DOF anthropomorphic robotic hand: design, actuation and modeling,” in *Humanoid Robots (Humanoids), 2013 13th IEEE-RAS International Conference on*. IEEE, 2013, pp. 368–375.

[33] “DLR Hand II,” Deutschen Zentrums für Luft- und Raumfahrt, visited on 08.11.2017. [Online]. Available: http://www.dlr.de/rmc/rm/desktopdefault.aspx/tabid-11671/#gallery/28631

[34] V. Ciobanu, A. Petrescu, N. Hendrich, and J. Zhang, “Tactile sensor value preprocessing pipeline,” in *2013 17th International Conference on System Theory, Control and Computing (ICSTCC)*. IEEE, 10. 2013. [Online]. Available: https://doi.org/10.1109/icstcc.2013.6689038

[35] J. A. Fishel and G. E. Loeb, “Sensing tactile microvibrations with the BioTac - Comparison with human sensitivity,” in *2012 4th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*. IEEE, 6. 2012. [Online]. Available: https://doi.org/10.1109/biorob.2012.6290741

[36] J. A. F. Chia-Hsien Lin and G. E. Loeb, “Estimating Point of Contact, Force and Torque in a Biomimetic Tactile Sensor with Deformable Skin.” SynTouch LLC, 2013. [Online]. Available: https://www.syntouchinc.com/wp-content/uploads/2016/12/2013_Lin_Analytical-1.pdf

[37] N. Wettels and G. E. Loeb, “Haptic feature extraction from a biomimetic tactile sensor: Force, contact location and curvature,” in *2011 IEEE International*

Conference on Robotics and Biomimetics. IEEE, 12. 2011. [Online]. Available: https://doi.org/10.1109/robio.2011.6181676

[38] M. C. Jimenez and J. A. Fishel, "Evaluation of force, vibration and thermal tactile feedback in prosthetic limbs," in *Haptics Symposium (HAPTICS), 2014 IEEE*. IEEE, 2014, pp. 437–441.

[39] V. Ciobanu, D. Popescu, and A. Petrescu, "Point of Contact Location and Normal Force Estimation Using Biomimetical Tactile Sensors," in *2014 Eighth International Conference on Complex, Intelligent and Software Intensive Systems*. IEEE, 7. 2014. [Online]. Available: https://doi.org/10.1109/cisis.2014.52

[40] Z. Su, Y. Li, and G. E. Loeb, "Estimation of curvature feature using a biomimetic tactile sensor," *proc. American Society of Biomechanics*, 2011.

[41] D. Xu, G. E. Loeb, and J. A. Fishel, "Tactile identification of objects using Bayesian exploration," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 3056–3061.

[42] A. G. Eguiluz, I. Rano, S. Coleman, and T. McGinnity, "A multi-modal approach to continuous material identification through tactile sensing," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 10. 2016. [Online]. Available: https://doi.org/10.1109/iros.2016.7759721

[43] K. Sivakumar and C. Priyanka, "Grapsing objects using shadow dexterous hand with tactile feedback," *International Journal of Innovative Research in Science, Engineering and Technology*, vol. 4, no. 5, pp. 3108–3116, 2015.

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

[45] A. G. Eguíluz, I. Rañó, S. Coleman, and T. McGinnity, "Reliable object handover through tactile force sensing and effort control in the Shadow Robot hand," in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, 2017, pp. 372–377.

[46] C. Elbrechter, R. Haschke, and H. Ritter, "Bi-manual robotic paper manipulation based on real-time marker tracking and physical modelling," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. IEEE, 2011, pp. 1427–1432.

[47] J. Kramer and M. Scheutz, "Development environments for autonomous mobile robots: A survey," *Autonomous Robots*, vol. 22, no. 2, pp. 101–132, 2007.

[48] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2. Kobe, 2009, p. 5.

[49] KenConley, “ROS Introduction,” visited on 07.10.2017. [Online]. Available: http://wiki.ros.org/ROS/Introduction

[50] M. Quigley, E. Berger, A. Y. Ng *et al.*, “Stair: Hardware and software architecture,” in *AAAI 2007 Robotics Workshop, Vancouver, BC*, 2007, pp. 31–37.

[51] K. A. Wyrobek, E. H. Berger, H. M. Van der Loos, and J. K. Salisbury, “Towards a personal robotics development platform: Rationale and design of an intrinsically safe personal robot,” in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*. IEEE, 2008, pp. 2165–2170.

[52] “Willow Garage, Inc,” visited on 07.10.2017. [Online]. Available: http://www.willowgarage.com/

[53] M. Quigley, B. Gerkey, and W. D. Smart, *Programming Robots with ROS: a practical introduction to the Robot Operating System*. " O’Reilly Media, Inc.", 2015.

[54] G. Bradski and A. Kaehler, *Learning OpenCV: Computer vision with the OpenCV library*. " O’Reilly Media, Inc.", 2008.

[55] “Connect to a ROS Network,” visited on 08.10.2017. [Online]. Available: https://de.mathworks.com/help/robotics/examples/connect-to-a-ros-network.html?requestedDomain=www.mathworks.com

[56] WilliamWoodall, “catkin/conceptual_overview,” visited on 08.10.2017. [Online]. Available: http://wiki.ros.org/catkin/conceptual_overview

[57] C. H. Lin, T. W. Erickson, J. A. Fishel, N. Wettels, and G. E. Loeb, “Signal processing and fabrication of a biomimetic tactile sensor array with thermal, force and microvibration modalities,” in *Robotics and Biomimetics (ROBIO), 2009 IEEE International Conference on*. IEEE, 2009, pp. 129–134.

[58] E. Olson, “AprilTag: A robust and flexible visual fiducial system,” in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 5. 2011. [Online]. Available: https://doi.org/10.1109/icra.2011.5979561

[59] *Six-Axis Force/Torque Sensor System Installation and Operation Manual*, 3. 2016. [Online]. Available: http://www.ati-ia.com/app_content/documents/9620-05-Transducer%20Section.pdf

[60] “Logitech Webcam C930e,” Logitech, visited on 30.10.2017. [Online]. Available: https://www.logitech.com/assets/64665/c930edatasheet.ENG.pdf

[61] Shafranovich, “Common Format and MIME Type for Comma-Separated Values (CSV) Files,” visited on 13.11.2017. [Online]. Available: http://www.ietf.org/rfc/rfc4180.txt

[62] S. M. K. Chebotar, Hausman, “BiGS: Biotac Grasp Stability Dataset,” visited on 29.10.2017. [Online]. Available: http://bigs.robotics.usc.edu/

[63] H. F. Eric Lorup, “Distanz-basierte Interpolation,” visited on 29.10.2017. [Online]. Available: http://www.gitta.info/ContiSpatVar/de/html/Interpolatio_learningObject2.html

[64] W. Meeussen, “ros_control,” visited on 01.10.2017. [Online]. Available: http://wiki.ros.org/ros_control

[65] A. R. Tsouroukdissian, “ROS control, an overview,” 9. 2014, ROSCon 2014. [Online]. Available: https://roscon.ros.org/2014/wp-content/uploads/2014/07/ros_control_an_overview.pdf

[66] E. Marder-Eppstein, “navigation,” visited on 05.11.2017. [Online]. Available: http://wiki.ros.org/navigation

[67] I. A. Sucan and S. Chitta, “MoveIt!” visited on 01.10.2017. [Online]. Available: http://moveit.ros.org/

[68] G. F. Franklin, J. D. Powell, A. Emami-Naeini, and J. D. Powell, *Feedback control of dynamic systems*. Addison-Wesley Reading, MA, 1994, vol. 3.

Eidesstattliche Erklärung

Hiermit versichere ich an Eides statt, dass ich die vorliegende Arbeit im Masterstudiengang Informatik selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel – insbesondere keine im Quellenverzeichnis nicht benannten Internet-Quellen – benutzt habe. Alle Stellen, die wörtlich oder sinngemäß aus Veröffentlichungen entnommen wurden, sind als solche kenntlich gemacht. Ich versichere weiterhin, dass ich die Arbeit vorher nicht in einem anderen Prüfungsverfahren eingereicht habe und die eingereichte schriftliche Fassung der auf dem elektronischen Speichermedium entspricht.

Hamburg, den 16.11.2017

Vorname Nachname

Veröffentlichung

Ich stimme der Einstellung der Arbeit in die Bibliothek des Fachbereichs Informatik zu.

Hamburg, den 16.11.2017

Vorname Nachname