DTU Electrical Engineering
Department of Electrical Engineering

Design and Implementation of Tactile Sensing for Robotic Gripper Applications

Master Thesis

Design and Implementation of Tactile Sensing for Robotic Gripper Applications

Master Thesis
January, 2021

Written by:
Jacob Fiskaali Hertz

Advisors:
Silvia Tolu, Associate Professor, Ph.D., DTU
Christian Bjerrum-Niese, CEO and Founder, Graspian ApS

ETCS: 30

Published by: DTU, Department of Electrical Engineering, Elektrovej, Building 326, 2800 Kgs. Lyngby Denmark
www.elektro.dtu.dk

Approval

This thesis has been prepared over six months at the Department of Electrical Engineering, at the Technical University of Denmark, DTU, in partial fulfillment for the degree Master of Science in Electrical Engineering, MSc Eng.

The project was carried out in collaboration with Graspian ApS, who provided the sensors thus laying the foundation for the project.

Jacob Fiskaali Hertz - s153770

....

Signature

....

Date

Abstract

Providing robots with tactile perception would be an enormous advantage in tasks related to grasping and dexterous manipulation. It could enable robots to perform repetitive tasks that currently need to be executed by humans. Though the subject has been part of the robotics research field for decades, and new tactile sensors are continuously reported, tactile sensors are generally unseen in industrial applications.

By designing a tactile sensing system and implementing it into a robotic setup, this thesis attempts to answer how tactile sensing may be useful for industrial applications and what requirements are posed on the sensors and system. The final design achieves to extract relevant information from two tactile sensor mounted on a 2-fingered gripper, and uses it as control feedback in a grasping procedure. It uses a microcontroller to perform the sampling and pre-processing of the tactile data and via a ROS-based interface combines it with the measurements from force/torque sensors within the robot.

The system is evaluated by carrying out attempts to grasp a set of different objects with and without the use of tactile perception and comparing their performance. The test results show that the sensing system helps in adjusting to the exact location of the target objects and allows to grasp objects that are easily damaged including tomatoes and mandarins. The study concludes that basic tactile perception may facilitate the manipulation of fragile objects like fruit and vegetables and that is may be advantageous in manipulation of objects with partially known properties.

Contents

1 Introduction

As humans we rely heavily on our sense of touch when grasping and manipulating objects. It is easy to pick up an object with closed eyes even without knowing the shape of the objects beforehand. This is possible because the nerve cells within our skin help us to perceive properties like mass and shape while continuously providing feedback about he contact forces within the grasp.

Research within the field of robotics has often found inspiration in how humans perceive the world and it is therefore not remarkable to think that including tactile perception in autonomous robot systems may provide the same advantages and be useful in manipulation.

While visual recognition systems have increasingly improved and are generally widely available today, the integration of tactile sensing in robotics is still lacking behind, both in regard to hardware and to the system and methods for processing the incoming data. Robotic grippers with tactile sensors integrated in their fingers do exists, but there is a gab in the market for inexpensive and easily implementable solutions.

Extending robotic manipulators with tactile perception helps estimating relevant object properties, thus letting the robot adjust its grip with a sensitivity that is hard to obtain from a visual input alone. This is relevant when dealing with unstructured objects, like when picking items from a bin, or when manipulating fragile objects such as fruits and vegetables.

1.1 Literature Review

Tactile sensing as a research topic has been heavily studied for the last couple of decades and new tactile sensors are being reported in an increasing amount[1]. Applying robots with tactile perception has allowed them to estimate properties about the manipulated objects such as size, shape, weight, compliance or even texture calculated from the forces, pressure distribution and vibrations exerted on the sensor [2], [3]. Furthermore some sensors can provide information about an object slipping though the fingers of the robotic gripper or contribute to the feedback when performing dexterous manipulation [4]. Generally tactile sensing is beneficial in unstructured environments, where the properties, position or orientation of the objects are either entirely or partly unknown [5]–[7].

Countless types of tactile sensors relying on various transduction principles have been researched and developed. Their advantages and disadvantages as well as common issues have been described by many [1], [3], [5].

Despite the apparent advantages of giving the robot a 'sense of touch', the actual use of tactile sensors in robotics has been quite lacking. This is especially true when comparing with the use of vision, which due to cheap and high performance cameras along with significant research in the field of computer vision, has become the primary type of perception in robotics [2]. However, tactile sensing is able to reveal information about relevant object properties that cannot be estimated from vision alone, especially if dealing with unknown objects, and studies have shown how object recognition performs significantly better when incorporating tactile information alongside with visual input [8]. Furthermore, in manipulative tasks, the tactile input may compensate for visual inaccuracies due to the placement of the cameras [2].

A variety of papers have discussed the possible reasons for the apparent lack of tactile sensing in robotic applications, which is particularly true for industrial settings, and may be summarized in the following [1]–[3], [5].

Firstly, most of the research within the area has focused mainly on developing the sensors and not paid as much attention to the perception system surrounding it, thus creating a lack of system-level products that are easily implementable [1], [2]. Secondly, the sensor needs to be robust towards collisions and a working environment that may vary in temperature and humidity [1], [3]. A last thing to be kept in mind if aiming to close the gap between academic research and industry is the focus of providing a solution that is simple, reliable and cost-effective [5].

1.2 Problem statement

Though there has been increasing research in the area of tactile sensing for intelligent systems, their usage in robotic applications — especially in industrial settings — is extremely low. This seems to be due to the fact, that research in the field has focused on 'bench top' sensor development and not given a lot of attention to developing system-level products. Furthermore, in order to increase the applicability of the sensors in the market, more focus is needed on robustness and consistency regarding the tactile sensor and the sensing system.

In regarding this the problem, this study attempts to answer:

- Would the implementation of a tactile sensing system into a robotic setup, be able to improve tasks related to an industrial application, such as gently grasping fragile objects or estimating stable contact points when picking up items?
- What are some specific applications or tasks, where the implementation of such a tactile perception system would create an advantage?
- What requirements does the implementation of a tactile perception system impose on the sensor and system-level hardware in use?

1.3 Goal and methods

Based on a prototype sensor, the aim of the project is to design and implement a tactile sensing system onto a robotic gripper in order to endow the ability to grasp and manipulate various objects reducing the risk of dropping or damaging them. The work can be divided into four main categories: designing the hardware for measurements on the sensor, creating a communication interface between the local sensor hardware and the main control unit, processing the acquired data in order to extract relevant higher-level information and writing a controller for implementing the sensor in a gripping task. In the process of designing the complete sensing system the acquired knowledge about the prototype sensor is used to provide recommendations for further development of the sensor. Thus the goals of the thesis are to:

- Analyze important characteristics about the sensor in order to assess its possible applications and specify the design criteria
- Create an integrated sensor-interface that allows the robot to receive information about delicate touch on the grippers
- Create a perception system that is able to estimate the pressure applied to the gripper from the sensor input

- Design a control loop that lets the robot pick up fragile objects, tomatoes for example, without damaging them
- Evaluate the implementation of the sensor by comparing performance of a grasping task with and without the sensor system in place

The design process follows a bottom up approach that takes the sensor and analysis hereof as starting point for exploring the solution space. Current research within the field is applied to the specific case in order to create a solution matching the sensor in question. The four step described above are carried out in a sequential manner and evaluated in accordance with the design requirements and stated goals.

With the sensor at an early development stage, the progress is heavily test-based and structured around trying out different solution in order to develop the understanding of possible applications, limitations and necessary alterations of sensor and the system.

2 Related Work

2.1 Tactile Sensing and Perception in Robotics

Tactile sensing can be divided into two submodalities, namely *intrinsic* and *extrinsic* tactile sensing, with the corresponding physiological terms being *kinesthetic* and *cutaneous* sensing. However, the term "*tactile*" is often used to represent just the extrinsic part of the modality. Furthermore the term *tactile* should be distinguished from the word *touch* as the difference between sensing at multiple or a single contact point.

Cutaneous sensing corresponds to what humans sense on their skin on direct physical contact, whereas kinesthetic sensing is carried out by receptors in the muscles, tendons and joints in order to provide information about the static and dynamic posture of the body parts. Thus in the current field of robotics the analogous sensory mechanisms for kinesthetic sensing would be the force/torque sensors or joint angle sensor mounted inside the joints of the robotic arm. The extrinsic tactile sensor, on the other hand, are mounted on the surface of the robot.

Perception based on both cutaneous and kinesthetic sensing is called *haptic* perception, which corresponds to combining the information from joint angles and velocities with the extrinsic tactile sensory input [3]. This type of perception is needed for extracting information like the overall shape of an object or its surface texture, but is generally more computationally demanding requiring algorithms for fusing the data [5].

If the sensors used for the extrinsic tactile sensing are stretchable and covers several links, of for example a humanoid finger, they may provide information for the kinesthetic perception as well. This is also the case of human skin. Figure 2.1 shows how the sensing and perception type are related. Furthermore the perception may be classified as *active* or *passive* depending on whether controlled movement is part of the sensing. Feeling the texture or hardness of something would classify as active perception. Generally the tactile information may be used either for exploration of the environment and as basis for decision making or directly as a control feedback [3].

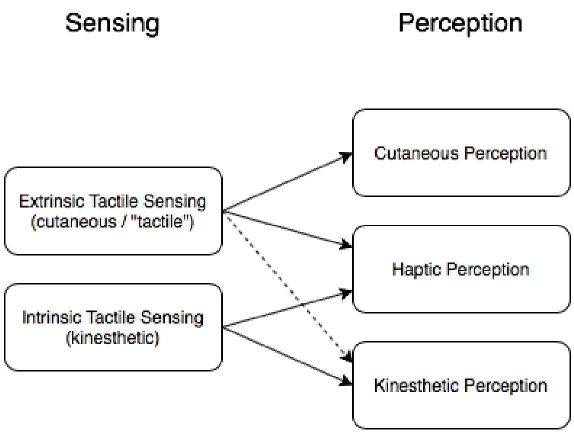


Figure 2.1: Classification of tactile sensing and perception as proposed by [3]

In prior implementation of tactile sensors into robotic systems a series of issues have found to be present. Typical issues with tactile sensors in robotics have been slow response times, lacking elasticity, poor durability or simply unacceptable decreases in dexterity due to improper sizes or numbers of wires. Other common issues are sensitivity to temperature or humidity and problems with lacking consistency between individual taxels [1], [3].

2.2 Principles of the Resistive Sensor

One way of translating an applied force or pressure into an electrical response, is through the principle of resistive sensing. These types of tactile sensors are classified by how a change in force will course a corresponding change in the resistance between the materials of the sensor.

When two conductive materials are pressed together, their contact resistance will depend on the force at which they are compressed. This is due to irregularities in the contacting surfaces, which are evened out when pressure is applied, thus increasing the contact area. Conductive polymers are typically used for these types of sensor.

The relation between interface resistance and contact area may be described by

$$R_S(F) = \frac{1}{A(F)} R_{SO} \tag{2.1}$$

Where $A(F)$ is a force-dependent area function with a value between 0 and 1, R_{SO} is the surface resistance of the sensor material when fully connected and $R_S(F)$ is the resulting interface resistance. Assuming a negligible electrical resistance in the materials themselves, the equation describes a linear relation between contact area and conductance across the connection.

The area function is highly influenced by the texture and internal properties of the sensor material and must be determined experimentally in order to fully model the relation between load and resistance. It is found to be non-linear in the measurements performed by [9], while experiments using other resistive based sensors like in [10] shows great results by adopting a linear mapping.

The simplicity in the construction of these types of sensors often make them both cheap and quite robust. In addition, the inverse relation between contact area and electrical resistance provides extra sensitivity at small loads, making them suitable for collision detection.

On the other hand since polymers have a tendency to creep, resistive based sensors are notoriously sensitive to drift, experienced by a gradual decrease in the resistance of the sensor taxel [9], [11].

Other downsides are the fact that they are prone to provide signals with long non-linear time constant and hysteresis [5] and that they often have a high power consumption due to the constant measurement current [12].

2.3 The Human Sense of Touch

Looking at humans has often been an inspiration for robotic development and can similarly provide insight into the field of tactile sensing and create a baseline for what is achievable. At the same time it should be kept in mind that the human perception system would not necessarily be the optimal approach for specific robotic applications.

The human skin contains 4 different types of receptors called *mechanoreceptors* responsible for detecting various tactile properties. Their most emerging characteristics are the *stimuli frequency* and *adaptation* and the the four types may be characterized as in table 2.1. Furthermore, the receptors are located at different depths in the skin meaning some types respond to light touch and others to deep pressure. The highest detectable frequency by the tactile receptors lies around 700Hz. For comparison vision has a maximum temporal resolution of ~ 50Hz and audio one of ~ 20000Hz

	Slow adapting	**Fast adapting**
Low frequency	*Merkel Cells* (0.4Hz - 3Hz)	*Meissner's Corpuscle* (3Hz - 40Hz)
High frequency	*Ruffini Corpuscle* (100Hz - 500^{+}Hz)	*Pacinian Corpuscle* (40Hz - 500+Hz)

Table 2.1: Characteristic of the four mechanoreceptors

A slow adapting receptor will continuously send activating signals while the matching stimuli frequency is detected, while a fast adapting receptor will fire when the stimuli is applied or removed.

The signals due to stimuli are transferred to the central nervous system via two different paths — the *Spinothalamic* and the *Dorsal-Column-Medial-Lemniscal* — one which is slower and one that is faster. Tactile information related to pressure and vibration where the temporal information is of importance takes the faster *Spinothalamic* path.

In some cases the signals endure local pre-processing on its way towards the central nervous system, like determination of the direction of force or curvature during manipulation, meaning the CNS doesn't have to deal with the raw data.

Before reaching the brain the information is much further filtered based on attention given to the specific body-part. This is part of the process of combining the information from countless receptors throughout the skin.

The spatial resolution of the mechanoreceptors in the skin varies throughout the body. In the fingertips it is found to be around 241 per cm^2 and in the palm around 58 per cm^2.

These facts can be used as guidelines for the development of tactile sensing systems in robotics, but it should be kept in mind that the human perception system would not necessarily be the optimal approach for specific robotic applications. The human sensing system is for example both hysteric, time varying, nonlinear and not least slow. All properties that are generally undesired in the current development of robotics [3], [5], [7].

2.4 Tactile Information Processing

As proposed by [13], information obtained using tactile sensing, may be categorized in a hierarchical manner based on the processing-stage of the information. Higher-level information build on top of lower-level and in the end on the raw data measured by the sensor.

Sensor-level information describes information which is directly detectable by the sensor. These may be *contact forces*, *vibrations*, *thermal inputs* and *pre-touch detection* and rely on the design and capabilities of the tactile sensor in use.

Force estimation is the most common type of tactile sensing and are the building blocks for the majority of the higher level information. Depending on the physical layout of the sensor contact force may be estimated in the normal direction only or as the full 3-dimensional force vector acting on the taxel as seen in [10]. The mapping from raw input data to force estimates may be affected by hysteresis, drift and damping due to the skin.

Vibration signals may be extracted from the dynamic response of the sensor, and useful for detection of slip and the sense of texture. Piezoelectric and ultrasonic based sensors are known for providing suitable dynamic responses [3], [12], but also acoustic sensors have been shown very effective to extract the vibrational modes in the sensor material [14].

Temperature measurements are less relevant in regards to manipulation tasks, but may be used for object distinguishing. In combination with a heating element the thermal conductivity can be estimated, for example helping with determining if the object is metallic or not.

On the edge of the definition of tactile sensing lie proximity sensors and other types of pre-touch sensors providing measurements of close distances to a contact surface. Information from these may be directly useful for control and planing in order to regulate speed and position before making contact or grasping an object.

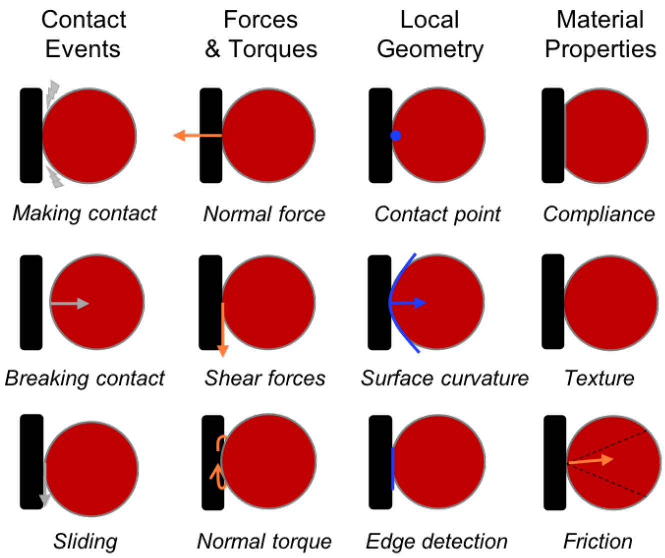


Figure 2.2: Overview of contact-level information. The illustration is inspired by a similar visualization by [13]

Contact-level information is related to the local contact area between the sensor and the object and is extracted from the information provided at the sensor-level. It is used either to further deduce knowledge about the touched object or directly for robot control. It may be ordered into four subcategories, as seen in figure 2.2, mainly contact event, local geometry, forces and torques and material properties.

Discreet state changes like making and breaking contact or the initiation of sliding or rolling motion are described as contact events. This type of information generally act as an error signal or a subgoal confirmation and are often required to be detected fast and reliably for the robot to react accordingly [13].

Slip may be determined either from the ratio between normal and tangential forces or by analyzing the high-frequency components of the force signals [10] which normally requires $> 1\,kHz$ feedback. Also making and breaking contact events may be determined using vibration information, this approach has the benefit of being able to detect contact events between the held object and the environment.

Force estimates are especially important as they provide the foundation for physical interactions with the object. The individual force measurements determined at the sensor-level can be combined to find the net-force acting on the object. Tangential or shear forces may be computed from the difference in normal forces but is highly dependent on the structure of the sensor ’skin’ [10]. Tangential forces components can then be combined to determine the normal torque.

Information about the local geometry consists of the contact position, the surface orientation and its curvature. Also the detection of edges and their orientation is of relevance. For 2D-sensors providing a tactile ’image’ of the contact region various image processing techniques may be used to extract features like edges, corners etc. or enhancing the resolution [15].

Determining material properties often requires active interaction with the object like sliding over the surface or making indentations. The key properties are compliance or stiffness, friction, surface texture, thermal conductivity, and adhesion. This type of information can be highly relevant for decision making in manipulation control.

According to [7] the most important information in grasping applications is the contact location as well as the net force between the robot finger and the object.

Object-level information tells something about the object as a whole. That could be its localization, shape, weight or center of mass. These are relevant for determining a dynamic model of the object in order to perform predictable motion and stable grasping when the environment is partially or completely unknown. The acquisition of this type of information most often rely on active exploratory movement of the robot like following the contour of the object, lifting the object or applying pressure to it to measure the force response.

The combination of the raw data into higher-level information can typically be done either analytically basing the computations on a physical model or by using learning-based approaches, where the mapping between raw and higher-level information is data-driven [13].

2.5 Control in Manipulation

Complex robotic tasks are often executed as a sequence of simpler action. Tactile information about the environment can thus be used to make decisions on what action to take next. It can also play a role in determining when an action should be finished or whether it had a successful or unsuccessful outcome.

During the individual subtasks feedback from the tactile sensors can be essential for performing the low-level control in order to maintain certain pressure on the object and for controlling the end-effector pose in accordance with the object shape.[13]

The type of low-level control needed for these types of interaction must be able to regulate the forces of the robot and its end-effector. A PI-based force control law can be described as in equation 2.2.

$$\tau = \tilde{g}(\theta) + J^T(\theta)\left(\mathcal{F}_d + K_p\mathcal{F}_e + K_i\int\mathcal{F}_e dt\right) \tag{2.2}$$

Where τ describes the torques applied at the robot joints, $J(\theta)$ is the Jacobian, $\mathcal{F}_d$ and $\mathcal{F}_e$ the desired end-effector wrench and error hereof respectively and K_p / K_i the controller constants. $\tilde{g}$ describes the torques present in the joints due to gravity [16, pp. 436–439].

When it comes to the grasping procedure different types of grasps exist. *Precision grasps* maintain a light grip around the object to be able to use the dexterity of the fingers to manipulate the object. A *power grasp* instead completely fixates the object in the hand in order to increase stability [6].

The grasping type comes down to the closure of the object. A *passive form closure*, like the power grasp, constrains the object based on the number and arrangement of the contact points. On the other hand, a simple two-fingered gripper with its movement constrained to one dimension will create a *passive force closure* when closing the fingers around the object, since the frictional forces play a role in keeping the object immobile. *Active force closure* describes situations where the object may be moved within the grip using the dexterity of the fingers [17].

3 Design and Implementation

The development of the sensing system is based on a specific sensor design that utilizes the principles of resistive-based tactile sensors. The characteristic of the sensor in question plays an important role in the choices and tuning of the hardware-related parts, but also the design of the high-level perception and control is in part affected by the size, shape and placement of the sensor.

Furthermore the solution is to some extent dependent on the robot arm in use, which is a 7 degrees-of-freedom Franka Emika robot. The robot is equipped with a standard 2-fingered solid gripper with a maximum width of $\sim 8cm$ depending on how the sensors are mounted. This naturally limits the size of possible objects that can be grasped.

Tactile sensing as a general concept is broad and extremely complex, so in order to create a tangible and testable goal, a specific grasping task is proposed to serve as a focus for the system design. This should not be understood as the definitive objective, but rather as providing a direction for the development of the system and a way to test the final system. The design process is based on the assumption that at least some prior knowledge exists about the environment and target objects.

The focus chosen for the design process is the task of grasping fruits or vegetables from a table without knowing the exact position of the objects and making sure not to damage them. The main objects used are tomatoes and mandarins, but in order to continuously assess the universality of the implementation a variety of objects are introduced in the testing phases in order to evaluate the limitations due to the focus bias.

The design process is split into four parts that each relies on the previous one and constitutes the repeating process of design and testing. The four elements of the system are:

1. Sampling Circuit
2. Perception and Data Processing
3. Communication Interface
4. Grasping Strategies and Control

The first part involves the establishment of the physical circuitry that allows for reading the state of the two sensors in use. Next, the perception part indicates the local pre-processing of the sampled data and the computations, which are expected to extract relevant information for the control task. The communication interface refers to the development of the firmware and software that defines the communication protocol and establishes the information transfer from sensor to controller. Lastly, this information should be used to design one or more real-time controllers that facilitate the grasping maneuver that is the focus of the process.

3.1 Requirements

In order to impose a set of design criteria on the individual steps, the goal scenario specified in terms of requirements. Thus, the robot should be able to autonomously

- localize a tomato or similar object from an uncertain position estimate
- adjust the pose of the robot-hand in order to get a stable grasp
- regulate finger movement such that the applied force on the object lies within a pre-specified range
- lift the object without dropping or damaging it

The procedure should preferably take less than 5-10 seconds.

The uncertainty of the object position is assumed to be on centimeter scale, an estimate that could be easily achieved if a vision system were to be introduced. The force required to hold a tomato is measured to be around $1.0N$-$1.5N$ at each of the two fingers, as described in appendix B.2.

Sensor requirements

- The sensitivity should enable accurate force estimation within the aforementioned grasping range.
- The sensor should respond quickly to *making* and *breaking* contact events.
- The layout and dimensions of the sensor should allow for a local estimate of the shape of the object.

Circuitry requirements

- The local processor should be able to sample the sensor at a rate that allows for fast detection of contact events.
- It must provide readings of the sensor that are accurate within a range, that does not conflict with the force limits and allows for determining the force distribution on the sensor.
- The circuitry may not introduce an amount of noise that prevents effective control.

Perception requirements

The system should be able to

- register *making* and *breaking* contact events in a reliable manner.
- compute the overall force applied to the sensor.
- gain information about the object position and orientation.

Interface requirements

- The code should provide a sensible interface for interacting with the sensor
- The software should be ROS-compatible and provide a ROS interface that allows for accessing the information and functionalities of the sensor system
- The communication should be fast to be able to react quickly to sensor events
- The communication should be reliable to avoid delays and incorrect readings which is essential in robotic applications.

Control requirements

- The gripper should be able to respond quickly enough to not squeeze the object when making contact
- All the control components should first of all prioritize precaution in order not to damage the handled object

In addition, since the sensor in itself is at a development stage, all the parts should be somewhat robust towards changes in the design of the sensor and its taxel layout.

3.2 Initial Sensor Analysis

The sensors used in the implementation work in similar ways to the description of resistive-based tactile sensors in section 2.2, where the change in contact area between a piece of conductive rubber and the underlying surface creates a difference in electrical resistance. In this case, however, the surface area is not modified due to imperfections on surface of the material itself, but rather due to an imprinted macro-structure on the underlying surface, that defines the resistive behavior when a pressure is applied.

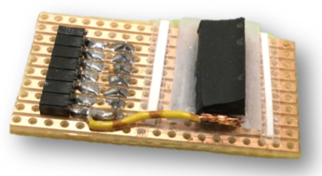


Figure 3.1: One of the used Graspian sensors

Preliminary tests on the sensor have indicated a linear relation between applied force and the measured conductance over the taxels as well as little indication of hysteresis. In regards to the principles of section 2.2 this suggests a linear area function $A(F)$.

This may be explained by the predominant influence of the micro-structural grading on the contact area compared with to microscopic surface imperfections determined by the texture and strain properties of the conductive rubber. This proportional relationship will be a prime assumption in the later design aspects.

The sensing area consists of 6 individual taxels forming a row that is $13\,mm$ wide, resulting in a resolution of $2.2\,mm$ along one dimension. A single piece of conductive rubber connected using a copper wire is responsible for the current flow through each taxel and is at rest supported such that it is hovering above the whole structure leaving a gap of air in between. In total 7 connectors act as access to each taxel and the conductive rubber. The rubber pad placed on top is $6\,mm \times 15\,mm$.

In accordance with the imposed requirements on the sensor, an initial analysis is carried out. First of all, by looking solely on the physical aspect of the sensor, some prospects may be concluded. Assuming the size of the target objects are on the scale of $\sim 5cm$ in diameter (see B.1), the $13\,mm$ width of the sensing area should be sufficient to provide an estimate of the surface contour. However, since the taxel layout is 1-dimensional, the measurements that can be instantaneously captured are confined to a single line on the object surface, and thus the gripper would have to be rotated to provide a 2-dimensional contact image. Imagining performing an entirely vertical grasp would imply that no knowledge is acquired about the vertical contour of the object meaning that the height and vertical placement would have to be known beforehand. This will in fact be assumed for the proposed task since the objects are placed on a table with a known height. The gap spanning from the end of the sensor board to the sensing area limits the height of the object to be at least $2\,cm$ in order to fully reach the rubber pad.

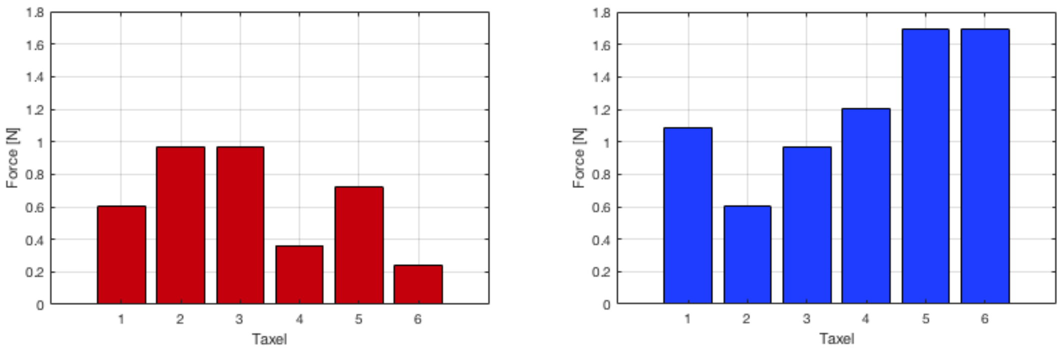
The fact that the conductive rubber is physically separated from the underlying taxel structure ensures that there is a clear distinction between *touched* and *untouched*, however it also creates a threshold for when detection is physically possible. This is, however, measured to be less than $0.1N$ which should be sufficiently low for performing the grasping procedure.

The sensor pad covering the taxels plays an essential role in how point-forces are distributed across the sensing elements and thus on the effective resolution of the sensor. A stiff rubber piece will distribute the pressure to affect more of the taxels making the contact point less distinguishable. Whereas a highly flexible overlay on a low resolution sensing area may result in the point-force ending up between the taxels. Thus the overlay must match with the spacing of the underlying taxels. In order to highlight the response induced by the object shape, the original sensor pad was replaced with a more flexible one, which also possesses a higher friction coefficient allowing for better grip without increasing the grasping force.

In order to analyze the dynamic range of the sensor a preliminary measurement is performed on each taxel of the two sensors. The aim is to determine an upper sensitivity limit ón the individual taxels. This is done by measuring the electrical resistance of the taxel when fully compressed and then decreasing the load on the sensor until the resistance starts increasing. The setup ensures that the applied weight is directed towards the specific taxel starting with a force of $2.5N$.

The measurements are carried out by tuning a variable resistor to match the resistance at max load. When connected in serial this causes the voltage to drop equally over the two. While reducing the load on the taxel, the point at which the voltage has changed 0.2% is noted as the maximum detectable force of the taxel. Since the sensor is expected to experience a sort of drift, this is compensated by repeatably lifting and re-placing the load and systematically performing the reading one second after placement.

Figure 3.2 shows the resulting maximum detectable force for each taxel on the two sensors.



(a) sensor 1 (b) Sensor 2

Figure 3.2: Saturation force

The corresponding taxel-resistance at maximum detectable load was measured using a multi-meter and showed in table 3.1.

Taxel	1	2	3	4	5	6
sensor 1	$0.18\,\mathrm{k\Omega}$	$0.18\,\mathrm{k\Omega}$	$0.19\,\mathrm{k\Omega}$	$0.20\,\mathrm{k\Omega}$	$0.21\,\mathrm{k\Omega}$	$0.19\,\mathrm{k\Omega}$
Sensor 2	$0.27\,\mathrm{k\Omega}$	$0.22\,\mathrm{k\Omega}$	$0.19\,\mathrm{k\Omega}$	$0.19\,\mathrm{k\Omega}$	$0.28\,\mathrm{k\Omega}$	$0.29\,\mathrm{k\Omega}$

Table 3.1: Measured minimum taxel resistances

An apparent issue experienced while performing the measurements is that at instances the resistance seemed to further drop when reducing load on the sensor. This would suggest that the maximum response from the sensor does not necessarily match with the highest applied force.

Secondly the taxel response seem to be quite sensitive to the exact positioning of the applied load. Moving it a millimeter in the orthogonal direction of the taxel row will in some cases change the resistance quite drastically.

Lastly, taxel 3 to 6 of *sensor 2* are measured to be somewhat correlated in their response. As an example the resistance of the central taxel 3 will drop when only the outmost taxel 6 is pressed, even when making sure the conductive rubber does not touch the grading in the center of the sensor. The exact reason for this is unknown, but some kind of shortcut must exist.

These are all thing the may play a significant role in the performance of the sensing system and and the superimposed control.

3.3 Hardware and Circuit Design

The first part of the system design consists of creating a physical interface that enables the measuring of the taxel resistances individually and in real-time. The interface should contain the appropriate circuitry, wiring and connections to the sensor and a microcontroller responsible for performing the sampling and communicating with the central unit.

The general focus points when considering the hardware and components is sampling time, accuracy, circuit complexity and signal condition. Also size and scalability is kept in mind. The solution draws inspiration from circuitry designs found in the literature [10], [18]–[20].

As it is typical for resistive sensors to be constructed as an array of taxels, the possibility of changing the taxel layout should be kept in mind when choosing the circuitry design. The aim is to create an interface with the necessary accuracy and timing requirements in order to asses the sensor and usefulness of the tactile system as a whole.

3.3.1 Microcontroller

An important aspect of minimizing the needed data transfer, is the possibility of processing the raw sensor data locally and preferably as close to the sensor as possible. To comply with this, the sampling circuit is constructed around an Arduino Uno board with its ATmega328P microchip, where sampling logic and data processing is carried out. The pin- and serial-interface on the board together with its six build-in ADCs allows for easy testing and prototyping relevant for this stage of the development process, while a later version may be constructed using the ATmega328P chip alone, disposing of any redundant components.

The ATmega328P chip contains 6 analog input pins and 14 digital I/O pins and is powered by $5V$ while also including an onboard voltage regulator providing a $3.3V$ source. Many other versions of Arduino-microcontrollers exist together with other microcontroller that are compatible with the Arduino interface, leaving the microcontroller quite easily replaceable.

Due to the dimensions and layout of both the board and the sensors, its not possible to remove the need for external wires connecting the two, however it is unlikely that the wire-crosstalk due to inductive coupling constitutes a main cause to decrease in accuracy.

The timing requirements are affected primarily by the conversion time of the analog-to-digital converter (ADC) and the Atmel® Atmega328 AVR 8-bit processor running at a

$16MHz$ clock frequency. The build-in ADC operates at a 10-bit accuracy within a clock frequency of $[50kHz; 200kHz]$ and requires 13 clock cycles to finish the conversion. With a default *prescaler* of 128, this results in a sampling rate of $\sim 9600Hz$.

$$f_{ADC} = \frac{f_{clock}}{prescaler} = \frac{16\,MHz}{128} = 125kHz \tag{3.1}$$

$$f_{sample} = \frac{f_{ADC}}{cycles} = \frac{125\,kHz}{13} = 9615Hz \tag{3.2}$$

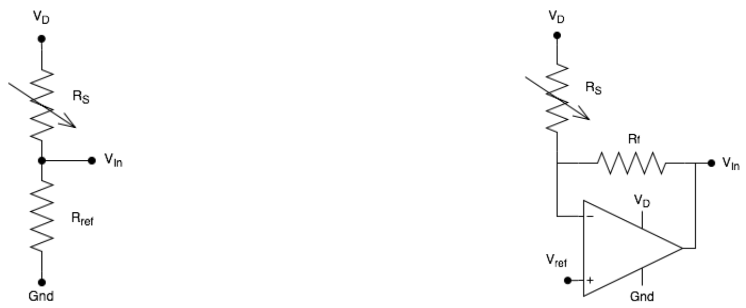
This should serve sufficient for sampling the 12 taxels of the two sensors, resulting in a sampling rate 12 times smaller of $801\,Hz$ corresponding to a sampling time of $1.2ms$. To achieve higher sampling rates it is possible to change the prescaler within the code reaching a sampling rate up to $\simeq 77kHz$, but this will have an effect on the resolution of the input.

The build-in serial interface is used for the communication and data transfer between the Arduino and the central processor.

3.3.2 Sampling Circuit

One of the main focus points in the design of the sampling circuit is to keep the circuit small and simple. This is mainly due the general desire of reducing unnecessary complexity that may introduce inaccuracies, especially since the aim is to create a baseline for testing the sensor and developing the understanding of how the different parts of the system are affected by each other.

Generally two underlying approaches are used for measuring resistance when timing plays a major role. Either by using a voltage divider with a known reference resistor, or by using a type of amplifier circuit.



(a) Voltage Divider (b) Amplifier

Figure 3.3: Two types of circuits for measuring the resistance of R_S

In the following the basic modeling of the two are described.

Voltage Divider

The voltage divider circuit seen in figure 3.3a is modeled by the formula

$$V_{in} = V_D\,\frac{R_{ref}}{R_S + R_{ref}} \tag{3.3}$$

Where V_{in} represents the measured input voltage, V_D the drive voltage, R_S corresponds to the variable resistance of a single taxel, and R_{ref} to the known reference resistor. This may be rewritten as

$$\left(\frac{V_D}{V_{in}} - 1\right)^{-1} = R_{ref}\,\frac{1}{R_S} \tag{3.4}$$

Using the same driving voltage for the sensor and the ADC, gives the relation

$$\left(\frac{ADC\_MAX}{input_{ADC}} - 1\right)^{-1} = R_{ref}\,\frac{1}{R_S} \tag{3.5}$$

Where the maximum value for the 10-bit ADC is $ADC\_MAX = 2^{10} = 1024$.

Since force applied to the taxel is expected to be proportional to the conductance $F \propto \frac{1}{R_S}$, the force can be estimated as

$$\hat{F} = \alpha\left(\frac{ADC\_MAX}{input_{ADC}} - 1\right)^{-1} + \beta \tag{3.6}$$

Here α and β are parameters that must be estimated through calibration. In theory α should be equal to R_{ref} and β equal to zero, but due to potentially unmodeled resistance in the circuit and wires α may vary, and in practice the conductive rubber of the sensor might not make contact until a certain force is applied or could be touching it without any pressure, creating incitement to add the β parameter.

To simplify the calculation and avoid division by zero, the formula is implemented as:

$$\hat{F} = \alpha\,\frac{input_{ADC}}{ADC\_MAX - input_{ADC}} + \beta \tag{3.7}$$

The sensitivity at a certain measured conductance may be described by the derivative of the input voltage

$$\frac{dV_{in}}{dC_S} = \frac{R_{ref}V_d}{(C_S R_{ref} + 1)^2} \tag{3.8}$$

Optimizing for a given C_S will results in:

$$R_{ref,opt} = 1/C_S = R_S \tag{3.9}$$

This shows that the voltage divider is most sensitive when the reference resistance match the measured resistance.

Amplifier Circuit

While different amplifier circuits exists, this one is based on the proposed circuit of [18]. Assuming ideal characteristics of the operational amplifier, the amplifier circuit can be modeled as

$$V_{in} = V_{ref} - (V_D - V_{ref})\frac{R_f}{R_S} \tag{3.10}$$

Here V_{ref} act as a virtual ground if comparing with an inverting amplifier having the non-inverting input terminal grounded. This will cause V_{in} to lie between $0V$ and V_{ref}, which is necessary as the microcontroller cannot read negative voltages. In this case the force estimation can be described simply by

$$\hat{F} = \alpha\,input_{ADC} + \beta \tag{3.11}$$

However to create a similarity to the calibration parameters of the voltage divider model, the relation is implemented as

$$\hat{F} = \alpha \cdot (ADC\_MAX - input_{ADC}) + \beta \tag{3.12}$$

The different relation between R_S and V_{in} of the two circuit types of a specific tuning is visualized in figure 3.4. The variable resistance R_S of the taxel is expressed as a conductance to keep a proportional relation to the corresponding applied force. Generally a steeper curve implicates higher sensitivity and is less affected by the resolution of the ADC, since a change in force will create a bigger change in measured input according to the model.

The striped lines in the plot indicates minimum resistances of the sensor taxels as measured in the initial sensor analysis (table 3.1). The circuit should be tuned such that these lie within the valid input range in order to exploit the full range of the sensor

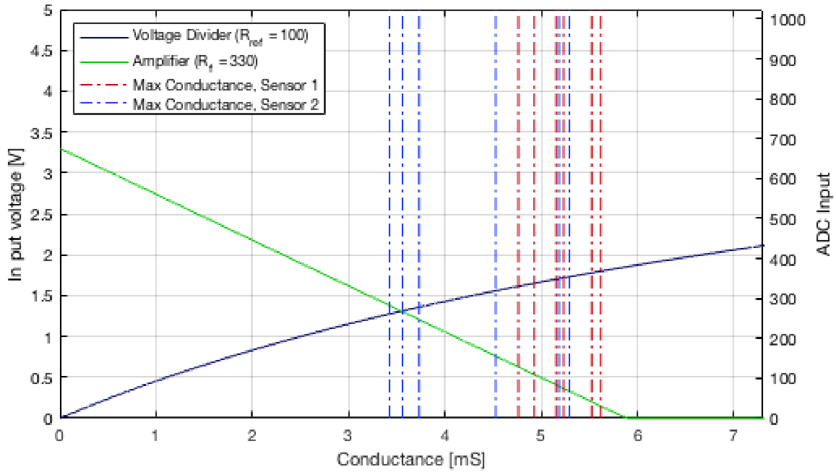


Figure 3.4: Theoretical model of measured input voltage and taxel conductance for the two circuits. The striped lines indicate max taxel conductance as measured in table 3.1

.

In order to measure each taxel individually a multiplexer is used as depicted in figure 3.5. Since each sensor consists of just 6 taxels, it would be possible to connect an analog input port for each of the taxel, using one microcontroller for each sensor. That however means, that the voltage divider or amplifier circuitry would need to be repeated as well, adding to the size and complexity of the circuit. The other downside is the fact that this would be much less adjustable, if the taxel-layout of the sensor were to change. Instead a single Arduino board is used for the sampling of both of the sensor, which also has the benefit of simplifying the serial communication when transferring the collected data.

Using a multiplexer does on the other hand have implications on the accuracy, as the *on*-resistance of the multiplexer affects the rest of the circuit. This inaccuracy should be more prominent for lower values of R_S. A static value chosen for the *on*-resistance may be introduced in the modeling in order to accommodate for this, though the true value varies based on the input voltage.

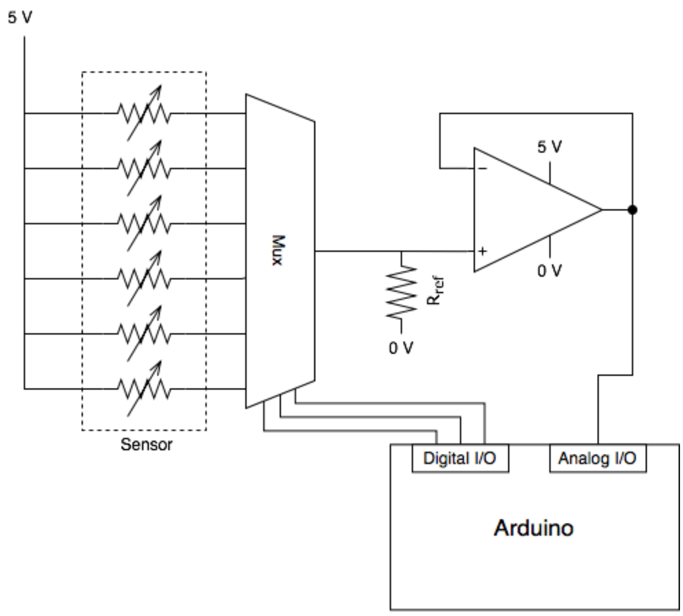


Figure 3.5: Sampling circuit using a voltage divider and a voltage follower. The drive and ground voltages are also provided by the Arduino board.

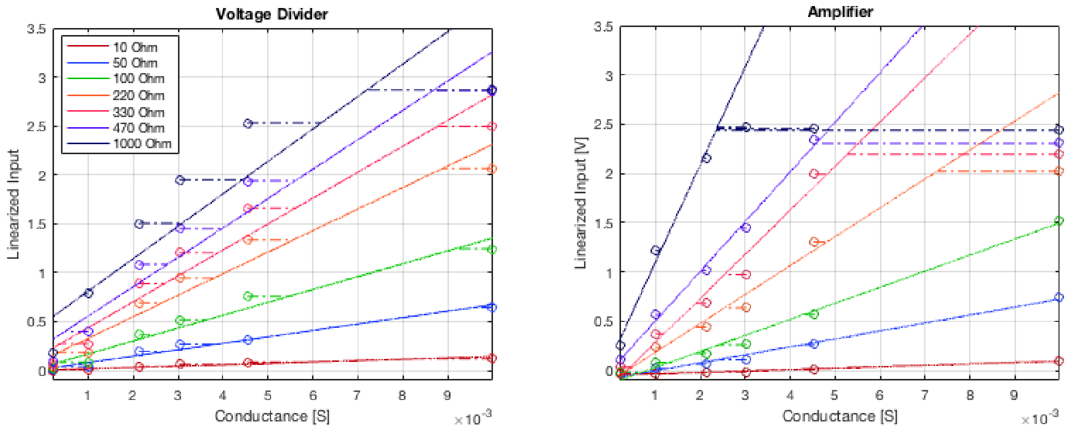
In the sampling circuit based on the voltage divider principle, an additional *voltage follower* (an operational amplifier circuit with unit gain) is introduced. This serves the purpose of minimizing the current flow between the voltage divider and the microcontroller in order to isolate the resistance of the reference resistor from the following wires and components. The diagram of the sampling circuit using the amplifier principle can be found in appendix A.1. In the implementation the reference voltage V_{ref} is provided by the $3.3V$ regulated supply of the microcontroller.

3.3.3 Tests of Circuit and Sensor

In order to evaluate the two proposed circuits a set of experiments are performed. This is done by using a testing board of seven known resistors representing the range of the possible taxel resistances in the sensor. The measurements are carried out using the circuits described by the diagrams of figure 3.5 and A.1, but with the sensor replaced with the known resistors. These have values of $10\,\Omega$, $50\,\Omega$, $100\,\Omega$, $220\,\Omega$, $330\,\Omega$, $470\,\Omega$ and $1000\,\Omega$.

The raw input is converted as seen below, such that the corresponding model may be fit to the data as a line in correspondence with equation 3.7 and 3.12. The resulting measurements can been seen in figure 3.6 together with the fitted model of each circuit. The amplifier model is fitted only to the measurements that lie within its the detectable region of the given tuning.

$$Y_{vd} = \frac{input_{ADC}}{ADC\_MAX - input_{ADC}}, \quad Y_{amp} = V_{ref} - V_D \frac{input_{ADC}}{ADC\_MAX}$$



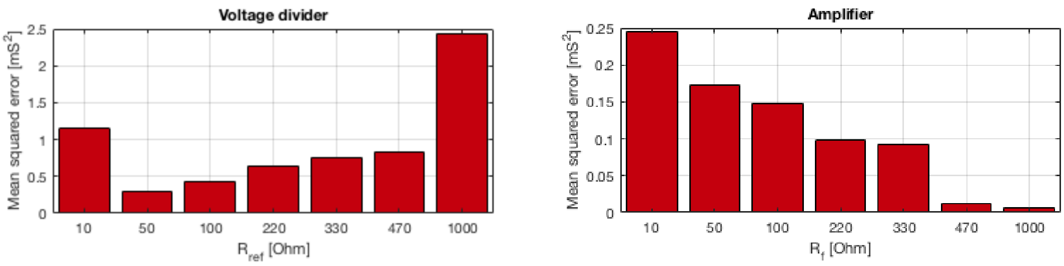
(a) Voltage divider circuit

Linearized input corresponds to equation 3.6 with $\alpha = 1$ and $\beta = 0$

(b) Amplifier circuit

Linearized input corresponds to equation 3.11 with $\alpha = -1$ and $\beta = V_{ref}$

Figure 3.6: Measuring known resistor values with the two circuits using different tuning.



(a) Voltage divider circuit

(b) Amplifier circuit

Figure 3.7: Mean squared error of measured versus known conductance

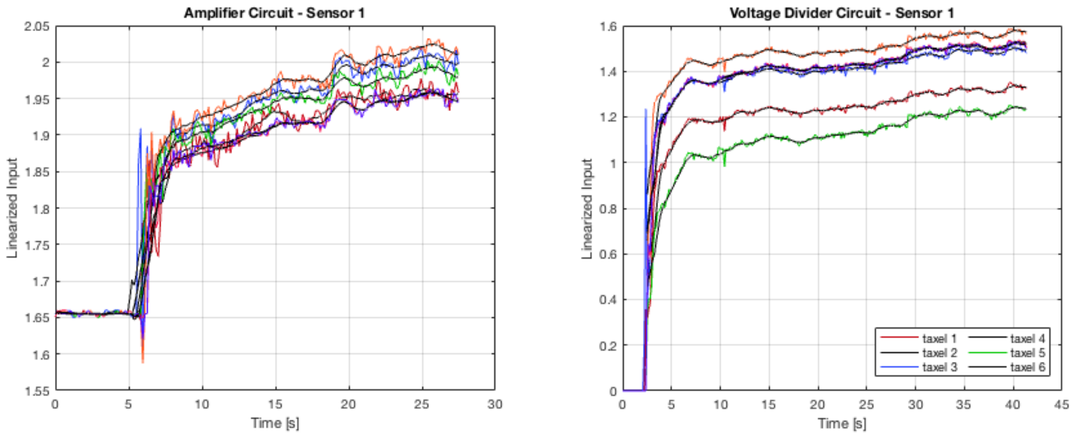
The fit is based on the difference between expected and measured *conductance*, as indicated by the dotted lines, since this is what defines the physical relationship with the corresponding applied force. This difference is also the base for the mean squared error computations seen in figure 3.7.

Looking at the error plot for the voltage divider it can be noted how an increase of the reference resistance R_{ref} decreases the model accuracy. The voltage divider is generally most accurate when the reference resistance and the measured resistance are equal. The reference resistance can therefore be used for tuning the sensitivity of the sensor where increasing the reference corresponds to tuning for lower force levels. This is why it does not fit as well for higher conductance.

Increasing the feedback resistance of the amplifier circuit limits its dynamic range. This can be seen from how the high-conductance measurements result in the same measured input. This means that the sensitivity is increased but the circuit loses the ability to measure larger forces applied to the sensor. Thus in order to benefit from the whole force range of the sensor, the feedback resistance should not exceed 220Ω if using the amplifier circuit.

Comparing the mean squared error of the two circuit types it is clear that the amplifier circuit provides much better predictability. A possible reason would be that the resistance of the multiplexer induces a greater disturbance in the voltage divider circuit. Though this provides a useful guidance for the choice and tuning of the circuit, the interplay with the actual sensor is important for selecting the final design.

The two sensors are tested with various tuning of each design by placing weight corresponding to $1N$ on the sensor and logging the response over time. All the measurement can be found in appendix B.5, but for comparison figure 3.8 shows the result when using the amplifier circuit with a feedback of $R_f = 220\Omega$ and the voltage divider circuit with $R_{ref} = 100\Omega$. The measurements are made with *sensor 1* and shows a clear difference in noise level using the two circuits.



(a) Amplifier circuit with $R_f = 330\Omega$ (b) Voltage divider circuit with $R_{ref} = 100\Omega$.

Figure 3.8: Time response from sensor 1 when applying $1N$ using each of the two circuits.

This noise problem seems to be a concern mainly with *sensor 1* and is occurring independent of the tuning of the circuit. The measurements are carried out using the exact same circuit for sensor 1 and 2. The cause of it is not exactly known, but would not provide an acceptable feedback signal for a control loop.

In both cases, but mostly with the amplifier circuit, some oscillatory behavior is present. The placement of the weight is done by hand meaning that disturbance during the first couple of seconds may be affected by this. However, it seems improbable that the later oscillations is caused by this. A more precise measurement setup would be needed to completely analyze this and more tests should be done on the sensor itself, in order to determine its cause. It is also clear from the result that the sensor is subject to short term drift.

Based on these results the choice for the circuit design is the voltage divider circuit. The reference resistor is chosen to be 100Ω as a trade-off between accuracy and input resolution. It should be clear, however, that the choice is mainly due to the poor results of the amplifier circuit in combination with *sensor 1*. Circuits used for sampling resistive sensors with taxels arranged in an array often implement circuits similar to the amplifier circuit in order to suppress an arising phenomenon known as crosstalk (further described in A.1.1). Thus with the preceding development of the sensor in mind the amplifier circuit would actually have been more fitting. This is further supported be the circuit comparison found in appendix B.9 based on a similar prototype sensor with an array arrangement.

The measurements in appendix B.8 shows the maximum measurable force of the two sensors when used in a gripping context.

Sensor Calibration

The circuit and sensors are further tested by performing a calibration of the individual taxels in correspondence with equation 3.7. This is done by placing increasing weight on the sensor pad and measuring the response of the individual taxels in comparison with the applied force. The fact that the force is applied to all the taxels at once serves the purpose of taking irregularities of the sensor pad into account. This ensures that the relative and total measured force is estimated correctly. The result is seen in figure 3.9 and 3.10 where the horizontal axes in the plots show the total applied force to the sensor.

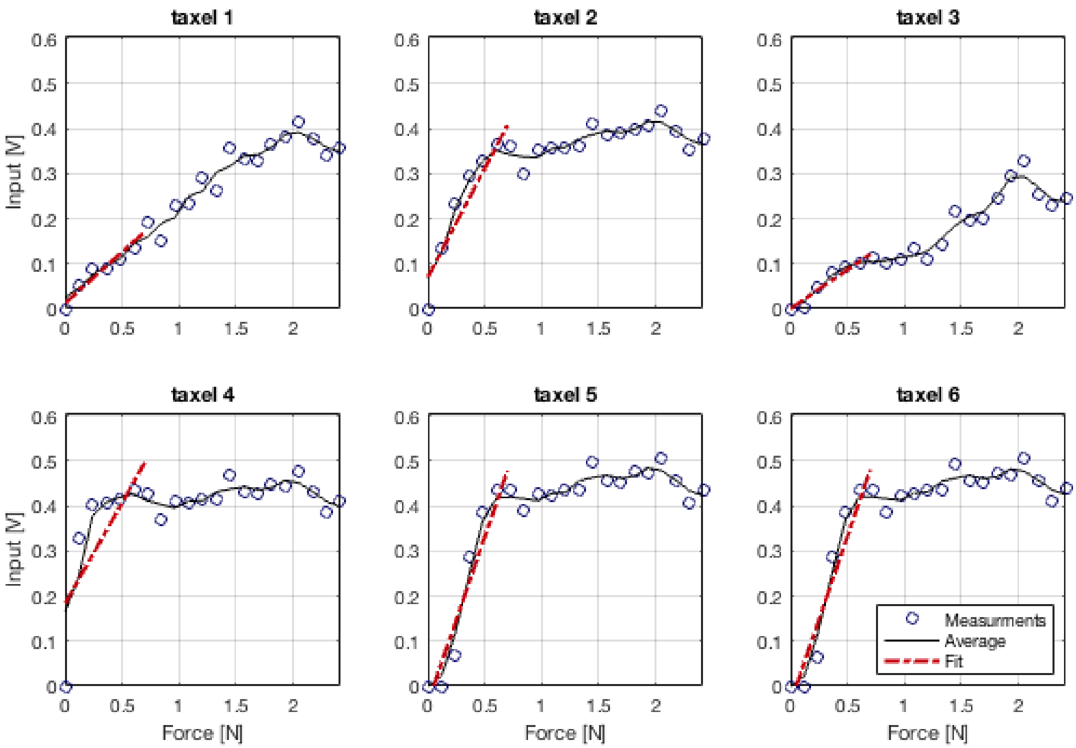


Figure 3.9: Calibration of sensor 1

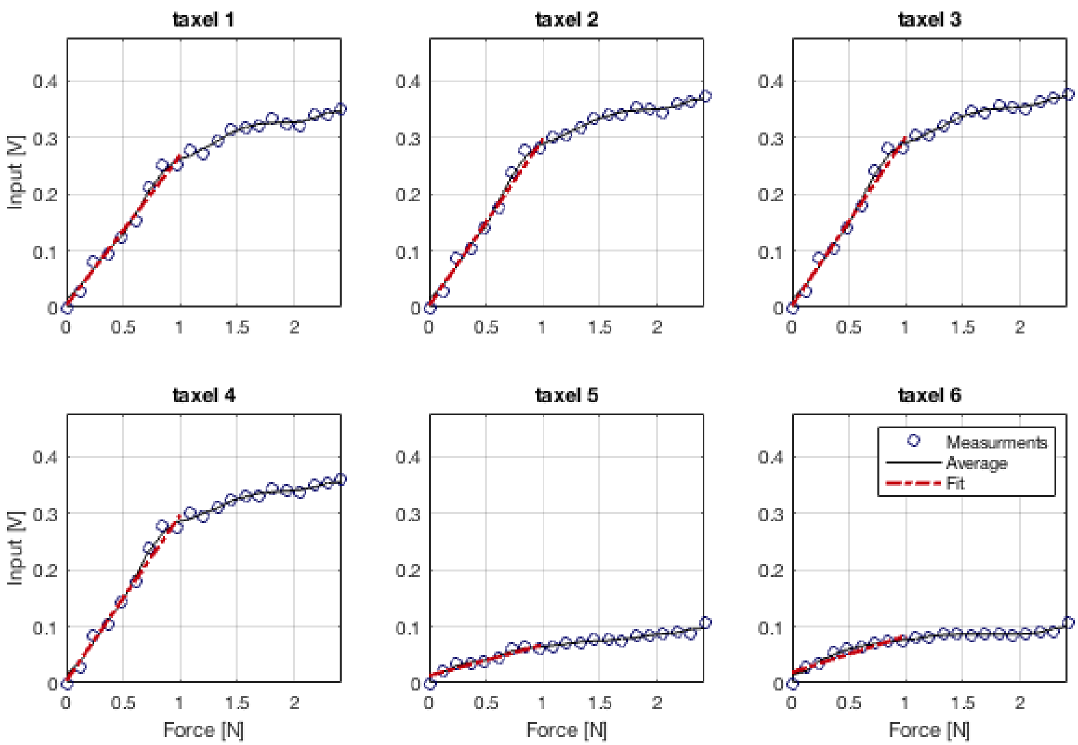


Figure 3.10: Calibration of sensor 2

The calibration shows quite different results for the two sensors. Most of the taxels of *sensor 1* saturate after just around $0.5N$ whereas the first four taxels of *sensor 2* are able to provide distinguishable inputs past $2N$.

For the case of *sensor 1*, the first and thirds taxels behaves differently than the rest. It is expected that the shape and behavior of both the underlying conductive rubber as well as to rubber pad overlay affect the distribution of forces. This may cause the smaller proportion to be applied to taxel 1 on the edge, which will explain the apparent low sensitivity but high force range. The results show that taxel 4 has a significantly low force range, which matches with the initial measurements in 3.2 (note the difference in force applied to the whole sensor and to the individual taxels).

Looking at *sensor 2* taxel 5 and 6 clearly have a much smaller sensitivity. These are also the taxels with experienced correlation issues, which may hint a type of construction error. There is a clear change in resistive behavior of the contacting materials when exceeding $1N$ of applied pressure. A hypothesis for this could be that the peaks of the underlay starts to slide into persistent deformations in the conductive rubber, but the actual cause is not known.

A thing to note from calibration plots of *sensor 1* is that the input starts to decrease again after around the $2N$ mark. This is the same problem as indicated during the measurements of minimum resistance. Taking taxel 5 as example the input drops to a value of around $0.4V$ which coincides with a force of approximately $0.5N$. This effectively means that force estimation above this value is unreliable for that taxel.

3.4 Perception and Information Processing

In addition to the adjustment of the taxels measurements, further processing is needed to provide relevant information for use in the proposed grasping application. The selection of what information is most relevant is based on the specific grasping application in focus as well as the possibilities of the sensor.

It should be clear that *making* contact events together with an estimation of the normal force is at least necessary in order to detect the target object and hold it without squeezing. Additionally some information about the object geometry needs to be inferred such that the grasping pose can be autonomously adjusted.

3.4.1 Contact events

The binary state of whether the sensor is touched or not may be determined by defining a lower threshold for measured contact force. As mentioned earlier, the physical aspects of the sensor makes the distinction between touched and untouched quite clear, and thus the threshold value should match that of the sensor.

To accommodate for differences in the sensor, individual thresholds are defined for each taxel. These may be determined automatically when the sensors are in an untouched state by measuring the variance of the input signal. The same threshold is also applied before any other computation of tactile information in order to suppress the noise of inactive taxels.

Detection of slip is not considered due to its computational complexity and the condition of the sampled signal.

3.4.2 Force and Pressure

As described in section 3.3.2, the normal force at each individual taxel is computed directly from the calibrated measurements and should contribute equally to the total net force normal to the sensor.

$$F_{N,contact} = \sum_{i=1}^{N} f_i \tag{3.13}$$

In some cases the *pressure* serve as a more relevant measure for the external impact on the sensor, which relies on an estimate of the contact area.

$$P_{contact} = \frac{F_{N,contact}}{A_{contact}} \tag{3.14}$$

The simplest way of estimating this would be to count the number of activated taxels and scale it with the sensor resolution. This, however, results in a discontinuous signal, making it unpractical for control purposes. To counteract this, the activation-level of each taxel is taken into account.

A computationally simple way to do this, which is valid no matter the taxel layout, is to treat all taxel above a certain threshold as fully touched, and weight all active taxels below based on their activation as described below.

$$w_A(i) = \begin{cases} 0, & f_i < T_i \\ \frac{f_i}{\eta \cdot f_{max}}, & f_i < \eta \cdot f_{max} \\ 1, & \text{otherwise} \end{cases}$$

The threshold is defined as a fraction of the maximum measured contact force with $\eta = 0.6$. The contact area is then computed as a proportion of the sensing area

$$\hat{A}_{contact} = \frac{\sum_{i=1}^{N} w_A(i)}{N} \tag{3.15}$$

Force applied tangential to the sensor pad will due to the deformation of the material increase the measured force on one side of the contact point while decreasing it on the opposite side. The exact distribution of these resulting forces dependent on the contact geometry and sensor materials, and an analytical approach of computing tangential force component would require a precise model of how the sensor material deforms based on the contact geometry.

This is highly complex in itself and thus using a learning based approach would be more fitting. This would however require a setup capable of applying precise forces with known direction and magnitude to the taxels of the sensor, which has not been available.

3.4.3 Local Geometry

The perception of local geometry has been focused on estimating the location of contact. Due to the relatively small size and resolution of the sensing area compared to the size of the target objects, all of the activated taxels are used for computing the central contact point. This is done using a "center-of-mass" approach as of equation 3.16.

$$x_c = \frac{\sum_{i=1}^{N} x_i \cdot f_i}{F_{N,contact}} \tag{3.16}$$

The solution is computationally simple and easily extended to cover two dimension if having a grid of taxels. Other ways of estimating the contact point would be to base it on the taxel with the maximum value and interpolating over the immediate neighbors or by fitting a parabola and using its vertex to indicate the contact location.

3.4.4 Compliance

The compliance of an object specifies relation between force and deformation of the contact surface. In contrast to the others, this perception must be carried out in an *active* manner in order to force on indentation in the object. This also implies that it cannot be computed locally altogether.

The inverse term is *hardness* and is defined as the experienced force due to a certain displacement after contact is made.

$$H = \frac{\Delta F}{\Delta X} \tag{3.17}$$

This measure may be found by closing the gripper around the object until a set force threshold or change in width is reached. The compliance information is not directly used in the implemented grasping application, but the computation is performed on different objects in B.8.

3.5 Communication Interface

By having an industrial application in mind, an important focus point for the sensing system as whole, is that it should be easy implementable, act as a package unit, and be universally applicable. Besides that it must be fast and reliable and not least be well structured and provide a sensible interface.

To make the sensing system work with most robot arms, the internal and external communication is build upon *ROS* (Robotic Operating System) and *ROS Control*[1]. ROS can be classified as a *meta-operating system* and provides various tools and frameworks including handling communication between multiple asynchronous processes. These independent processes are labeled *nodes* in ROS terminology and implements a topic based publish-subscribe (pub/sub) protocol for passing data within the network of nodes.

[1] http://wiki.ros.org

The language used for writing the software is C++, but given the language independence of ROS means that also Python (and a few other languages) may be used for interacting with the system.

Within the ROS framework information in transferred using *Messages*, *Services* and *Actions*. While *messages* are simply data containers, *services* assume a request / response behavior. The same goes for *actions* with the addition of supplying continuous feedback of the ongoing execution.

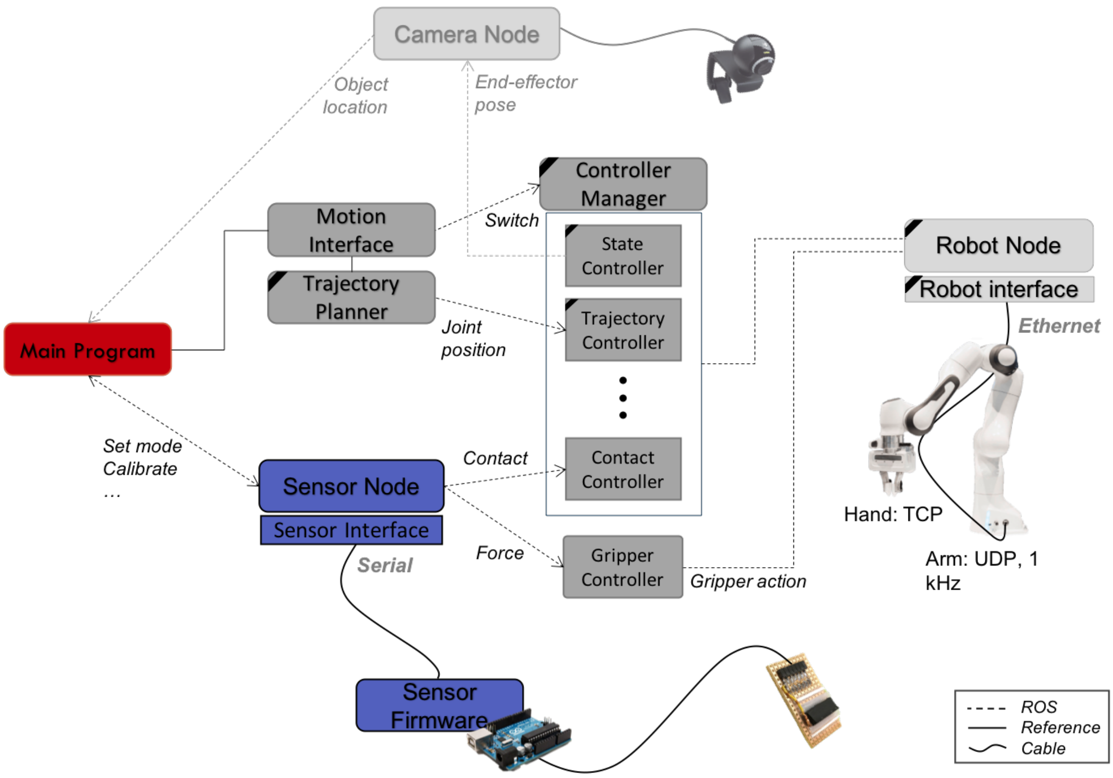


Figure 3.11: Overview of information dependencies between main components in the ROS-based structure. The pub/sub pattern should allow for easily adding a vision system later on.

Figure 3.11 shows an overview of the most relevant communication paths between different parts of the system. Components with a black label in the corner are parts of external libraries like *ROS Control*, *franka_ros*[2] and *MoveIt!*[3]. The *Main Program* which is responsible for performing the grasping procedure interacts with the sensor using service calls and with the robot through the Motion Interface. As the requirements of the tactile sensing system are based on the idea of having an initial estimate of the target object the illustration also depicts how a vision system would be added into the communication pattern for providing this information.

3.5.1 Robot Interface

Franka Emika provides a clean C++ API for controlling the robot as well as a ROS interface which includes support for *MoveIt!* and *ROS Control*. The robot operates based on a set of controllers defined according to the ROS Control framework and guarantees an execution rate of 1 kHz for commanding its 7 joints.

[2] https://frankaemika.github.io/docs/franka_ros.html
[3] https://moveit.ros.org

The gripper, on the other hand, does not support real-time control and uses a completely separate interface that relies on the much slower TCP-protocol compared with the UDP-protocol that is utilized for controlling the arm. The response time of commands sent to the gripper is not subject to a hard-time limit and is measured to be on the scale of hundreds of milliseconds. This poses a major problem in regards to the grasping task as fast response is absolutely critical when dealing with fragile objects.

The issue is embodied within the firmware of robot and it would require a new release by Franka Emika in order to make real-time control possible on the fingers of the robot[4]. This is a problem that may be found in other commercial grippers as well.

3.5.2 Structure and Packaging

The system is structured into a set of packages with each their purpose as seen below.

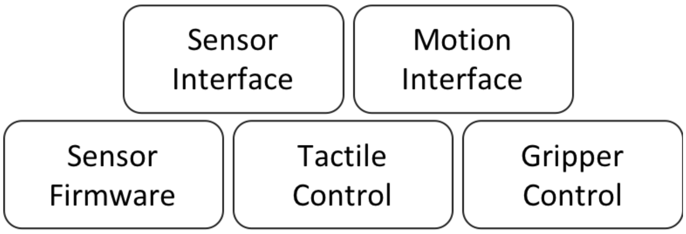


Figure 3.12: System Packages

Performing a successful grasp relies on a combination of both high-level and low-level control. The planning required to move the gripper to a desired pose is computed by the Trajectory Planner and passed on to the Trajectory Controller included in the *franka_ros* package[5]. The purpose of the *Motion Interface* is to combine the relevant functionalities of ROS Control and MoveIt! to provide a high-level interface for moving the gripper to the estimated target location before activating the low-level tactile control.

These controllers are contained in the *Tactile Control* package, and directly uses tactile information published by the sensor node as control feedback. Each controller compromises an action server which allow the Motion Interface to specify the desired control output in terms of action calls. Their implementations are described later in section 3.6.

Because of the difference in the way that the gripper-interface is constructed its controller cannot be part of the ROS Control framework and needs to run as its own separate node. The fact that the control of the arm and the fingers runs completely parallel is however beneficial when performing high level movements, as the Gripper Controller may continuously adjust gripping force and send a notification if the object is dropped.

The part of the system regarding the tactile sensors are divided into code running on the microcontroller labeled as *Sensor Firmware* and that on the central unit providing the interface to the rest of the system.

The goal of the sensor firmware is three-fold and consists of performing first of all the sampling of the sensor, then the pre-processing of the raw data into relevant information as described in section 3.4 and meanwhile communicating with the central unit.

The code is structured so that it is easy to update the specifications of the sensor or changing the circuit design which is of especially great importance as the sensor and

[4]According to Franka Emika this feature is under development at the moment (Jan. 2021)

[5]Defined in the *panda_moveit_config* package which is not included in the franka_ros after version 0.6.0

circuit are under development.

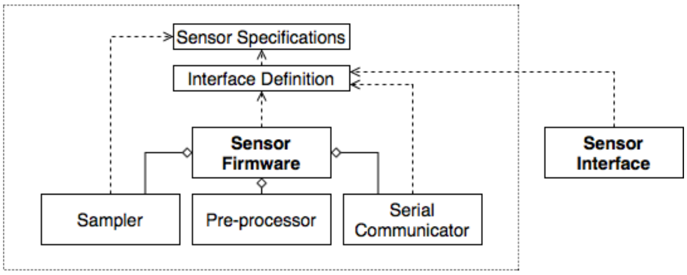


Figure 3.13: UML style class diagram of the sensor firmware

As an example the hardware related part of the *Sampler* consisting of setting the mux and reading the ADC is decoupled from the information computations etc.

3.5.3 Sensor Interface

In an equal manner to the robot API, the interface for the sensor is constructed as a C++ interface with a ROS integration on top in order to keep a one-way dependent structure. An Arduino-compatibility version of ROS does actually exist, which allows for installing the ROS framework directly onto the microcontroller, however due to the very limited memory on the board this means that the message buffers and protocol overhead quickly take up all the space when defining multiple message-topics.

Instead the data is transferred as its little-endian byte representation disregarding any unnecessary overhead. The transmission is instead verified by the known size of the information from both ends of the channel.

Since some of the tactile information is much more time critical than other, in particular information about initiation and breaking of contact, it is important that this type of data is not delayed due to simultaneous transmission of other information. Two solutions exists for this. Either multiple physical communication channels may be set up to carry the information at different rates, or the information may be filtered or predetermined based on the current task. This is somewhat analogous to the attention based perception as presented in section 2.3.

For the implementation the connection is realized using a single cable creating the need for filtering the information. This is solved by defining a set of *modes* for the microcontroller to instate, calling for different pre-processing of the tactile data and transmission hereof.

The different modes provide packaged information that is relevant for performing different tasks. As an example the *Geometry* mode provides the normal force, the estimated contact area and the contact center point.

This setup means that the sensor system can be commanded to provide information solely on breaking and making contact events and thus drastically decreasing the computation and transmission time allowing for faster detection.

As more features are included to the sensor-level perception, the mode-interface should be updated so it to greater extend resembles a pub/sub communication pattern. This would allow exact tailoring of the information to fit any type of task.

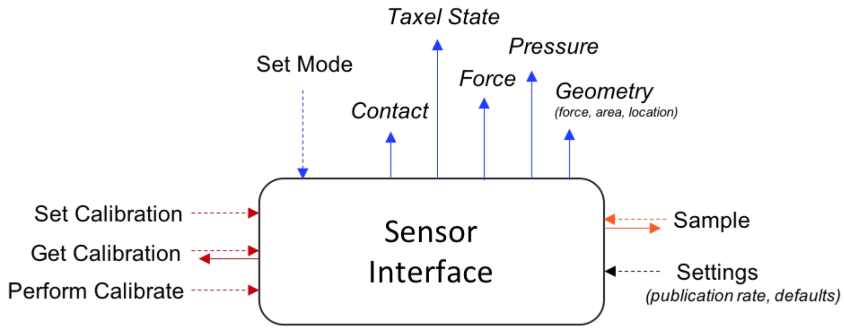


Figure 3.14: Services and messages of the ROS-integrated sensor interface

Besides the mode-*service* together with its corresponding information *messages* the interface defines a set of services responsible for setting, receiving or resetting the current calibration parameters.

It is also possible to tell the sensor to auto-calibrate to an unpressed state to account for any long term drift in the sensor. After calibration the parameters may be saved to the internal ROM of the Arduino in order as a default. Other setting like the rate at which the information is transmitted may also be set. The sampling service is used for providing average measurements over a given time period and is implemented in order to carry out test and manual calibration like in 3.3.3.

Serial Interface
The serial connection relies on the UART (universal asynchronous receiver-transmitter) protocol which transmits the data in packets of 5-9 data bits and 2-4 bits of overhead. The connection is configured to use a single start and stop bit, no parity bit and 8 bits for the data frame. By specifying the maximum baud rate allowed by the Arduino of $f_s = 115200$, this result in a transfer rate of $R_{byte} = 11520\, bytes/s$ or a transmission time of 86.8 microseconds per byte.

As a trade-off between speed and precision the datatype used for the data processing on the microcontroller is 32-bit floating point. This results in the theoretical transmission values as seen in table 3.2.

Mode	Bytes	Time [μs]
Taxel State	$2 \times 6 \times 4$	4167
Contact	1	87
Force	2×4	694
Pressure	2×4	694
Geometry	$2 \times 3 \times 4$	2083

Table 3.2: Transmission times for each mode

Another very used wiring technique and communication protocol within robotic systems is the CAN-bus. Its differential based signal representation renders it much more robust against noise and it has speed range up to $1Mbps$, more than 8 times that of the UART connection.

The transmission time estimates of the table indicate that the build-in serial port on the microcontroller is sufficient for the focus application, but the CAN-bus interface should be a strong consideration in later development and external CAN-bus modules do exist for Arduino.

3.6 Grasping Control

The design and implementation of the control relies on each of the previous design steps and thus concludes the sensing system as a whole. This should provide the basis for understanding the possibilities of tactile control but also for evaluating the system and the influence of the design choices on the final execution.

3.6.1 Sensor incorporation

The sensors are mounted in a static manner on the gripper facing inwards. Two custom fingers were designed and 3D printed in replacement of the original metal fingers in order to create stable support for the two sensors.

The information received on the tactile level needs to be transformed to world coordinates in order to determine the contact point of the detected object in absolute terms. The sensor coordinate frame is defined such that the origin lies at the center of the sensing area with the x-axis pointing along the taxel row and the z-axis pointing out of the sensor pad as illustrated in figure 3.15. It important that this matches with the contact point computations described in section 3.4.3 defining $x = 0$ at the same center location.

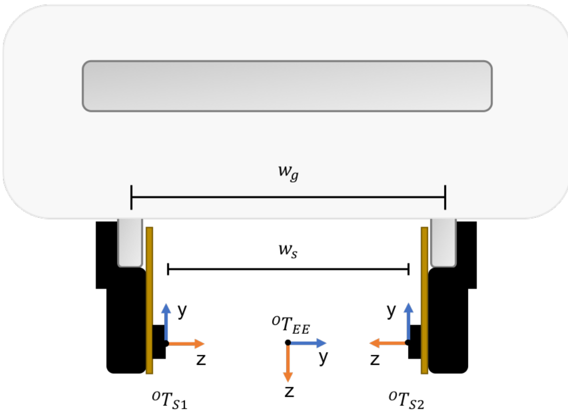


Figure 3.15: Sensor coordinate frame

The placement of the sensors are chosen such that the sensing area match with the original fingers along the x and y coordinates. This means that the end-effector frame is essentially the same. The difference between commanded and actual gripper width is compensated by including an offset term.

$$w_s = w_g + w_{off} \tag{3.18}$$

By closing the gripper around an object of known width the constant is determined as

$$w_{off} = -0.002$$

This makes it possible to define the two homogeneous transformation matrices relating the sensor frames with the end-effector frame.

$$^{EE}T_{S1} = \begin{bmatrix} 1 & 0 & 0 & x_{off,1} \\ 0 & 0 & 1 & -\frac{w_s}{2} \\ 0 & -1 & 0 & -y_{off,1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad ^{EE}T_{S2} = \begin{bmatrix} -1 & 0 & 0 & -x_{off,2} \\ 0 & 0 & -1 & \frac{w_s}{2} \\ 0 & -1 & 0 & -y_{off,2} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.19}$$

Due the placement described above $x_{off} = 0$ and $y_{off} = 0$ in both cases. It should be noted that these transformations are not constant but depend on the current state of the gripper width.

A detected contact location on one of the sensors may be transformed to world coordinates by computing.

$$^{O}p_c = {}^{O}T_{EE}\, {}^{EE}T_S \begin{bmatrix} x_c \\ y_c \\ 0 \\ 1 \end{bmatrix} \tag{3.20}$$

Here $^{EE}T_S$ is representing the specific sensor in question and $^{O}T_{EE}$ the forward kinematic transformation matrix from base to end-effector provided by the robot interface. Since the sensor in use is one-dimensional y_c will always be 0.

Combining Force Estimates
Since the robot is equipped with force/torque sensors in every joint that enables the computation of an external force estimate at the end-effector, an interesting incorporation of the sensor would be the combination of this estimate together with that of the tactile sensor.

Using screw theory, the transformation of forces and torques can be computed using the adjoint map of the transformation matrix between frames. Thus transforming from end-effector to sensor frame is computed as the following.

$$\mathcal{F}_S = \left[Ad_{T_S^{EE}} \right]^T \mathcal{F}_{EE} \tag{3.21}$$

Where the wrench $\mathcal{F}$ describes the combined forces and moments as a stacked vector and the adjoint matrix is defined as below.

$$[Ad_T] = \begin{bmatrix} R & 0_{3\times3} \\ [p]_\times R & R \end{bmatrix} \quad \mathcal{F} = \begin{bmatrix} m \\ f \end{bmatrix} \tag{3.22}$$

In this case R is the top-left 3-by-3 rotation matrix and $[p]_\times$ is the skew-symmetric matrix of the top-right 3-by-1 origin of the sensor transformation matrix from 3.19.

The last element of $\mathcal{F}_S$ now describes the force estimate form the force/torque sensors transformed to z-basis of the sensor frame. By referring to this scalar as f_R and to the normal force provided by the tactile sensor as f_S these describe two estimates for the same force.

The tactile sensor has the benefit of higher accuracy at light touch whereas the force/-torque is able to register a higher force load. Therefore the two are combined by using a weight W that should be close to 0 at light touch and reached 1 when approaching the saturation limit of the sensor.

$$f_{combined} = W \cdot f_R + (W - 1) \cdot f_S \tag{3.23}$$

By defining a lower and upper threshold for when the measurement should rely entirely on the tactile sensor and the force/torque sensors respectively, the weight is defined as the piece-wise function:

$$W = \begin{cases} 1, & f_R >= f_{T1} \\ 0, & f_R < f_{T1} \text{ and } f_S <= f_{T0} \\ \frac{f_{T1}}{f_{T1} - f_{T2} + f_R - f_S}, & \text{otherwise} \end{cases}$$

When between the thresholds the weight follows the linear interpolation of the two values.

This combination means that if the robot attempts to reach a certain reference force based on the tactile sensor but an obstacle is blocking the arm without touching the sensing area, the control will only increase the applied force until a certain point where the force/torque measurements exceeds f_{T1}. This provides a safe fail for the tactile controller and means it may easily be hold off with a hand if something happens.

3.6.2 Control Strategy

One of the main purpose of the control design is to evaluate the previous steps in the implementation process. This also means that the primary goal is not to develop an optimal control strategy, but rather to establish a baseline for determining the possibilities of the sensing system and its benefits.

First of all, the fact that it is not possible to interact with the robot-hand in a real-time manner as described previously, means that the object localization cannot be carried out by simply closing the gripper around the object. Instead the localization must be carried out using one finger at a time.

This has the downside of allowing the object to move when touched, which is significant if the object is very light and therefore has a low friction coefficient or if it rolls easily. This will put an additional limit on the possible target objects.

For the sake of starting simple a few assumptions are made. First a all it is assumed that the target object is a squeezable fruit or vegetable with homogeneous mass distribution, a diameter of $\sim 5cm$ and is more or less spherical. A tomato as an example. Furthermore it is assumed that the location is partially known down to an uncertainty of $\sim 1cm$ obtained from a camera or similar and that it rests on a known surface.

This means that the initial contact can be made simply by moving the end-effector to the expected location while maintaining a gripper width corresponding to the expected object size (which could also be by using a camera) added the maximal location uncertainty plus a safety margin.

In order to be able to adjust the grasping pose without loosing track of the object, the robot needs to be able follow its surface. This can be achieved either by rolling, sliding or a combination of the two.

Sliding across the surface is subject to friction and by that more inclined to push the object. It means that the normal force needs to be small and regulated precisely with no oscillations.

The main problem with pure sliding is that the movement happens at a 90 degree angle to the direction at which the sensor is actually detecting the force. Since the sensing area is flat this means that the adjustment will easily end up relying on the taxels on the edge and their response to a force applied in an angle to the sensor.

Rolling, on the other hand, does not exert any tangential forces on the sensor, but is limited in its movement based on the size of the sensing pad. Another aspect is, that rolling around a curved surface may cause the rest of the gripper to move drastically, which is inconvenient if the rotational plane contains any obstacles. This would be the case if rotating in a plane that is not parallel to the surface where the target object is located. Figure 3.17 illustrates this on two simple object shapes.

In addition, since the sensing area is one-dimensional, it is only sensible to role along that direction on the surface. These two constraints make rolling effective only if the gripper is maintained vertical to the table when thinking in terms of the application in focus.

This suggests that controlling the vertical position of the gripper should rely primarily on sliding whereas adjustments in the horizontal plane is more reliable by rotating the gripper. In other words, the gripper should start off by aligning the normals of the sensor and the object surface using a rotational gesture before making any transnational adjustments.

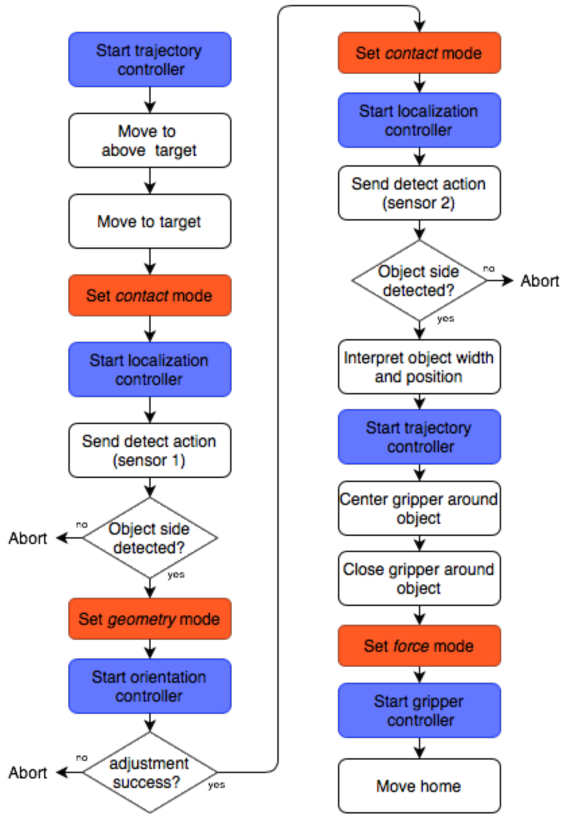


Figure 3.16: Control flow of the grasping procedure

This initial adjustment in the horizontal plane is what the implementation focuses on while assuming that the height of the objects is known. Figure 3.16 shows the control flow of the grasping procedure.

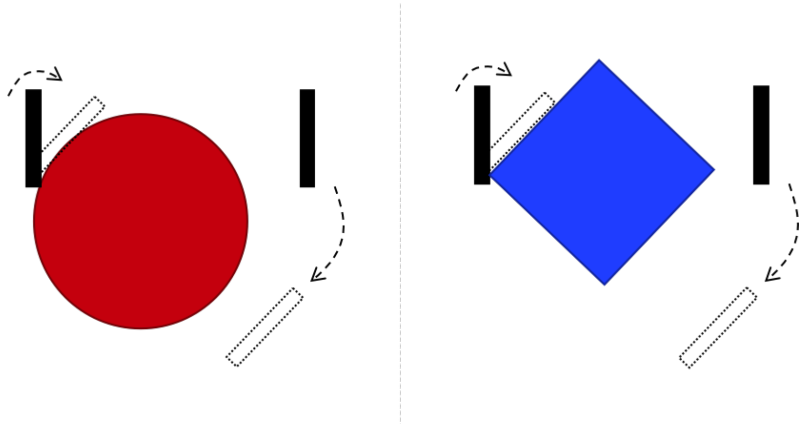


Figure 3.17: Rolling around an object surface. The black rectangles represent the sensing pad seen along the sensor y-axis.

Based on the previously mentioned assumption, the overall expectation is that a good contact location for grasping is easy to find from the initial contact. If that is not the case, the shape of the object should be taken into account in larger scale, either by more active tactile exploring or by determining a set of expected grasping pose from the visual input, which could be verified by the tactile input when attempting the grasp and perform re-grasping if it is not the case.

3.6.3 Localization Control

As described above, the target object should be detectable by simply performing a side-ways movement along the y-basis of the end effctor frame. The speed of this movement must allow for an abrupt break as soon as the object is touched. Failing to do this could cause the object to start rolling or get knocked over. Specifying a limit for maximum de-celeration of $3.0m/s^2$ together with an $0.1mm$ upper bound for the distance the gripper is allowed to travel after detection, means that the gripper is allowed to travel at velocity of $0.0245m/s$.

To achieve this the controller builds on the internal velocity controller of the robot frame-work, to ensure that the gripper moves at a maximal speed reference that allows the end-effector to come to a halt within the critical braking distance.

The relatively low deceleration provides a smooth stopping and is far within the limitations of the robot. In order to comply with the jerk restrictions of the robot[6], the change in ac-celeration is saturated according to a maximum jerk of $5000m/s^3$. This effectively causes the breaking distance to increase but with less than a micrometer.

3.6.4 Orientation Control

The orientation control is based on hybrid motion-force control. This means that the 6-dimensional task space is divided into subspaces that adopt either a force or a motion control law. The reason for this is to be able to apply continuous pressure on the surface of the object while performing the adjustment of the gripper pose.

The only natural constraint present is that along the contact normal which restricts the robot from moving into the object, at least as long as the object does not slip on the table. The remaining 5 constraints in the 6-dimensional task space of SE(3) is constrained artificially.

In order to achieve pure rolling around the surface as described in previous section, the movement tangential to the sensor should be constraint at the contact point. This however relies on the assumption that the contact area can be described as an infinitely small point, which in practice is not the case. Furthermore this would add the uncertainty of the contact point estimation to the motion part of the control, which may be avoided. Instead, the movement is constrained at the sensor frame origin causing the rotation to happen around the center of the sensing area. In this reference frame the natural constraint acts as a combination of force and torque, but due to the small width of the sensor the displacement may be ignored. This means that the assumption that sensor does not move in the force-controlled direction is essentially violated. However, since the object is not completely stiff and might not even be completely stationary, it could not be completely assured anyway.

Two measures are taken to resolve this issue. First of all a damping term is added to the control law in order to limit any unconstrained acceleration. Secondly, in case the sensor looses contact with the object completely, the control law is changed to match a velocity controller, until contact is made again and the reference position used for the motion control is updated accordingly.

[6] https://frankaemika.github.io/docs/control_parameters.html#cartesian-trajectory-requirements

The control laws are specified in the sensor frame and transformed to the end-effector frame by using the adjoint map as described in equation 3.22.

$$\mathcal{F}_{EE,d} = \left[Ad_{T_{EE}^{S}}\right]^{T} \mathcal{F}_{S,d} \tag{3.24}$$

The desired wrench is then transformed to base frame from which the commanded joint torques may be calculated in accordance with equation 2.2. To include velocity damping in the control, the twist is similarly converted to the sensor frame.

$$\mathcal{V}_{S} = \left[Ad_{T_{EE}^{S}}\right] \mathcal{V}_{EE} \tag{3.25}$$

The same control law is used for both the force control and task-space motion control

$$\mathcal{F}_{S,d} = K_p e + K_i \int e\,\mathrm{dt} + K_d \mathcal{V}_S \tag{3.26}$$

This is possible because the term $K_d\mathcal{V}_S$ acts as velocity damping when representing force control but as a differential term when the error describes the configuration error in task space. A low-pass filter is applied to the tactile force feedback in order to suppress noise in the signal and a $0.5N$ reference force is used as the desired contact force.

The estimated contact location x_c received from the tactile sensor is a value in the range $[-1;1]$ with 0 representing a central location. The orientation reference along the gripper y-axis is thus continuously updated as follows.

$$\theta_y^r = \frac{x_c}{\hat{R}_{obj}} - \theta_y \tag{3.27}$$

Here $\hat{R}_{obj}$ is the approximated radius of the object, which is in the implementation fixed to 0.05.

3.6.5 Gripper Control

Two major problems concern the control of the gripper width. The first being the slow processing of the gripper commands. The other being the drifting nature of the sensory input. Individual control of each finger is not possible instead the fingers are moved by commanding a desired gripper width to the hand.

Due to its slow response time, the gripper cannot be stopped abruptly. The control action may be canceled in order to stop the gripper movement, but the exact response time varies which makes it impossible to enforce any hard limits on how much it is allowed to squeeze the object. Because of this the fingers must be controlled by executing small steps of change in width. This means that a delay in response will not result in the gripper continuing to close, but has the consequence that the closing speed depends on the connection speed and the maximum allowed step size.

The commanded change in gripper width is computed as:

$$\Delta w_s = \min\left(\max\left(K_p \cdot (F - F^r),\, \Delta w_{max}\right),\, -\Delta w_{max}\right) \tag{3.28}$$

Here F^r is the force reference specified in the ROS-action sent from the Motion Interface and F is the sum of the two normal force estimates provided by the tactile sensors. The upper limit Δw_{max} defines the maximum step size and may be specified based on the knowledge of the object.

The relationship between the measured input as a *force* and the commanded change in *position* means that a proportional controller will theoretically have no static error, however in practice a minimum change in gripper width of $0.1mm$ is required before sending the action. When this lower limit is reached the controller responds with success, and if any of the sensors drop below a critical force of $0.001N$ after this, the event is registered as a drop.

4 Testing and Evaluation

4.1 Sampling and Computation Time

In relation to the calculation of sampling time in section 3.3.1 the actual time used for scanning the sensor is measured and compared below in table 4.1. It also shows the time used on adjusting the raw data based on the calibration parameters.

	Calculated	Measured
Scan Time	1248μs	1359μs
Adjustment	-	602μs

Table 4.1: Calculated and measured scanning time

The higher measured scanning time is probably due to overhead in the function call and the switching of the multiplexer. The Arduino also has to switch its internal multiplexer between the the input port which would contribute a tiny bit as well.

The computation time of pre-processing the tactile information is also measured and the average timing are seen in table 4.2.

Mode	Processing Time [μs]
Full taxel state	0
Contact	56
Force	141 / 146
Pressure	169 / 278
Geometry	680 / 1211

Table 4.2: Measured times for information processing.

Most of the measurements vary based on the input, and the timings are shown for activating only one or both of the sensors. The most significant variation is seen in the Geometry computation varying between [96μs; 1416μs].

In figure 4.1 the measurements are combined with the transferring times calculated and shown in table 3.2. To predict what would happen if the number of taxels were increased, the right-hand-side plot shows a scaled according to the number of taxels.

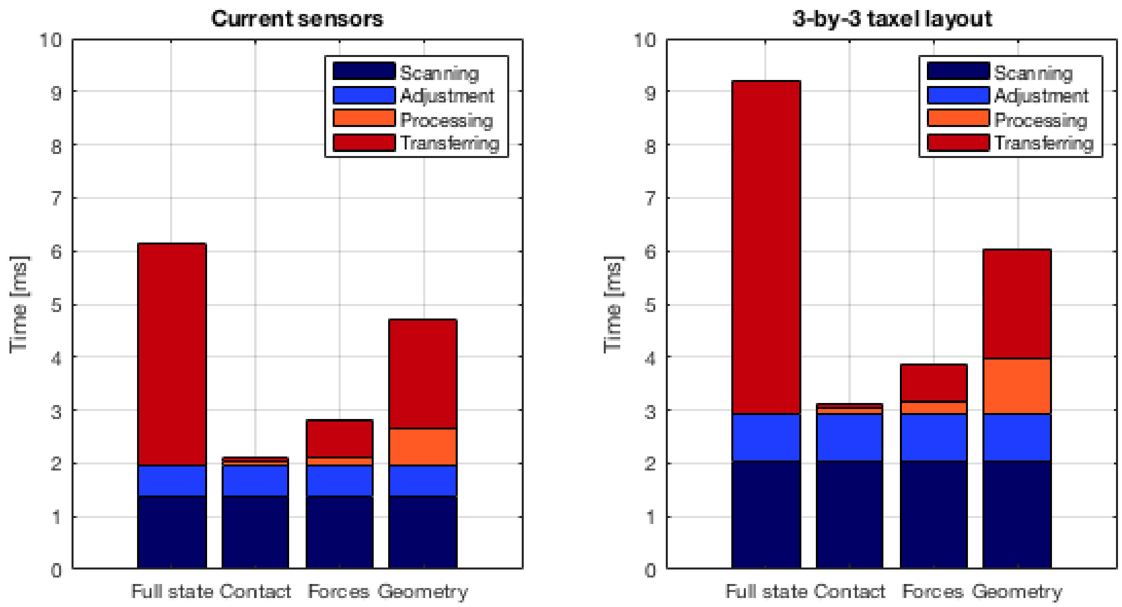


Figure 4.1: Timings of the information processing and transmission

These results clearly shows the benefit of pre-processing the information especially if the number of taxels in the sensor is increased. In some cases pre-processing will be a trade-off between the slower computation speed of the microcontroller against having to transfer less data, but as seen from the measurements simple preparation of the raw data is quite a lot faster than transferring it.

Since the calculations performed by the microcontroller are generally simple, the limiting factors are the scanning time and transferring time. Especially if increasing the sensor size the scanning time start to play a bigger role.

Changing the prescaler of the ADC is one solution, but with the consequence of reducing accuracy. Instead several ADCs may be used in parallel. This is more relevant when the taxels are arranged in an array of $n \times m$, since only $min(n, m)$ ADCs are then needed. Both the scanning time and computation time could be halved by using a microcontroller for each sensor.

An optimization technique for lowering both computation time and transferring time would be to use fixed point arithmetic and lower the data type to a 16-bit int. This would half the transfer time, but the amount of time saved on computations is questionable and the resulting loss of maintainability means that this should be done only if strictly necessary. A better approach would be to update the transmission line and communication protocol as already mentioned in section 3.5.3.

4.2 Gripper Control

The control of the gripper is tested by placing a tomato between the fingers, such that it barely touches each side and hereafter commanding a reference force causing it to close around the object. The time-response of the measured force is in seen in figure 4.2.

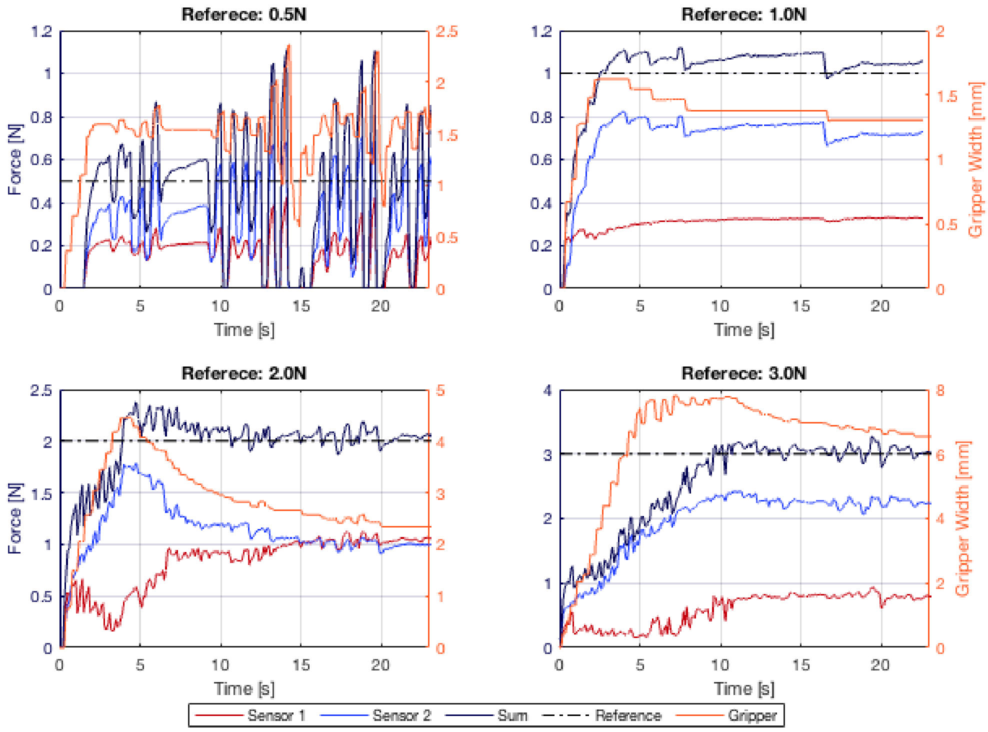


Figure 4.2: Gripper control performed on a tomato using different force references

Besides the reference and the summed force as defined in the control law 3.28, the individual sensor measurements and also the actual gripper width are shown.

It is very visible how different the two sensors respond, which matches with the calibration results of section 3.3.3 showing quite different behavior for force above $1N$. An alternate approach would be to rely entirely on *sensor 2* since it produces the more reliable response, but that would be a very temporary solution.

The strange divergent behavior between the sensors on the plot for the $2.0N$ reference, may be an indication that the object was not complete centered between the finger causing frictional force between the object and underlying surface to influence the measurements.

The control seems to work well for references of $1N$ and above, however with heavily increasing rise time. The unstable behavior in the measurements with 0.5N reference seems to arise from the relative high noise level of the sensors due to increasing sensitivity at lower force levels entering the control loop. Applying a low-pass filter on the force response could help to solve this issue.

It should be noted how the the gripper movement (orange line) seems to spike and then afterwards starts to relax its grip on the object even though the measured force seems to have settled at the reference force. This can be explained be the incremental drift in sensor value also seen in figure 3.8. Modeling and then counteracting this time-dependent behavior would be extremely difficult especially because small disturbances, seem to "reset" the drift.

The solution would be to either increase the reference grasping force to ensure that the object is not dropped but thus squeezing the object more than necessary in the beginning or to simply wait for the measurements to settle before attempting the lift.

The reference force used for grasping the tomato in the proceeding tests is $1.5N$ to make sure the sensor inaccuracies does not cause a drop while still leaving some a gap to the $3N$ limit. Section B.7 shows similar tests performed with different types objects.

4.3 Grasping Tests

In order to evaluate the sensing system as a whole a set of grasping tasks are carried out in accordance with the strategy described in section 3.6.2. Three tests are performed, two of which assess the rejection of disturbances in position and orientation to comply with the overall requirements of section 3.1, and one that attempts to lay out the limitations in a qualitative manner by introducing other target objects. The image in figure 4.3 shows the different objects used in these following set of tests.

Figure 4.3: Objects used for testing

4.3.1 Testing object orientation and displacement

Displacement

The requirements specify that the robot should be able to grasp the object if the position is known to an accuracy of around a couple of centimeters. The direction in which the uncertainty actually plays a significant role is when the object is displaced tangential to the sensors. This influence is tested by adjusting the position of a mandarin and noting successful grasps and failures. The requirements for a successful grasp is that the object is lifted in a stable grasp and that the procedure doesn't leave any marks that can be seen or sensed with a finger.

For comparison, the same tests are carried out using a baseline strategy relying only on the internal force sensing of the gripper but otherwise performing the grasp without feedback about the environment.

Each grasp was attempted 5 times at each location the results are shown in table 4.3. Also the cause for failure is indicated.

Δx $[cm]$	Activated		Deactivated	
	Lifted	*Cause*	*Lifted*	*Cause*
-3.0	0/5	P	0/5	P, D
-2.5	4/5	P	5/5	
-1.0	5/5		5/5	
0.0	5/5		5/5	
1.0	5/5		5/5	
2.5	5/5		0/5	P, D
3.0	0/5	P	0/5	P

Table 4.3: Results of displacement test. P: pushed away during localization or gripping, D: dropped during lift

Furthermore, all the grasps performed with the tactile sensing deactivated caused clear deformations in the mandarins, whereas the procedure using the tactile sensing didn't mark any of them.

The results are visualized in figure 4.4 where the green columns indicate successful lifts and red indicates grasps that damaged the fruit. It is clear that even if disregarding the harshness of the grasps, the tactile sensing still excels by letting the gripper adjust to the position. It should also be noted that since the baseline execution does not make adjustments, the mandarin is lifted off-center and only sustained by the hard squeeze of it. The max displacement is of course dependent on the size of the object and the

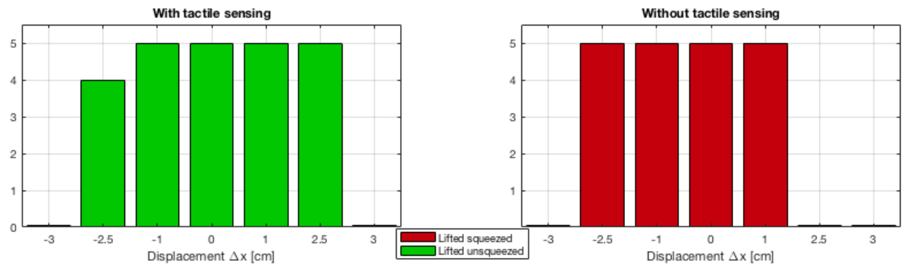


Figure 4.4: Displacement

observed requirement for successfully adjusting is that at least the edge-taxel is touched.

Orientation

For testing how disturbances in orientation affects the performance of the system, a quadratic 5cm $\times$ 5cm $\times$ 5cm visible in figure 4.3 is used. This object type has not been strictly within the focus of the strategy, but does however provide a good insight into the performance.

Since the object is symmetric the test is carried out by placing it at 8 different angles between 0 and 90 degrees with the same center. Each grasp is attempted 3 times and results are presented in figure 4.5 in similar manner as the displacement test.

A successful and unsuccessful lift is discriminated by whether the box can be wiggled within the grip. This is the case if grasped around the edges instead of on the sides. The

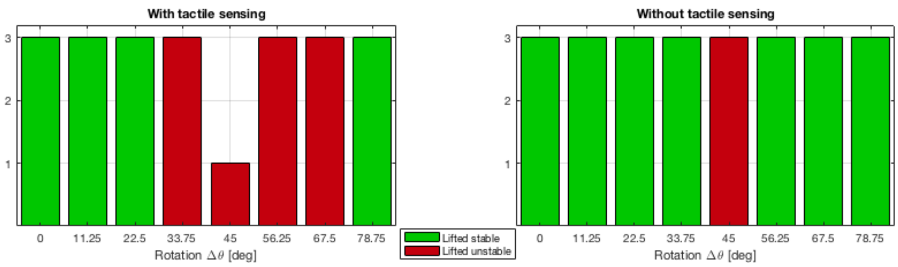


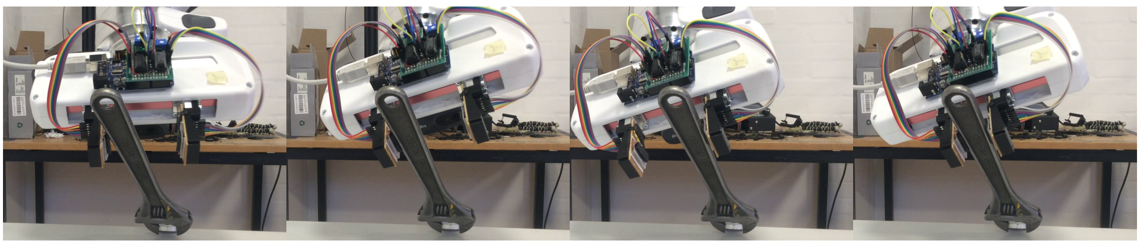
Figure 4.5: Orientation

reason for this happening comes from the simplicity of the grasping strategy and from the illustration in figure 3.17 it may be seen how the box ends up being grasped off-center.

A possible way to fix this issue would be to implement edge-detection into the sensing system and based on the make a decision to re-grasp, ensuring that the whole sensing area is covered before closing the grippers.

The apparent superior performance of the baseline strategy comes from the fact that the object is not restricted in any way and can freely slide on the table. So simply grasping hard around the object, will cause it to align with the finger. This wouldn't however be the case if the object was constraint to its orientation, like if placed in a equally shaped square hole.

Figure 4.6 shows a demonstration of a orientation task where the object is constrained. The wrench is fixed and in order for the fingers to get a proper grip the orientation needs to be correct, since the wrench itself will not comply.

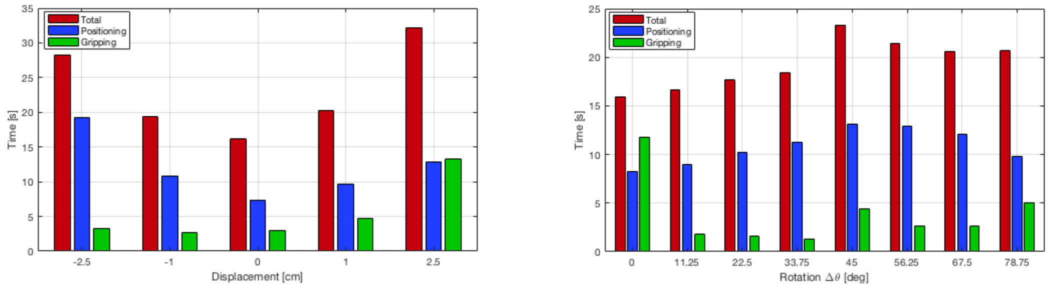


$t = 3s$ $t = 11s$ $t = 13s$ $t = 17s$

Figure 4.6: Robot adjusting its gripping position to get a proper grip around a wrench

Execution Time

In the two previous tests the execution time was logged and averaged for each position or orientation. These are plotted in figure 4.7. For comparison the baseline strategy had a consistent execution time of 6.7s (1.4s spend closing the gripper) with a max variation of 0.3s.



Displacement Orientation

Figure 4.7: Execution time for displacement and orientation adjustment tests. The total time includes moving to and from the target, positioning the gripper, and closing the fingers.

It can be noted that the pre-grasp positioning of the gripper performs faster in one direction than the other. This is most probably due to the asymmetry in taxel response, as seen in figure 3.9, which despite of the calibration manage to affect the contact point computation. Especially the activation level of the edge-taxels is influencing this.

In a few cases the gripper control has a hard time stabilizing around the reference value and thus delays the procedure significantly before continuing. As discussed in the test of the gripper control, a filter may help in solving this issue as well. The positive side of it is that the robot does not attempt to lift the object before making sure that it is properly grasped. Generally the gripping time of a couple a seconds is also slow, and comes directly from the issues with the gripper response time. Solution for this is discussed later.

The focus on success rate compared to time efficiency is clear from the results, and the conclusion must be that optimization of the orientation control with with timing requirements in mind is needed for making the adjustment useful in a practical application.

Generally the two tests show, that for object where an imprecise object position can be compensated by simply squeezing the object hard, the sensing system in its current form does not create an advantage. However, if dealing with fragile objects or symmetric shapes that must be picked from a certain angle the tactile sensing system makes this possible.

4.3.2 Testing various objects

In order to test the limitations of the system, but also to provide further insight into the preceding steps needed to improve the grasping procedure, the grasping strategy is performed on a quantity of different objects.

The focus is to survey different causes of failure and thus number of attempted grasps of each object varies. The objects where placed in a random manner near the target center, but in cases more deliberately to challenge the system. This also means the success rate should be taken as an indication of ability and not an exact measure for the repeatability.

The results are categorized into three categories, the first classifying a *successful* stable and non-damaging grasp, the second describing an *acceptable* non-damaging but less stable grip and the last being *failure* due to any of the reasons noted under *cause*. The results are shown in table 4.4. The *rough surface* is created by the rubber pad also visible in figure 4.3.

Object	With tactile sensing				Without tactile sensing			
	Succ.	*Acc.*	*Fail.*	*Cause*	*Succ.*	*Acc.*	*Fail.*	*Cause*
Soft tomato	8/10	2/10	0/10	(A) Imperfect grip	0/1	0/1	1/1	(F) Ripped open
Hard tomato	7/11	2/11	2/11	(A) Imperfect grip (F) Pushed away (F) Tiny visible mark	0/1	0/1	1/1	(F) Ripped open
Sponge (rough surface)	7/9	0/9	2/9	(F) Angle too steep *(Doesn't touch sensor)* (F) Too wide	0/1	1/1	0/1	(A) Completely squeezed *(not detectable)*
Sponge (slippery surface)	0/1	0/1	1/1	(F) No detection	0/1	1/1	0/1	(A) Completely squeezed *(not detectable)*
Glass (rough surface)	9/10	0/10	1/10	(F) Runtime error	1/4	1/4	2/4	(A) Imperfect grip (F) Pushed away
Weight (511g)	0/3	0/3	3/3	(F) Dropped during lift *(sensor pad slips)*	0/3	0/3	3/3	(F) Dropped during lift *(sensor pad slips)*
Egg (rough surface)	10/13	0/13	3/13	(F) Bad positioning (F) Bad positioning due to rolling (F) No detection due to rolling	0/4	0/4	4/4	(F) Unsuccessful grasp (F) Dropped during lift (F) crushed*

Table 4.4: Results from testing the tactile sensing strategy and baseline strategy on the various test objects. *The egg was crushed when using the original metal finger.

The two tomatoes used for testing are slightly different with one being softer, more spherical and slightly bigger, and the other being harder and a bit more oval in its shape. The result from the grasping attempts without tactile sensing, clearly shows that the internal force sensing in the gripper is not sensitive enough to avoid damaging the tomatoes when closing the gripper.

With the tactile sensors activated the robot manage to successfully grasp the tomatoes in many of the attempts. However, it is not always perfect in its adjustment. This is due to the simplistic solution that expects the center of the object to lie in extension of surface normal, which is not true for an oval shape. The error could be minimized by performing the orientation control in an alternating pattern between the to sensors and not attempt to close the gripper before both sensors were normal to the surface. But it is not clear whether there would be an upper bound on the number of needed iterations.

The mark caused in one of the attempts was induced by the corner of the sensor board due to improper adjustment, and the other failure due to the placement and orientation of the tomato causing it to be hit by the edge of the finger before the actual sensing area.

An inherent issue with the adjustment strategy can be seen from the sponge test. If resting normally on the table the gripper will push the object along when trying to localize it, without providing the frictional force needed to activate the sensor. Of course a more sensitive sensor would be able to solve this, but the real problem arises from the fact that the other finger cannot be used as a backstop due to response time issue. The same problem is present with the egg that easily rolls.

Since there's a gap from the sensing area to the edge of the finger, there is a limit to how much an object like the sponge can be angled and still touch the sensor. Also the widths of the fingers poses as a limiting factor is this case. It should be noted that even if the baseline strategy perfectly grasps the sponge, it is not able to tell whether something is actually held between the fingers.

The test using the weight of half a kilo shows that there's a weight limit to what the sensors can withstand. It was observed that even with increasingly applied gripping force, the rubber pads of the sensors would simply start to slide of causing the weight to drop. This indicates a common problem for externally mounted sensors, namely robustness and resistance to wear.

Looking at the results as a whole, there is a clear indication that having the sensing system in place allows the robot to grasp certain easily damaged objects which is not possible without it using a two-fingered hard gripper.

5 Discussion

5.1 Usefulness of Tactile Sensing

Broadly speaking the use of tactile sensing results effective in unstructured environments where sensory feedback in general plays an important role. That being the case when some aspects of the environment is initially unknown.

This study has primarily focused on the use of tactile sensing when performing grasping tasks. The results show, that using tactile sensors in the fingers of a robot arm is helpful for adjusting the gripper to an objects position in order to improve the stability of the grasp. If a camera is present in the robotic setup, this means that it may compensate for visual inaccuracies when locating the object, but also provide information about the object within the grip even if the object is completely or partially blocked from the view of the camera.

The implemented sensing system is especially useful when exact positioning of the gripper is required for performing a proper grasp, like picking an object from a constraint location where the object itself will not adapt to the position of the gripper. The test results also show that extrinsic tactile sensing is beneficial for handling fragile objects, due to its higher sensitivity when compared to intrinsic force sensors. This means that tactile sensing can provide the possibility for grasping easily damaged objects like fruits, vegetables and eggs.

This advantage should however be viewed in comparison with other ways of manipulating fragile objects. Soft grippers of flexible materials controlled by strings, airflow or electrical stimulation provide eminent solutions to such tasks as well. The intrinsic compliance their materials reduce the need for precise control and complex perception, making them simple solutions for industrial applications. On the other hand the softness of the gripper result in lost precision when it comes to picking and placing. Thus in cases where the robot needs precise control over the environment or must be able to exert accurate forces, the use of tactile sensing should be preferred.

Repeated grasping of identical objects can in many cases be achieved by using a custom gripper design for the specific target object which then doesn't require complex perception and control. However, if the robot need to able to handle various objects or perform multiple different tasks, tactile feedback helps in increasing the dexterity of a universal gripping tools.

Tactile sensors may also provide information about the environment that is not measurable using cameras or soft grippers alone. As showed, the compliance of an object may be measured using the tactile feedback in relation with the precise movement of the fingers, useful for identifying the object or its condition.

5.1.1 Possible applications

From the preceding discussion, the advantages that arise from the improved sensitivity when using extrinsic tactile sensing, may be categorized into three types.

- Improved dexterity in manipulation
- Increased carefulness in execution
- Additional information about the environment

The improved dexterity means that tactile sensing facilitates the ability to handle small parts, where pure visual input is too inaccurate. This includes assembling tasks where the act of inserting a peg into a hole and similar operations require the sensitive feedback provided by extrinsic sensors.

With the implemented sensing system in mind, handling objects that are partially constraint would need the grippers ability to adjust precisely to the object before attempting any manipulation. Turning knobs, pulling handles or in other ways moving parts that are connected using hinges or similar would be relevant applications.

With the currently implemented grasping strategy the robot should be able to get a good and centered grip around a screw cap, as one example. Also in tool manipulation, where the robot should be able to use tools designed for human interaction, a proper grip is of outmost importance, and if multiple tools are required for a task it might be impossible or impracticable to design custom grippers whenever a new tool is introduced.

Grasping of completely unknown objects and more complex in-hand manipulations are also operations where tactile sensing play an essential role, but these are at the moment more relevant in research setting and their introduction in industrial applications should not be expected soon. Industry tend to implement solutions that are simple and reliable.

The ability to handle objects gently may be useful in the food industry where object are often damageable. Especially picking fruit and vegetables, as seen in the tests, poses a relevant application. As the tactile input is also able to provide information about the hardness of the object, this may be able to tell the ripeness of the fruit before harvesting it, which would not be possible with a simple soft gripper.

Careful execution is also a relevant factor for human-robot interaction. Being able to distinguish the compliance when grasping a human finger instead of a given target object could significantly reduce the risk of working next to a robot. Externally placed tactile sensor also have the benefit of keeping the same sensitivity independent on the configuration of the robot.

5.2 Assessment of the Sensor

As the sensing system is designed to work well with the specific sensors, both their performances and physical aspects plays an affects on the system as a whole. The sensor, as a prototype product, cannot be expected to perform optimally and therefore naturally limits the possibilities and performance of final control. Some of the facets of the sensor and their influence on the system and the grasping task is discussed in the following.

First of all the one-dimensional taxel layout of the current sensors is rather insufficient for proper adjustment during the pre-grasp. Having to rotate or move the gripper in order to extract geometry information orthogonal to the taxel row is both impractical and time consuming. If the aim is to improve manipulation, the further development of the sensor should definitely focus on arranging the taxels in a two-dimensional structure.

The sensitivity range of the sensor is on the edge of being able to perform universal grasping tasks. This is especially true for *sensor 1* that saturates around $0.5N$. The maximum immediately detectable force is just a couple of newtons and comparing with the needed force of $\sim 8N$ to lift some the testing objects (see B.2) the grasping applications are limited. The irregularities between the taxels is a key factor in this since the maximum force that may be detected in any circumstances is bounded by the least sensitive taxel.

It should be noted that the low force range of *sensor 2* is partially due to the chosen calibration. The sudden change in the resistive behavior of the sensor means that performing a piecewise fitting of the model would improve its accuracy on forces above $1N$. This would also allow the detection of higher forces and the exact force limit of the sensor can therefore not be precisely determined from the presented results.

An important factor about the sensor construction is that it seems to provide a linear relation between applied force and conductance which facilitates the simple analytical modeling of the sensor input. It may also be the reason why the sensor does not seem to be subject to hysteresis, something that is otherwise typical for resistive sensors. This should be kept in mind if changing the structural grading of the taxels in attempt to optimize the dynamic range.

The settling time of the taxel response is a major problem when intending to do accurate real-time control based on the input. This short term drift manifest itself as an continuous increase in measured pressure which in a grasping task causes the gripper to loosen its grip. This a crucial problem that is especially hard to counteract due to it being influenced by both time, varying applied force and properties of the rubber. It must be minimize in the physical design and should be a key focus point in the further development of the sensor.

As mentioned previously, having a gap between the rubber and the grading creates a clear difference between touch and untouched, but it also creates a physical threshold for the minimum detectable force. This has importance if the conductive rubber in not completely planar which will cause the activation of the taxels to happen irregularity and make the input much harder to model and predict. The positive effect of having the gap is that it limits the flow of current through the sensor when untouched, decreasing the power consumption.

The most relevant factor regarding the physical aspect of the sensors is robustness. For industrial applications resistance to wear and tear is critical and due to the exposed placement of the sensor, it will be subject to collision, changing temperature, humidity, dust etc.

5.3 Assessment of Hardware

The need of low-pass filtering the force signal when used in the orientation control together with the results of the gripper control tests, indicate that some measures may be taken to lower the introduction and amplification of noise in the sampling circuit. Operational amplifiers generally help in reducing noise and a further investigation into the choice of amplifier may prove useful and help determine the cause of noise amplification when used with *sensor 1*. As mentioned in section 3.3.3, operational amplifiers are often required for suppressing crosstalk in 2D resistive sensors, though some methods exists without the use of amplifier but instead require complex computations[21].

The use of multiplexers are likely to introduce an accuracy error into the system. Multiplexing is not strictly necessary if relying on a setup with enough ADCs, however for easier scalability using multiplexer is a benefit, especially at a development stage where the size of the sensor may change. The internal resistance of the multiplexer may instead be estimated and included in the model for improved accuracy.

Long wires carrying analog signals should generally be avoided and thus having a central microprocessor sampling both of the sensors is not the optimal solution, since the connecting wires must be long enough to allow the gripper fingers to move. Preferably the circuit and microprocessor should be contained within the same PCB board and act as a *System on a Chip* solution.

When having to do more complex computations like surface curvature and edge detection in bigger taxel arrays the computation time may become a problem when the information is needed for low-level control. This is especially the case if having to do Fourier transformations in order to detect slip-events at a later stage in the development. It has not been analyzed whether the current sensor would be able to reliably provide high frequency responses for slip detection, but with slip detection requiring faster than $1\,kHz$ response rate, the measured sampling time of $1359\mu s$ corresponding to a frequency of $0.7\,kHz$ does not cut.

A possible upgrade for improving computation and sampling time would be to use a Teensy 4.0[1]. With its ARM Cortex-M7 processor the Teensy has a processor speed of 600MHz compared to the mere 16MHz of the ATmega328P on the Arduino Uno board. The Teensy also has a built in CAN bus interface, which could be used to improve the transferring speed of the pre-processed information. This replacement should be relatively easy as the Teensy is programmable using the same libraries and software tools as the Arduino.

5.4 General Issues and Further Improvements

One of the main issues encountered during the design of the system, is the inability to control the gripper in real time. This means that the localization of the object has to be done using just one of the fingers, allowing the objects to slide or roll away. It also heavily increases the time it takes to grip around the object in a gentle manner.

Adding proximity sensors to the fingers may solve half the problem of initially localizing the object, but any exploratory control would still have to rely on one finger only and adjusting the gripping force would still be slow and unreliable in its response time.

It is not possible to estimate if this is a common issue with commercially available grippers, but according to *OnRobot A/S* their *RG2* and *RG6* grippers does not allow for real-time control, while the *2F-85* and *2F-140* grippers from Robotiq have a response rate up to $200Hz$ [2].

Tactile perception is inherently connected with the control of the fingers and thus the information and control flow between sensors and fingers should be as short as possible. Ideally the tactile sensing system should be implemented within the robotic hand to allow this. Another solution would be to let the control be part of the sensing system. By using a small servo motor to displace the sensor within the finger, it would be able adjust directly based on the tactile information and function as a closed solution not having to rely on the response time of the gripper.

The implemented control strategy serves the purpose of evaluating the possibilities of tactile sensing, but for robust and reliable grasping in an industrial application a more intelligent grasping technique is required. Tuning of the individual controllers is also necessary to reduce the execution time of the grasping procedure. This should be done using a more technical manner than the current manual approach and rely on the model of the specific robot.

[1] https://www.pjrc.com/store/teensy40.html

[2] https://assets.robotiq.com/website-assets/support_documents/document/2F-85_2F-140_UR_PDF_20200211.pdf?_ga=2.140351603.1376946866.1611580662-2041040198.1548007754

6 Conclusion

In this thesis, a tactile sensing system was created in order to develop an understanding for the possibilities and requirements of tactile perception in robotics. Using an Arduino Uno, a sampling circuit was constructed capable of reading the state of two resistive based prototype sensors and process the data in real time. This allowed for the perception of contact force, pressure, location and object compliance. By the construction of a ROS-based interface the sensing system was included into a setup with a robotic arm. This allowed the tactile information to be combined with measurements from the internal sensors of the robot to create a haptic perception of the environment.

For testing the system a grasping task was proposed consisting of picking different objects from a table without knowing their exact location. Based on a simple control strategy the robot achieved to locate and grasp mandarins and tomatoes without squeezing them. Compared to a baseline strategy with the robot grasping ”blindly” without any sense of touch, the tactile sensing system was shown to exceed its performance when handling fragile object. The system also showed the ability to orient its gripper to align with the surface of an object, which is necessary to get a proper grip if the target object is constraint in its movement.

The grasping strategy was tested on different objects with varying success rate. The execution performed robustly on round symmetric objects, but was not able to grasp very light or heavy objects. From the tests, it was concluded that even simple gripper adjustments based on tactile feedback can prove useful in tasks where the target objects are partially known. The study concludes that tactile sensing may be advantageous in industrial applications where fragile objects need to be handled without loosing the dexterity that a robot with fingers provide. In addition, tactile perception may prove beneficial in tasks where visual perception is insufficient, like when handling small parts in assembling tasks.

The design process has shown that in order to perform good control based on the tactile contact information, the sensors must provide reliable measurements with continuously increasing taxel response as the load increases. The taxels of the sensor should behave similarly and have the same dynamic range to allow for accurate analytical processing of the information. Drift in the sensor signal is especially difficult to compensate and must be minimized if the ability to apply precise pressure to an object is a key issue. To allow for implementation in an industrial setting the sensors must be extremely durable and insensitive to dust and harsh environments.

It was further discovered that commercially available grippers, as typically used in industrial settings, may not necessarily provide the response rate needed to perform both fast and precise grasping when the sensors are statically fixed to the fingers of the gripper.

Bibliography

[1] C. Chi, X. Sun, N. Xue, T. Li, and C. Liu, "Recent progress in technologies for tactile sensors," *Sensors (Switzerland)*, vol. 18, no. 4, 2018, ISSN: 14248220. DOI: 10.3390/s18040948.

[2] R. S. Dahiya, M. Valle, and G. Metta, "System approach: A paradigm for robotic tactile sensing," *International Workshop on Advanced Motion Control, AMC*, vol. 1, pp. 110–115, 2008. DOI: 10.1109/AMC.2008.4516050.

[3] R. S. Dahiya, G. Metta, M. Valle, and G. Sandini, "Tactile Sensing - From humans to humanoids," *IEEE Transactions on Robotics*, vol. 26, no. 1, pp. 1–20, 2009.

[4] M. A. Abd, I. J. Gonzalez, T. C. Colestock, B. A. Kent, and E. D. Engeberg, "Direction of slip detection for adaptive grasp force control with a dexterous robotic hand," *IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM*, vol. 2018-July, pp. 21–27, 2018. DOI: 10.1109/AIM.2018.8452704.

[5] P. S. Girão, P. M. P. Ramos, O. Postolache, and J. Miguel Dias Pereira, "Tactile sensors for robotic applications," *Measurement: Journal of the International Measurement Confederation*, vol. 46, no. 3, pp. 1257–1271, 2013, ISSN: 02632241. DOI: 10.1016/j.measurement.2012.11.015.

[6] J. Tegin and J. Wikander, "Tactile sensing in intelligent robotic manipulation - A review," *Industrial Robot*, vol. 32, no. 1, pp. 64–70, 2005, ISSN: 0143991X. DOI: 10.1108/01439910510573318.

[7] R. D. Howe, "Tactile Sensing and Control of Robotic Manipulation," *Journal of Advanced Robotics*, vol. 8, no. 3, pp. 245–261, 1994.

[8] T. Taunyazov, W. Sng, H. H. See, and B. Lim, "Event-Driven Visual-Tactile Sensing and Learning for Robots," *Robotics: Science and Systems*, 2020.

[9] K. Weiß and H. Worn, "The working principle of resistive tactile sensor cells," in *IEEE International Conference on Mechatronics and Automation, ICMA 2005*, 2005, pp. 471–476, ISBN: 0780390458. DOI: 10.1109/icma.2005.1626593.

[10] Y. Wang, X. Wu, D. Mei, L. Zhu, and J. Chen, "Flexible tactile sensor array for distributed tactile sensing and slip detection in robotic hand grasping," *Sensors & Actuators: A. Physical*, vol. 297, p. 111 512, 2019, ISSN: 0924-4247. DOI: 10.1016/j.sna.2019.07.036. [Online]. Available: https://doi.org/10.1016/j.sna.2019.07.036.

[11] C. Wang, L. Dong, D. Peng, and C. Pan, "Tactile Sensors for Advanced Intelligent Systems," *Advanced Intelligent Systems*, vol. 1, no. 8, p. 1 900 090, 2019, ISSN: 2640-4567. DOI: 10.1002/aisy.201900090.

[12] R. S. and M. Valle, "Tactile Sensing for Robotic Applications," *Sensors: Focus on Tactile Force and Stress Sensors*, no. December, 2008. DOI: 10.5772/6627.

[13] Q. Li, O. Kroemer, Z. Su, F. F. Veiga, M. Kaboli, and H. J. Ritter, "A Review of Tactile Information: Perception and Action Through Touch," *IEEE Transactions on Robotics*, pp. 1–16, 2020, ISSN: 19410468. DOI: 10.1109/TRO.2020.3003230.

[14] Z. Su, K. Hausman, Y. Chebotar, A. Molchanov, G. E. Loeb, G. S. Sukhatme, and S. Schaal, "Force estimation and slip detection/classification for grip control using a biomimetic tactile sensor," *IEEE-RAS International Conference on Humanoid Robots*, vol. 2015-Decem, pp. 297–303, 2015, ISSN: 21640580. DOI: 10.1109/HUMANOIDS.2015.7363558. [Online]. Available: https://syntouchinc.com/wp-content/uploads/2016/12/2015%7B%5C_%7DSu%7B%5C_%7DHumanoids-1.pdf.

[15] K. Flores De Jesus, M. H. Cheng, L. Jiang, and E. G. Bakhoum, “Resolution Enhancement Method Used for Force Sensing Resistor Array,” *Journal of Sensors*, vol. 2015, 2015, ISSN: 16877268. DOI: 10.1155/2015/647427.

[16] K. M. Lynch, F. C. Park, K. M. Lynch, and F. C. Park, *Modern Robotics: Mechanics, Planning, and Control [Bookshelf]*, 6. 2019, vol. 39, pp. 100–102, ISBN: 9781107156302. DOI: 10.1109/MCS.2019.2937265.

[17] T. Yoshikawa, “Multifingered robot hands: Control for grasping and manipulation,” *Annual Reviews in Control*, vol. 34, no. 2, pp. 199–208, 2010, ISSN: 13675788. DOI: 10.1016/j.arcontrol.2010.09.001. [Online]. Available: http://dx.doi.org/10.1016/j.arcontrol.2010.09.001.

[18] J. S. Kim, D. Y. Kwon, and B. D. Choi, “High-accuracy, compact scanning method and circuit for resistive sensor arrays,” *Sensors (Switzerland)*, vol. 16, no. 2, 2016, ISSN: 14248220. DOI: 10.3390/s16020155.

[19] J. A. Hidalgo-López, R. Fernández-Ramos, J. Romero-Sánchez, J. F. Martín-Canales, and F. J. Ríos-Gómez, “Improving Accuracy in the Readout of Resistive Sensor Arrays,” *Journal of Sensors*, vol. 2018, 2018, ISSN: 16877268. DOI: 10.1155/2018/9735741.

[20] J. A. Lopez, O. Oballe-Peinado, and J. A. Sanchez-Duran, “A Proposal to Eliminate the Impact of Crosstalk on Resistive Sensor Array Readouts,” *IEEE Sensors Journal*, vol. 20, no. 22, pp. 13 461–13 470, 2020, ISSN: 15581748. DOI: 10.1109/JSEN.2020.3005227.

[21] Y. Shiiki and H. Ishikuro, “Interface with Opamp output-impedance calibration technique for a large integrated 2-D resistive sensor array,” *Proceedings - IEEE International Symposium on Circuits and Systems*, vol. 2019-May, pp. 2–6, 2019, ISSN: 02714310. DOI: 10.1109/ISCAS.2019.8702594.

A Design Diagrams

A.1 Amplifier circuit diagram

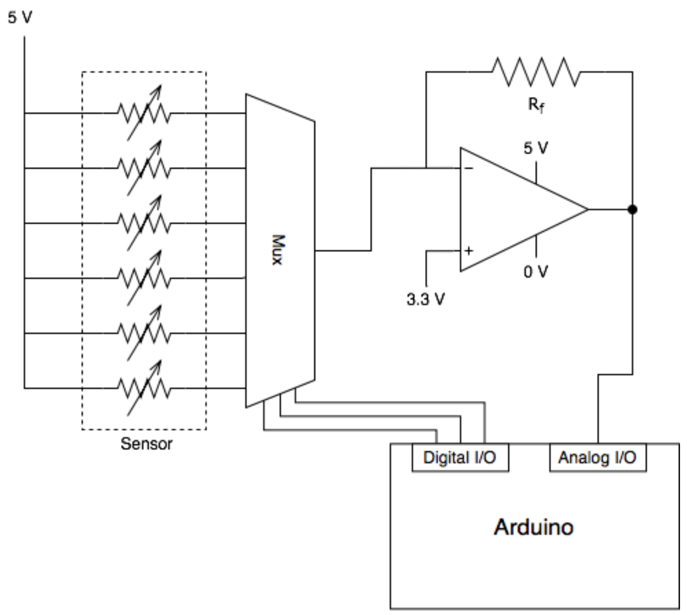
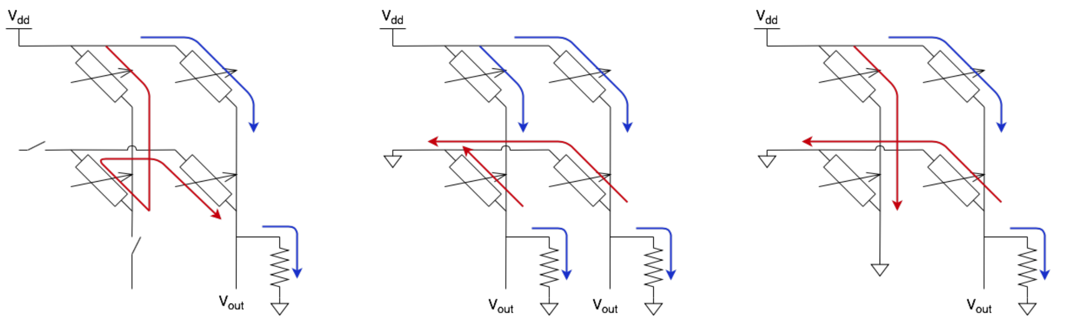


Figure A.1: Sampling circuit using an amplifier. The drive and ground voltages together with the $3.3V$ reference votlage are also provided by the Arduino board.

A.1.1 Crosstalk in Sensing Array

In order to limit the number of wires, resistive sensor having their taxels arranged in a grid, is normally connected using rows and columns. Crosstalk describes unwanted current flow between rows or columns that affect the output signal. Enforcing the current direction by adding diodes is impractical.

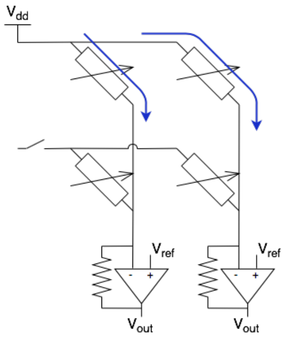
The effect of crosstalk increases with the size of the taxel array.



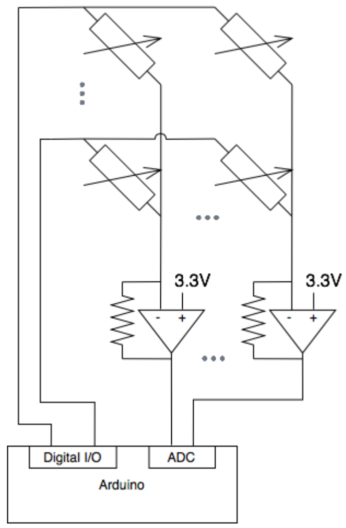
(a) HI on row and column (b) Grounding row (c) Grounding row and column

Figure A.2: Paths of crosstalk current in circuits using voltage dividers

Blue arrows indicate the desired current paths and red indicates crosstalk current.



(a) Array circuit using OpAmps



(b) Circuit connected to Arduino Board

Figure A.3: Zero-potential method for reducing crosstalk with the use of operational amplifiers.

The amplifiers drive the column to V_{ref}. The row may be practically disconnected be setting the I/O pin as high impedance.

B Additional measurements

B.1 Object properties

Object	Weight [g]	Size [cm]	Diameter [cm]	Height [cm]
Tomato	91	-	~ 5	~ 5
Mandarin	66	-	~ 5	~ 4
Box	117	5.0×5.0	-	5.0
Weight	511	-	4.9	2.8
Sponge	4	6.0×9.0	-	2.8
Glass	171	-	6.7	11.6

Table B.1: Properties of test objects

B.2 Grasping forces

Table B.2 shows the measured force at one finger when grasping a set of test objects as seen in figure 4.3. The measurements are carried out by holding the objects between the sensors described in section 3.2 while using the end-effector force estimation of the robot. *Grasp force* is recorded as the force applied when holding the object naturally. *Slip force* is the recorded force when the object starts slipping and *Damage force* indicates when the grip deforms the object.

Object	Weight [g]	Grasp force [N]	Slip force [N]	Damage force [N]
Tomato	91	$1.0 - 1.5$	~ 0.5	~ 3.0
Mandarin	66	$1.0 - 3.0$	~ 0.5	~ 5.0
Box	117	$1.0 - 2.0$	~ 0.5	-
Weight	511	$8.0 - 15$	~ 7.0	-
Sponge	4	$0.4 - 1.0$	~ 0.2	-
Glass	171	$1.5 - 4.0$	~ 0.7	-

Table B.2: Grasping forces of test objects

B.3 Grasping Indentation

The following measurements provide an initial assessment of the limits to which the gripper should close around the objects. The object is first placed between the finger of the gripper such that the finger are just making contact with the surface of the object. The values in the table indicate the decrement in gripper width from this position. 'Lifting indentation' describes the minimal decrement needed such that the object may be lifted without slipping, 'Damage indentation' is the amount needed to cause a visible or noticeable dent in the object after release and 'Max indentation' is the measured indentation when the gripper is commanded to close the gripper using its minimal possible grasping force without tactile feedback, that is $20N$.

Object	Lifting indentation [mm]	Damage indentation [mm]	Max indentation [mm]
Tomato	1.0	3.0	6.0
Mandarin	2.0	3.5	10.5

Table B.3: Grasping identations of test objects

B.4 Tuning Feedback Resistance for Amplifier Circuit

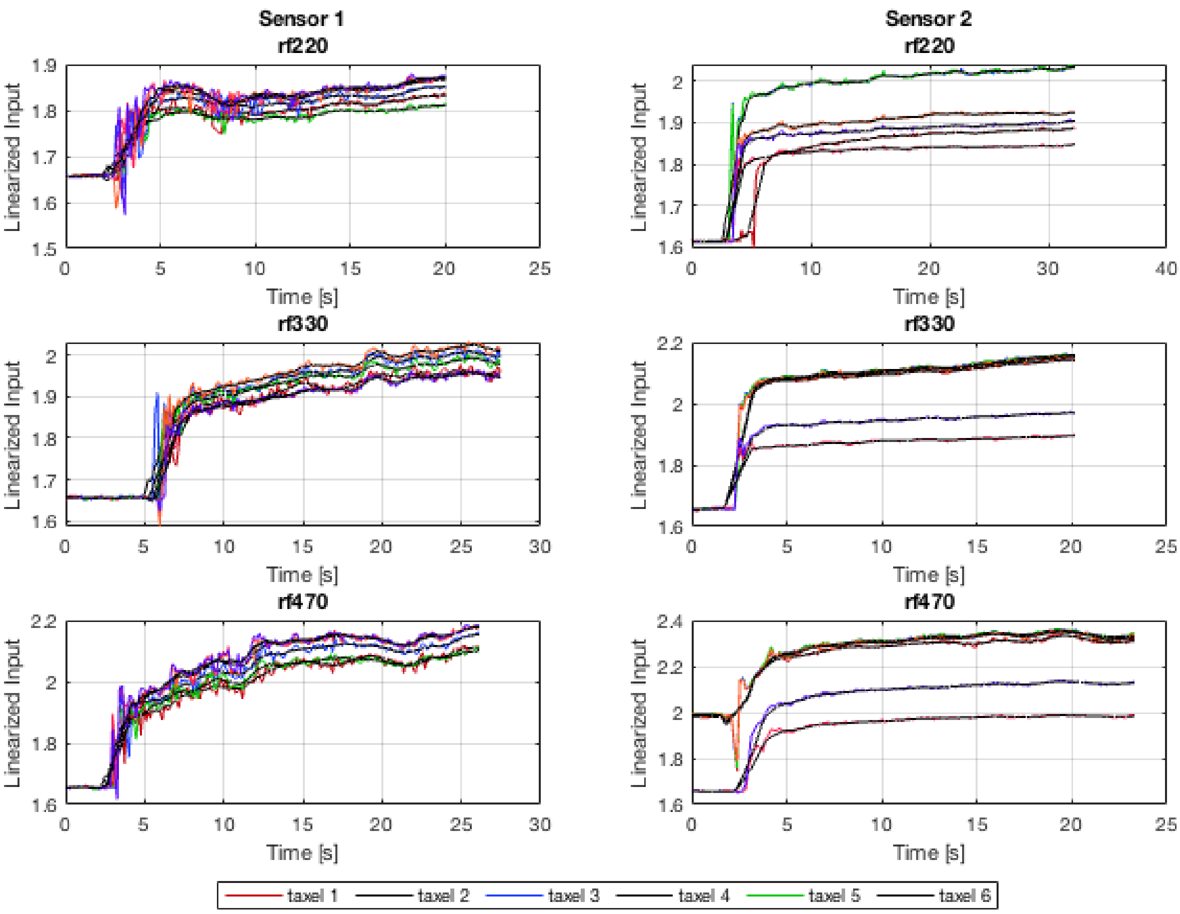


Figure B.1: Feedback testing for amplifier circuit

B.5 Tuning Reference Resistance for Voltage Divider Circuit

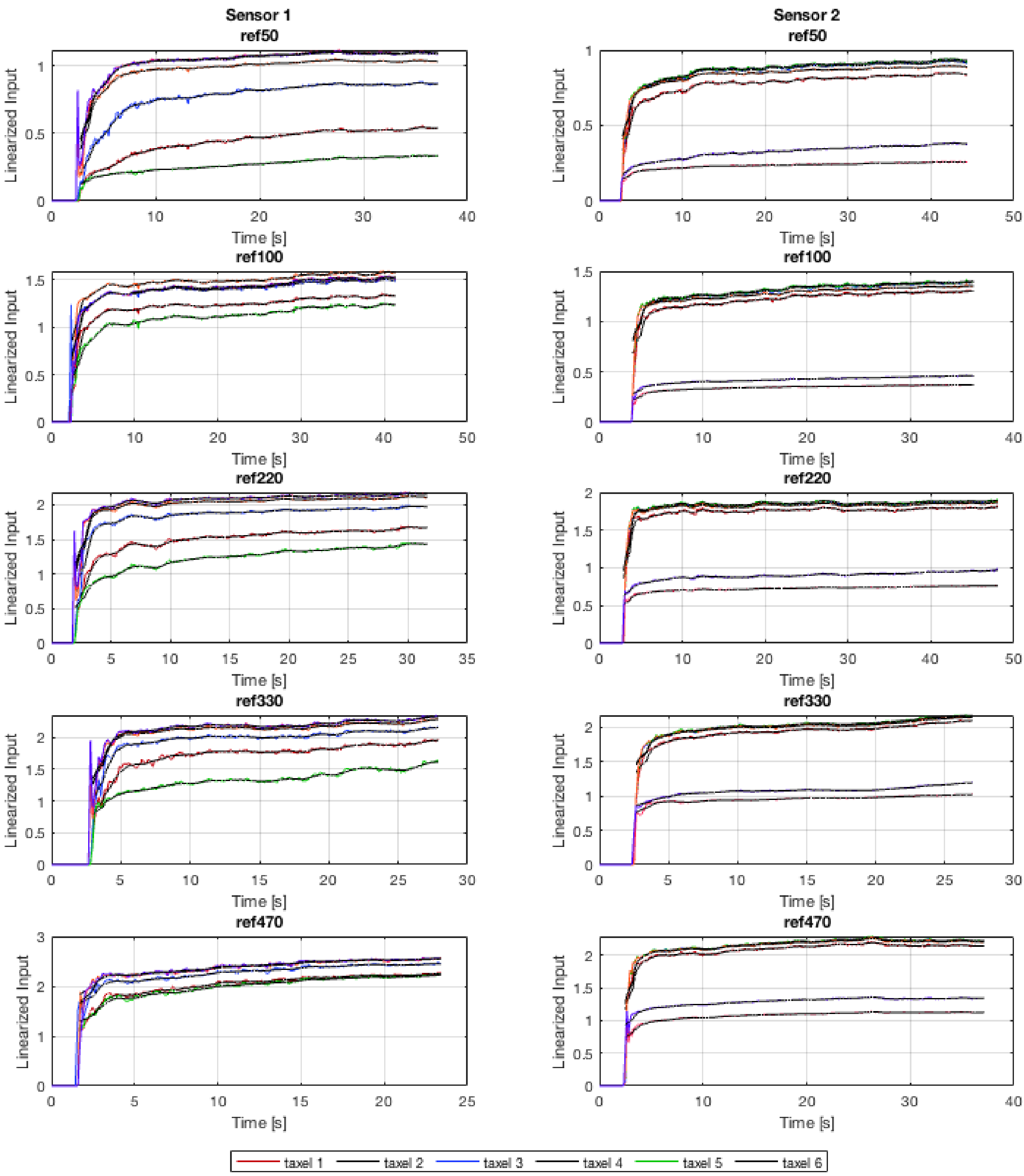


Figure B.2: Reference testing for voltage divider circuit

B.6 Calibration using Amplifier Circuit

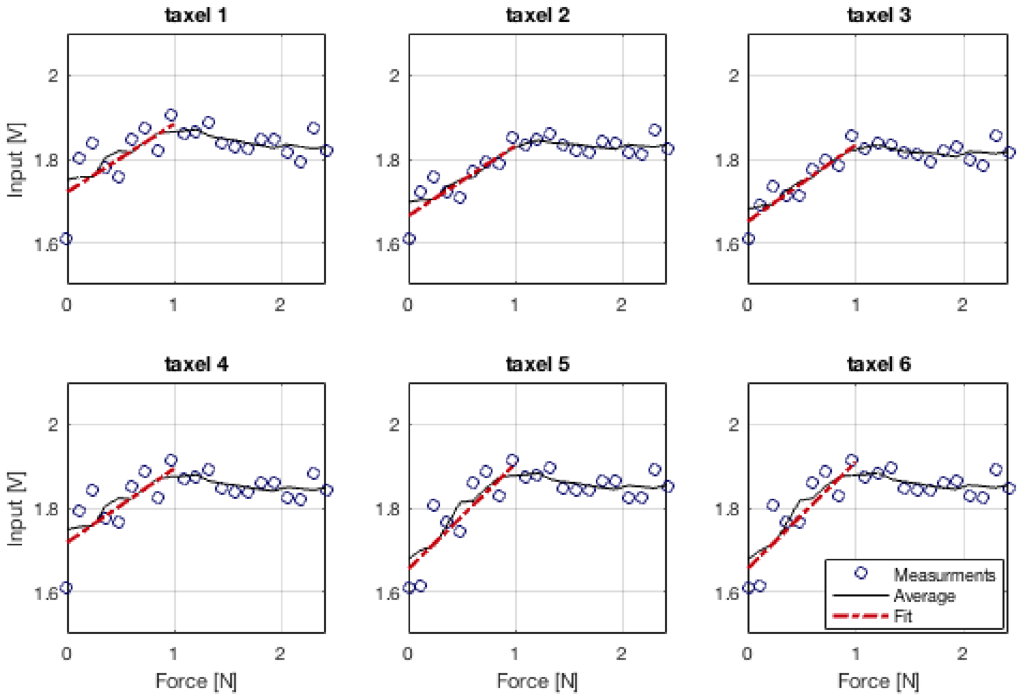


Figure B.3: sensor 1

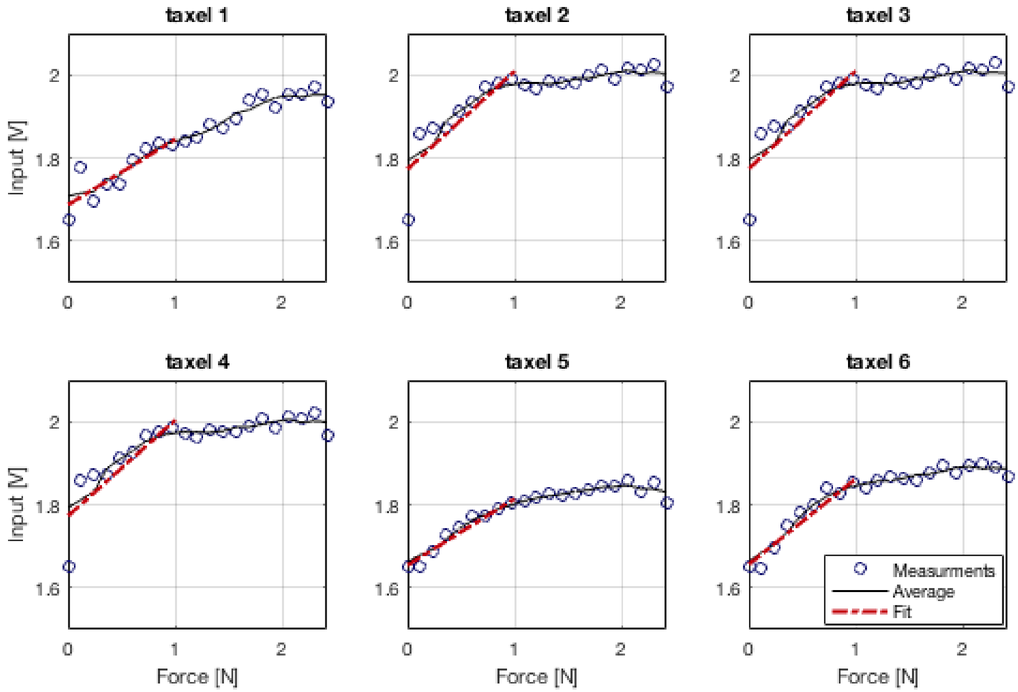


Figure B.4: Sensor 2

B.7 Gripper Control on Test Objects

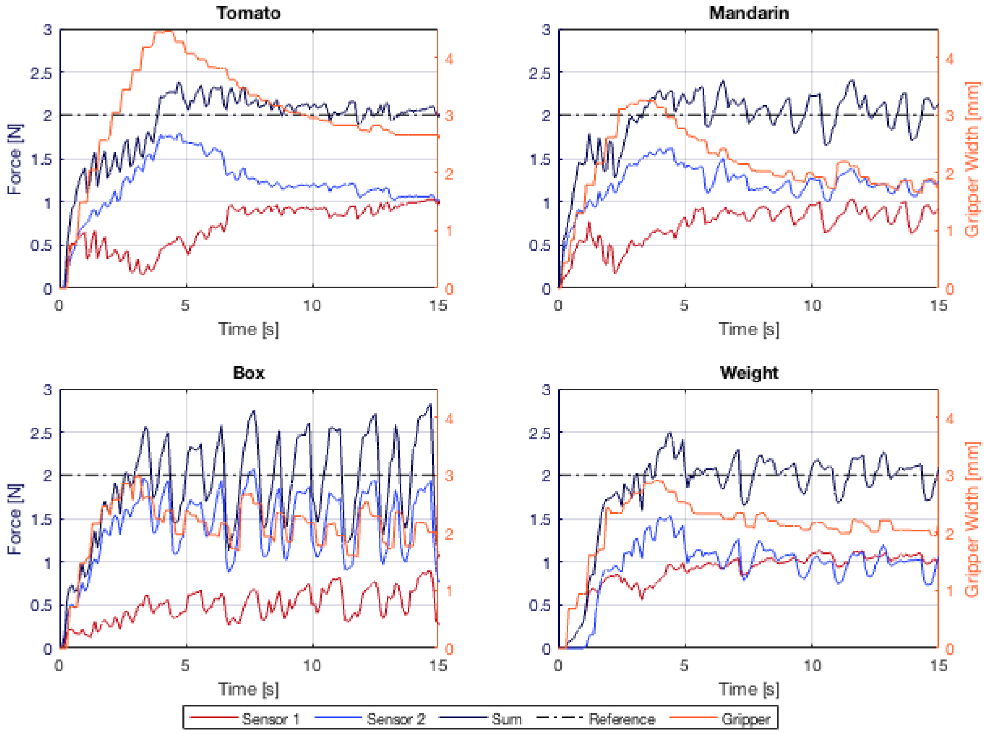


Figure B.5: Gripper control performed on various objects with a $2N$ reference force

B.8 Force Response when Gripping

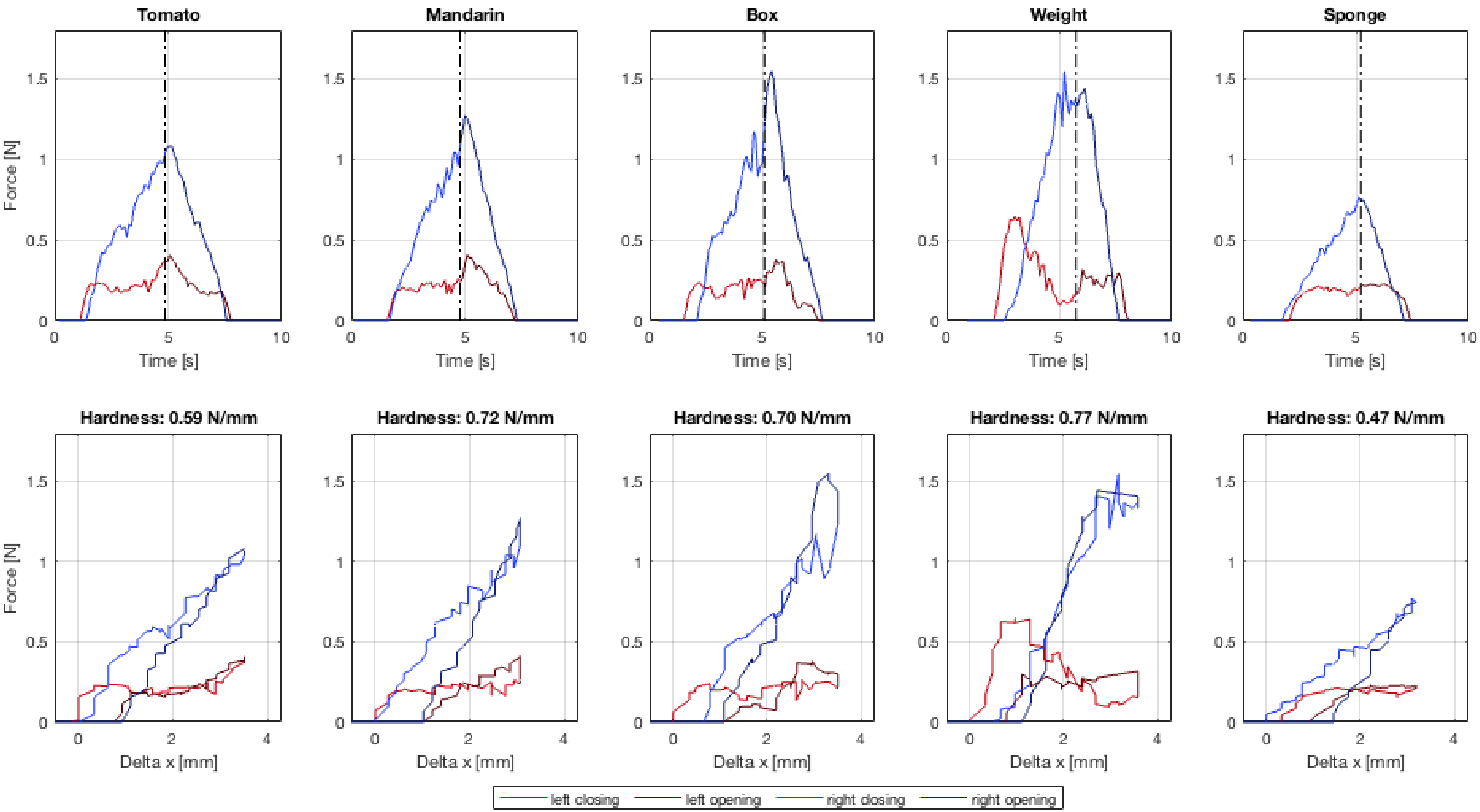


Figure B.6: Force response when closing gripper around various objects

The hardness is measured when the gripper is maximally closed around the object using *sensor 1* only. This corresponds to approximated slope of the blue hysteresis.

B.9 Crosstalk Measurements

Based on a prototype sensor with 2-by-3 taxel layout, the presence of crosstalk was tested for two circuits by sustaining one taxel pressed and then touching another. The effect of crosstalk can be seen by how the response of the sustained taxel drops.

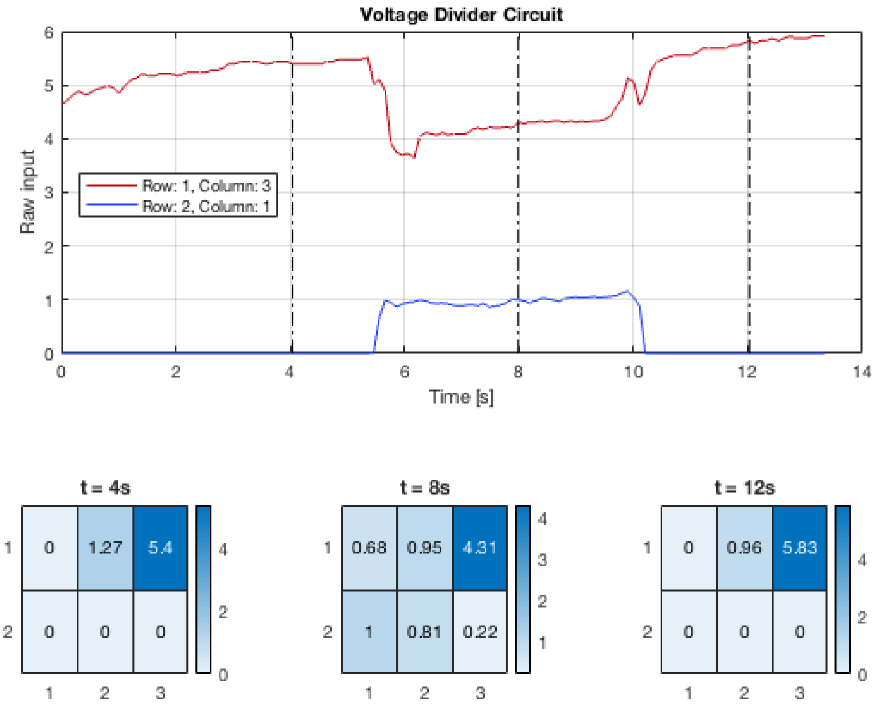


Figure B.7: Using the voltage divider circuit from figure A.2a

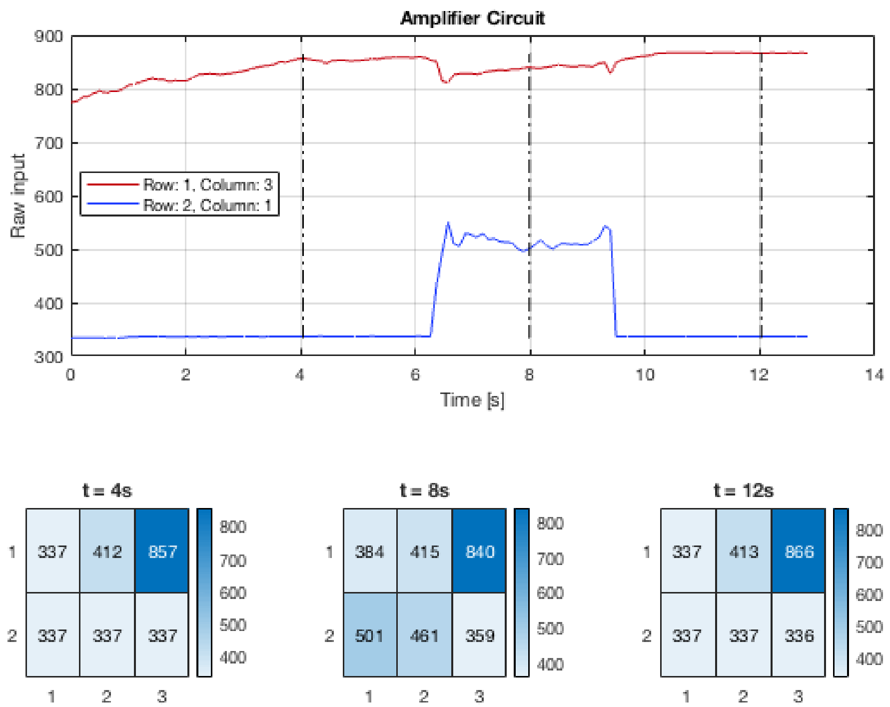


Figure B.8: Using the amplifier circuit seen in figure **??**

Technical
University of
Denmark

Elektrovej, Building 326
2800 Kgs. Lyngby
Tlf. 4525 1700

www.elektro.dtu.dk