SECONDA UNIVERSITÀ DEGLI STUDI DI NAPOLI

DIPARTIMENTO DI INGEGNERIA INDUSTRIALE E DELL'INFORMAZIONE

DOTTORATO IN INGEGNERIA ELETTRONICA ED INFORMATICA
XXVIII CICLO

Thesis

A Distributed Force/Tactile Sensor for Physical Human Robot Interaction

S.S.D. ING-INF/04

Author
Ing. Andrea Cirillo
Matr. D124/017

Coordinator
Prof. Giovanni Leone

Advisor
Ing. PhD Salvatore Pirozzi

Academic Year 2014-2015

Thesis supervisor:

Ing. PhD Salvatore Pirozzi
Dipartimento di Ingegneria Industriale e dell'Informazione
Seconda Università degli Studi di Napoli

This work has been carried out by Andrea Cirillo during his PhD course in Robotics, under the supervision of Ing. PhD Salvatore Pirozzi and of Prof. Ciro Natale at the Dipartimento di Ingegneria Industriale e dell'Informazione, Seconda Università degli Studi di Napoli, Aversa, Italy. His PhD has been supported within the European Project SAPHARI (Safe and Autonomous Physical Human-Aware Robot Interaction, grant agreement no. 287513).

A Distributed Force/Tactile Sensor for Physical Human Robot Interaction
by Andrea Cirillo

Non temete i momenti difficili,
il meglio viene da lì.

Rita Levi Montalcini

A tutti coloro che hanno lottato affinchè fossi ancora qui oggi...
alla mia famiglia, a mio fratello, a mia mamma...
a chi oggi, mi sostiene, mi comprende,
mi entusiasma, mi riempie
e mi regala un sorriso ogni giorno...

Perchè questi tre anni non sono legati
solo ad una mia maturazione professionale,
ma soprattutto ad una crescita personale...

Perchè questa tesi non la sento solo mia...
ma è di tutti voi...

E non importa quante volte cadrò,
troverò sempre la forza di rialzarmi...

Sommario

Oggi, l'interazione fisica tra robot ed esseri umani rappresenta una sfida particolarmente interessante per i ricercatori che lavorano nel campo della robotica. L'ultima tendenza nella robotica avanzata è quella di sviluppare una nuova generazione di robot, come umanoidi, robot volanti, per l'assistenza e robot medici con un alto livello di autonomia, che siano in grado di cooperare e di interagire tra di loro, così come con gli esseri umani secondo il concetto della physical Human-Robot Interaction. Attività autonome o cooperanti richiedono che i robot debbano operare senza produrre danni a se stessi, agli esseri umani e ad altri oggetti circostanti, interagendo con l'ambiente in modo sicuro. L'adozione di un sistema sensoriale diventa fondamentale ed il senso del tatto è particolarmente importante in quanto molte operazioni richiedono che il robot riconosca le collisioni involontarie, ma anche che entri intenzionalmente a contatto con oggetti o persone. L'interazione fisica sicura tra uomo e robot richiede la conoscenza delle forze di interazione e dei punti di contatto, al fine di svolgere compiti di cooperazione e co-manipolazione e limitare i danni causati da urti accidentali. Questa informazione fondamentale può essere ottenuta attraverso misure dirette o mediante tecniche di stima che fanno uso di metodi alternativi a seconda della tecnologia disponibile.

Questa tesi presenta un sensore di forza/tatto distribuito descrivendo nel dettaglio la sua progettazione e realizzazione, e verificando sperimentalmente le sue capacità in diverse applicazioni. Il sensore è composto da moduli sensibili indipendenti, in grado di stimare la componente normale e le componenti tangenziali della forza di contatto. Ciascun modulo sensibile è costituito da quattro coppie di LED e di fototransitor ad infrarossi, coperte da uno strato di silicone che trasduce la forza esterna in una deformazione meccanica. Il vettore di forza applicato sulla superficie deformabile è stimato come un'opportuna combinazione dei quattro segnali di tensione misurati dai quattro fototransistor. In questo lavoro è stato realizzato, caratterizzato e testato un prototipo completo formato da una matrice 6×6 di moduli sensibili. Il prototipo è in grado di discriminare più aree di contatto e di stimare le forze risultanti per ciascuna di esse. Il sensore, dapprima sviluppato in tecnologia rigida, è stato riprogettato e prodotto in tecnologia flessibile per garantire la conformabilità meccanica e l'adattabilità alle superfici curve, come ad esempio quelle dei manipolatori robotici. Vengono, inoltre, fornite delle linee guida per l'installazione del sensore su un sistema robotico generico. Il sensore è stato installato con successo su alcuni

manipolatori ridondanti sviluppati da aziende diverse, ad esempio KUKA e YASKAWA, e, attraverso la definizione di architetture HW/SW e lo sviluppo di software driver adeguati, esso è stato utilizzato in applicazioni di interazione fisica tra uomo e robot, in cui le forze di contatto possono verificarsi su aree distribuite. Inoltre, sono stati progettati ed implementati tre algoritmi che permettono di utilizzare il sensore come interfaccia Uomo-Macchina e, quindi, di impartire dei comandi alla piattaforma robotica tramite dei gesti tattili tracciati sulla superficie del sensore. Adottando la formulazione classica del controllo di ammettenza, il sensore è stato utilizzato per task di manual guidance, intuitive programming, collision detection e reaction. Come mostrano i risultati sperimentali, le tecniche basate sull'adozione di un sensore distribuito dedicato alla misura delle forze di contatto tendono ad essere preferibili ai metodi alternativi basati sull'uso di modelli dinamici dei robot e sull'uso di misure di coppia ai giunti.

Abstract

Today, physical interaction between robots and humans represents an interesting challenge for robotic researchers. The latest trend in advanced robotics is to develop a new generation of robots such as humanoids, flying robots, assistant and medical robots with a high level of autonomy, which are able to cooperate and interact each other as well as with humans according to the physical Human-Robot Interaction concept. Autonomous or cooperative tasks require that the robots should operate without damage themselves, humans and other surrounding objects, interacting safely with the environment. Sensing becomes fundamental, and tactile sensing is particularly important since many tasks require the robot to recognise unintentional collisions or to have intentional physical contact with objects or humans. Safe and efficient human-robot physical interaction requires the knowledge of interaction forces and contact locations in order to perform cooperation and co-manipulation tasks and to limit damage from accidental impacts. This crucial information can be obtained through direct measurements or by estimation techniques, by using different methods depending on the available technology.

This thesis presents a distributed force/tactile sensor by describing its design and development, and by verifying experimentally its capabilities in various applications. The sensor is constituted by independent sensing modules, able to estimate both normal and shear contact force components. Each sensing module consists of four couples, constituted by an infrared Light Emitting Diodes and a Photo-Detectors, covered by a silicone layer that transduces the external force in a mechanical deformation. The applied force vector is estimated as a suitable combination of the four voltage signals measured by the four receivers. A complete prototype, with a 6×6 matrix of sensing modules, has been realized, characterized and tested. The prototype is able to discriminate multiple contact areas and to estimate the force resultants for each contact area. The sensor, firstly developed in rigid PCB technology, has been re-designed and manufactured in flex PCB technology in order to guarantee mechanical compliant and conformability to curved surfaces, such as robot arms. Guidelines for the installation of the sensor on a generic robotic system are provided. The sensor has been successfully installed on few redundant manipulators of different brands, KUKA and YASKAWA, and, through the definition of proper system architectures and sensor drivers, it has been exploited in applications of safe physical Human-Robot Interaction, where contact forces over large distributed areas can occur. Moreover, three

algorithms that allow the use of the sensor as Human-Machine Interface and, then, the recognition of the touch gestures traced on the sensor surface have been designed and implemented. By adopting the classic formulation of the admittance control, the sensor has been used in manual guidance, intuitive programming, collision avoidance and reaction tasks. As shown with the experimental results, the use of a dedicated distributed sensor for measuring the contact force vectors is aimed at overcoming current methods based on the use of robot dynamic models and joint torque measurements.

Contents

List of Figures

List of Tables

CHAPTER 1

Human-Robot Interaction

In the last two decades, new industrial and humanoid robots are designed following the physical Human-Robot Interaction (pHRI) concept, to allow the robot to share the same workspace and cooperate with humans in applications such as assisted industrial manipulation, collaborative assembly, domestic work, entertainment, rehabilitation or medical applications. Sensors, able to detect external contacts, that occur between the robots and the humans and/or the environment, are developed and used to provide the robots with cognitive abilities and to develop new communication techniques used in different fields, e.g. for the knowledge transfer from human to robot, for the realization of human-friendly interfaces for robot programming and control, for human guidance in the completion of the robot task, for the detection of the human presence to avoid unintended collisions (safe operation around humans) [1]. In particular, safety issues become of primary concern when robots and humans share the same environment. The robot should have the ability to avoid unintended collisions by anticipating dangerous situations, and it should react promptly recovering a safe working condition.

1.1 The SAPHARI project

Recent progress in pHRI research showed in principle that human and robots can actively and safely share a common workspace. The fundamental breakthrough that enabled these results was the human-centered design of robot mechanics and control. This made it possible to limit potential injuries due to unintentional contacts. Previous projects, in particular the PHRIENDS project in which a part of the consortium has been involved, provided remarkable results in these directions, constituting the background foundation for this proposal. Inspired by these results, SAPHARI[1] (Safe

[1]Please, refer to the project website for more details: http://www.saphari.eu.

Figure 1.1: SAPHARI project logo.

and Autonomous Physical Human-Aware Robot Interaction) performed a fundamental paradigm shift in robot development in the sense that the human is placed at the centre of the entire design. The project took a big step further along the human-centered roadmap by addressing all essential aspects of safe, intuitive physical interaction between humans and complex, human-like robotic systems in a strongly interconnected manner. While encompassing safety issues based on biomechanical analysis, human-friendly hardware design, and interaction control strategies, the project developed and validated key perceptive and cognitive components that enable robots to track, understand and predict human motions in a weakly structured dynamic environment in real-time. Robots have been equipped with the capabilities to react to human actions or even take the initiative to interact in a situation-dependent manner relying on sensor based decisions and background knowledge.

Apart from developing the necessary capabilities for interactive autonomy, the human safety has been tightly incorporate also at the cognitive level. This enabled the robots to react or physically interact with humans in a safe and autonomous way. Keeping in mind the paradigm to "design for safety and control for performance", research developments have been pursued in several areas, starting with the fundamental injury mechanisms of humans cooperating with robots. The analysis have been first carried out for stiff robots and then extended to variable stiffness actuation systems in terms of safety, energy, and load sustainability. Biomechanical knowledge and biologically motivated variable compliance actuators have been used to design bi-manual manipulation systems that have design characteristics and performance properties close to humans. Real-time task and motion planning of such complex systems required new concepts including tight coupling of control and planning that lead to new reactive action generation behaviours. Safe operation has been enforced in mobile manipulation scenarios with large workspaces by smart fusion of proprioceptive and exteroceptive sensory information, sensor-based task planning, human gestures and motion recognition and learning, and task-oriented programming, including configuration and programming of safety measures. Finally, self explaining interaction and communication frameworks have been developed to enhance the system usability and make the multimodal communication between human and robot seamless.

The project involved important partners engaged in the European robotics research, such as Università di Roma La Sapienza (UNIROMA1), Airbus Group (AIRBUS), Institute of Robotics and Mechatronics (DLR), Fondazione Istituto Italiano di Tecnologia (IIT), Fraunhofer-Institute of Optronics, System Technologies and Image Exploitation (IOSB), KUKA Laboratories GmbH (KUKA), Centre National de la Recherche Scientifique (LAAS-CNRS), Technische Universität München (TUM), Università di Napoli Federico II (UNINA), Università di Pisa (UNIPI), Universität Bremen (UNIHB).

1.2 Toward Human-Robot coexistence

Safety in terms of industrial robots usually consists of isolating the manipulator workspace from the one of humans by a safety guard with locked safety doors or light barriers [2]. Once the safety door is opened or the light barrier is crossed, the robot immediately stops its task. An increasing interest has recently been observed in domestic and industrial service robots, characterized by desirable, and under certain circumstances even unavoidable physical interaction [3]. Therefore, a resulting essential requirement is to guarantee safety for human users in regular operation mode as well as in possible fault modes of the robotic system. The latest trend in advanced robotics is to develop a new generation of robots such as humanoids, flying robots, assistant and medical robots with a high level of autonomy, that are able to cooperate and interact each other as well as with humans. Autonomous or cooperative tasks require that the robots should operate without producing damages to themselves, to humans and to other surrounding objects, interacting safely with the environment. Sensing becomes fundamental, and tactile sensing is particularly important since many tasks require the robot to recognize unintentional collisions or to make intentional physical contact with objects or humans.

For humans the skin is a remarkable organ and the sense of touch is fundamental for manipulation tasks execution. It consists of an integrated, stretchable network of sensors that provide information about tactile and thermal stimuli to the brain, allowing us to operate within our environment safely and effectively. The development of electronic networks comprised of flexible, stretchable, and robust devices that are compatible with large area implementation and integrated with multiple functionalities is a testament to the progress in developing an electronic skin. The effort to create an artificial skin with human-like capabilities is motivated by the possibility of such large, multi-sensory surfaces being highly applicable for autonomous artificial intelligence (e.g., robots), medical diagnostics, and replacement prosthetic devices capable of providing the same, if not better, level of sensory perception than the human skin. Enhancing robots capabilities with the tactile sense could extends their range of applications to include highly interactive tasks, such as caring for the elderly. An artificial skin with such sensory capabilities is often referred to in the literature as sensitive skin, smart skin, or electronic skin [4].

The idea of the artificial skin has been used for the first time several years ago in science fiction and movies, e.g, *Six Million Dollar Man* series, *Star Wars* series and *Terminator* movie series. In the 1980s, HP (Hewlett-Packard) marketed a personal computer that was equipped with a touchscreen, allowing users to activate functions by simply touching the display. It was the first mass-marketed electronic device capitalizing on the intuitive nature of human touch. In 1985, GE (General Electric) built the first sensitive skin for a robotic arm using discrete infrared sensors placed on a flexible sheet at a resolution of $\approx 5\,\text{cm}$ [5]. The latter was proximally aware of its surroundings, allowing the robot arm to detect potential obstacles and effectively moves within its physical environment. In the 1990s, scientists began using flexible materials to create large-area, low-cost and printable sensor patches. Jiang et al. proposed one of the first flexible tactile sensor for shear forces by creating silicon (Si) micro-electromechanical (MEM) islands by etching thin Si wafers and integrating them on flexible polyimide foils [6]. At the same time, flexible arrays fabricated from organic semiconductors began to emerge [7], having a look at innovative solutions for enhancing the reliability of large sensor to mechanical bending [8].

Significant progress in the development of artificial skin has been achieved in recent years, in which particular emphasis has been placed on mimicking the mechanically compliant property of human skin. Suo et al. have developed stretchable electrodes [9, 10, 11], and Rogers et al. have transformed a typically brittle material, Si, into flexible electronics by using ultrathin (100 nm) films connected by stretchable interconnections [12, 13]. Someya et al. have fabricated flexible pentacene-based organic field-effect transistors (OFETs) for large area integrated pressure-sensitive sheets with active matrix readout [14, 15, 16], while Bauer et al. have investigated novel pressure sensing methods using foam dielectrics [17] and ferroelectrets [18] integrated with FETs. Other groups have developed stretchable optoelectronics, including light-emitting diodes (LEDs) [19] and organic photovoltaics (OPVs) [20] for integration with artificial skin. In [4] a significant pictures representing a timeline of the major milestones towards the development of artificial skin is reported (see Fig. 1.2). From the previous analysis on the artificial skin evolution it is easy to understand that stretchability and flexibility are two important characteristics to consider in a sensor design. However, while arrays of flexible electronics have been developed by using very thin plastic substrates, stretchable devices have been more difficult to achieve, and new processes and materials are often required.

A further description and a more deep analysis of the solutions presented in literature can be found in Sec. 1.4.

1.2.1 The tactile sensors in robotics

Robotic technologies have found enormous advantages in improving efficiency and reducing the cost of repetitive, well-defined manufacturing tasks. Recently, there are a great interest in designing robots that can work in less-structured environments by collecting information about their surroundings to make appropriate responses [21]. Such capabilities would allow them to work in close quarters with humans and complete more complicated and dynamic tasks (e.g., providing basic services to elderly people or undertaking dangerous rescue missions) [22]. However, highly functional tactile sensing will be required to improve the safety and effectiveness of current robotic technology [23]. In addition to robotic applications tactile sensing arrays could transform the medical field. Tactile sensors integrated into prosthetics could allow amputees to regain considerable functionality, and touch sensitive sensor skins could be useful in augmenting surgical gloves [24] or for measuring the health of patients [25]. Many essential aspects of life are mediated by the multifunctional tactile sensing capabilities of skin, including:

- normal force sensing for grasp control, object manipulation, and orientation determination
- tensile strain monitoring for proprioception (essential for simple movements such as standing or walking)
- shear force sensing for grasp control and friction determination
- vibration detection for slip detection and texture determination

The characteristics outlined above should be considered as the minimum requirements for an artificial skin. In the 1990's, one of the first survey of robotics developers outlined desired parameters for tactile skins [22]. In most cases, these requirements agree well with the capabilities of human skin, and both are summarized

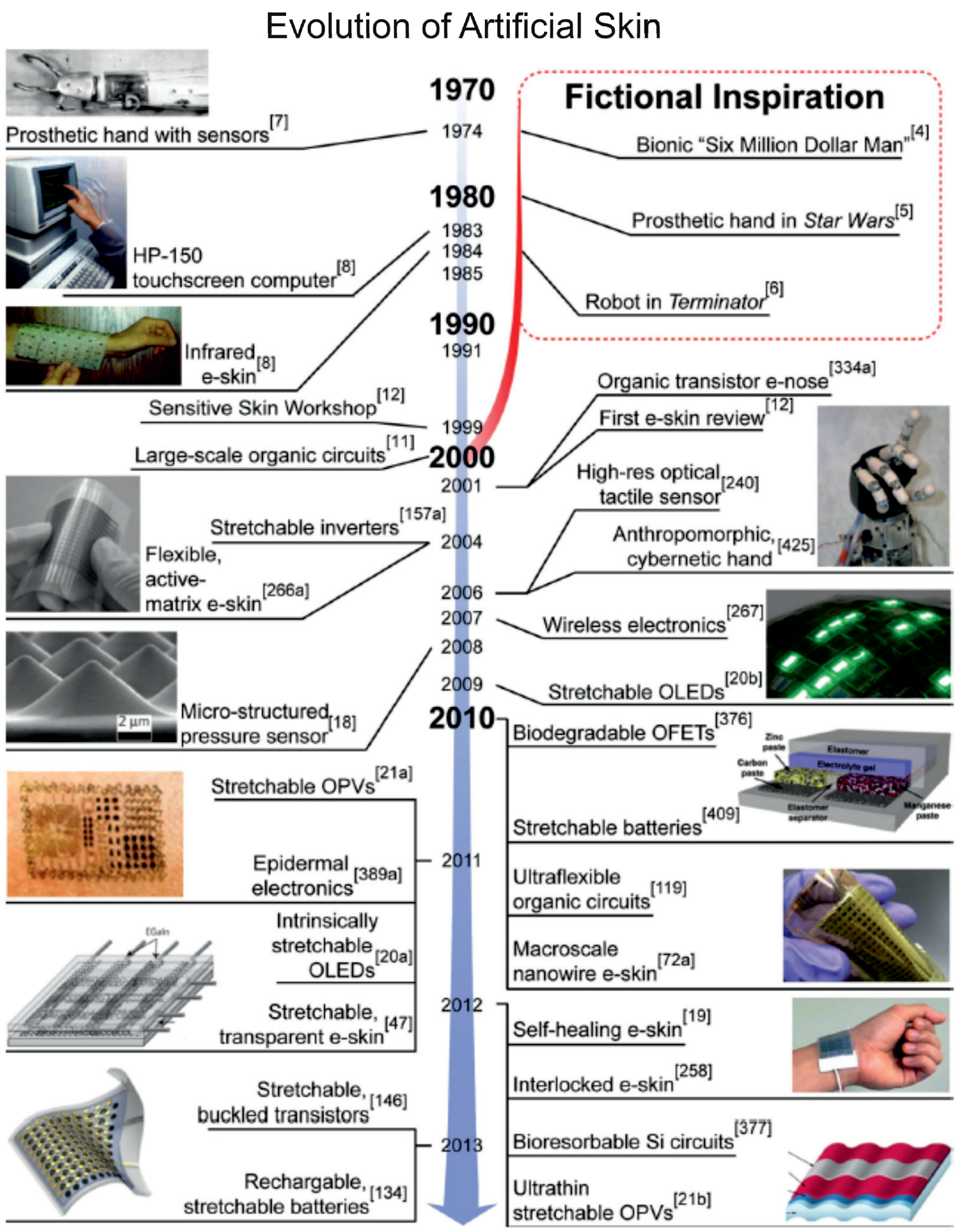


Figure 1.2: The picture has been published by Mallory et al. and it reports a brief chronology of the evolution of the artificial skin.

Table 1.1: Summary of the properties of human skin and corresponding requirements for artificial skin.

Parameter	Human skin	Requirement for artificial skin
Spatial resolution	1 mm	1 – 2 mm
Temporal resolution	20 – 40 ms	1 – 10 ms
Working range	> 10 kPa	1 – 1000 g
Hysteresis	High	Low

in Table 1.1. However, artificial skin with enhanced capabilities, such as improved sensitivity, higher sensing elements density and faster response times, could endow robots and prosthetics with capabilities that surpass those of our own skin. Tactile sensing has attracted significant interest, and development in the area has been rapid.

In order to measure the magnitude of a tactile stimulus, it must be converted into an electrical signal. Methods for accomplishing this conversion are described below.

Piezoresistivity. Piezoresistive sensors transduce a change in the resistance of a device into a measurement of strain and have been investigated extensively because of their simple structure and readout mechanism. The change in resistance can be derived from several factors, including changes in: the geometry of the sensing element, the resistivity of a semiconductor, the contact resistance between two materials and the resistivity of a composite.

Capacitance. Capacitive sensors for tactile sensing have demonstrated high strain sensitivity, compatibility with static force measurement, and low power consumption. However, since the capacitance is proportional to the contact surface, reducing the taxel size of these devices for miniaturization reduces the capacitance and the signal-to-noise ratio. Furthermore, capacitive sensors are susceptible to interference from external sources.

Piezoelectricity. Piezoelectricity refers to the ability of a material to generate a voltage in response to an applied force. The force causes a change in the length and separation between dipoles in the material, leading to the build-up of compensating charges on the electrodes. The ability of the material to convert normal forces into electrical charges is quantified using a piezoelectric strain constant. Given the high sensitivity of piezoelectric sensors to dynamic pressure and their fast response speed, they are often used to measure the vibrations associated with slip. However, piezoelectric materials suffer of drift in sensor response over time and have unreliable static sensing properties.

Optics. Optical sensors convert a tactile input into an electrical output using light as an intermediate. These sensors consist of a light source, a transmission medium, and a detector. The modulation of light intensity through force-sensitive waveguides, flexible optical fibers or reflecting surfaces has been used to fabricate optical sensors.

In many cases, tactile sensor elements are encapsulated within an elastomer to protect them from mechanical stress and environmental exposure, as well as also to provide conformal contact with stimuli.

Our senses of vision and hearing are facilitated by a small number of localized sensors. However, tactile sensing requires a multitude of sensors distributed over a large area and developing methods to collect and process such a large amount of information has been a persistent challenge in this specific field. The simplest strategy for device readout is direct addressing, in which each device is contacted by a separate connection. Good temporal resolution can be achieved with this method, but large arrays quickly lead to an unmanageable number of connections.

1.3 The modern approaches

Safe and efficient human-robot physical interaction requires the knowledge of interaction forces and contact locations in order to perform cooperation and co-manipulation tasks and to limit damage from accidental impacts. This crucial information can be obtained through direct measurements or by estimation techniques using different methods depending on the available technology. As said in the previous sections, in the last decades, different kinds of artificial skins have been developed. Most of them are based on tactile sensing and are able to measure the contact point where a force is applied. Some of them have been developed within EU Projects whose goals were improve the knowledge of the robotics community in cognition, human-robot interaction and coexistence through the implementation of humanoid robots. For example, in [26] the authors presents a tactile skin based on a set of distributed capacitive tactile sensing elements that have been integrated on the iCub cognitive robot covering the limbs and providing a tactile feedback, in terms of contact points, for possible contacts with the environment. Information such as force magnitude and direction is not easily reconstructed. In fact, as reported in [27, 28, 29], the estimation of the contact forces and the control of the interaction forces exerted between the iCub robot and the environment need for additional sensors installed in the robot limbs, i.e., six axes force/torque sensors, as well as knowledge of the dynamic model of the robot. It is clear that the prevailing trend of the research in this field is covering the whole body of the robot or some of its parts with an array or patches of force/tactile sensors.

On the other hand, indirect estimation of the interaction forces can be obtained using alternative techniques mainly based on residual estimation methods or on the use of exteroceptive sensors, i.e., depth cameras [30]. Haddadin et al. [31, 32] propose a collision detection mechanism based on the estimation of the collision torques. The algorithm requires the computation of the dynamic model of the robot that is used to detect the disturbance torques through the use of a model-based torque observer.

One of the most effective approaches proposed in the literature is that based on the residual method, which allows to estimate the joint torques generated by the external forces applied to the body of a robot manipulator [31, 32, 33]. This information, together with the measurement of the contact location, that can be obtained, e.g., with tactile sensing or depth cameras, allows to compute a good estimation of the external force, also in the case of multiple contacts [34]. The residual technique has been successfully employed also in [35, 36, 37] for applications where force feedback is required to control the intentional physical human-robot interaction. The nice feature of the residual method is that it does not require the installation of force sensors on the robot, although exteroceptive sensing is needed for the identification of the contact locations. A drawback is that, as all the model-based techniques, an accurate knowledge of the robot dynamic model is required; some dynamic parameters, like the robot payload, and some torque disturbances, like the joint friction, are uncertain and may change during robot operation. Another disadvantage with respect to the solutions based on distributed sensors is that the forces and moments that do not produce joint torques (i.e., those that are balanced by the mechanical structure) cannot be measured. Finally, the accuracy of the estimation depends on the location of the contact point on the robot structure.

On the basis of the previous discussion, existing methods to measure contact forces in multiple contact points can be classified in two classes. In the first one the contact force estimation is carried out through a residual-based method using the dynamic model of the robot, provided that the environment is properly sensorized with sensors external to the robotic system, i.e., camera and depth sensors such as Microsoft Kinect. The second one does not use sensors external to the robotic system, but it introduces a tactile sensor to determine the application point of the external force, while the estimation of the applied force is based on the computation of the robot dynamic model whose parameters are assumed perfectly known. Moreover, the method requires six-axis force sensors located along the robot mechanical structure.

The design and the development of a distributed force/tactile sensor able to provide both a measure of the applied force and the geometric information on the contact point (without requiring the use of other sensors and the knowledge of dynamic parameters of the robot) results to be an ambitious, effective and valid challenge aimed at improving the cooperation and collaboration between robots and humans.

1.4 A look at distributed sensors: tactile sensing for robots

In the last decades, the design and the use of embedded and distributed sensor systems that extend the perception of the robot of the surrounding environment has been enforced. Nowadays, there are no commercially available touch sensitive covers for robotic systems. A possible reason may be the complex shape of the covering structure of modern robotic systems. Humanoid robotic systems are usually designed to mimic the shape of a human body in order to increase the acceptance of the robotic systems by the human user. To resemble the shape of a human body, the covering structure of humanoid robots has to be based on complex organic volumes with 3D-curved surfaces that can no longer be formed by the combination of simple cuboid or cylindrical volumes. These biologically inspired surfaces are mostly non-developable and cannot be covered with a single distributed sensor based on a rigid or one-dimensionally bendable circuit board. So, the prevailing trend is to cover the whole body of the robot or some of its parts with an array of individual sensors that, in general, use nearly all modes of transduction as piezoresistive [38, 39, 40], capacitive [41, 42, 43, 44, 45], optoelectronic [46]. Dahiya et al. [47] provide an exhaustive review on distributed tactile sensor technology and on its features highlighting various requirements and expectations such as flexibility/conformability, spatial resolution, wiring problems and technologies for communication and data transmission. It can be recognised that, while the use of tactile sensors in the contact point detection and pressure estimation is a diffuse practice, the development of a distributed sensor able to estimate both the magnitude and the direction of the applied forces is still an open challenge for robotic researchers. Some examples of sensors that can detect contact pressure have been presented in [26, 48, 49, 50, 51, 52, 53, 54]. In [48], [49], and [50], the authors presented distributed tactile sensors that use the FSR (Force Sensor Resistive) material. The contact is detected with a measure of resistance, since resistance of such material changes with the applied pressure. In [48], the proposed artificial skin is used to cover the whole body of an assistant robot: an energy-absorbing layer is introduced in the skin structure to decrease the risk of dangerous injuries in human-robot interaction, but it suffers of high hysteretic behaviour. In [49] the sensor is designed by the superposition of several layers with different thicknesses and materials, involving a complex manufacturing process. Strohmayr et al. [51] address the problem of providing a solution able to avoid tradeoffs between sensor sensitivity and robustness. The transduction principle of the proposed stretchable sensor is based on the acquisition of the electrical transfer resistivity between polymer-based electrodes arranged in two orthogonal layers. Ulmen and Cutkosky [52] proposed a solution, based on capacitive technology, where a force sensing capacitor is the sensing element and a three–plates capacitor structure acts as capacitor plates and noise shields at the same time. The main disadvantage of this solution seems to be the large dimension and the complexity of the circuit for addressing the sensor elements. Inspired by the characteristics of the human tactile sensation, Hoshi and Shinoda [54] proposed a tactile sensor based on two layers of compressible insulators (urethane foam) and on three pieces of stretchable conductive sheets. They also proposed a way to connect the sensor elements based on sensor/communication integrated circuits placed at the boundaries of the conductive pieces that measure the capacitance between the conductive layers. A layer of urethane foam is also used by Ohmura et al. [53] which proposed a scalable and modular distributed sensor based on optoelectronic technology. They introduced the concept of *cut-and-paste* tactile sensor to refer to the possibility of adapting their sensor skin to any part of the robot body. Surely, a weak point of such sensor is the required large current, and thus high power consumption: one LED requires about 50 mA. For 1000 tactile sensing elements, the total current amounts to 50 A, which is too large for this kind of applications. To the best of the authors knowledge, the optoelectronic technology is not widely used in the design of distributed sensor. Instead, a Vertical Cavity Surface Emitting Laser (VCSEL) is used in [55, 56, 57]. In [55], the VCSEL is used to develop an optical ultra-micro-displacement sensor fabricated using MEMS tecnology; in [56], the VCSEL, combined with a photodiode, is used to develop a flexible sensor able to estimate the shear stresses applied to its surface. Differently, in [58] an optic subsystem is introduced in the developed embedded device only to provide a reliable and high speed communication channel for the sensor data transmission and not for the force measurement. None of the sensors cited so far is able to estimate the force vector in a large contact area and only few different solutions are able to estimate the three components of the force vector applied to a small contact surface by an external colliding object. In [59, 60] a flexible tactile sensor array for an anthropomorphic artificial hand with the capability of measuring both normal and shear force distributions, using Quantum Tunneling Composite (QTC) as a base material, is presented.

It consists of four fan-shaped electrodes in a cell that decomposes the contact force into normal and shear components. However, as discussed by Amarasinghe et al. in [61], QTC technology suffers from different drawbacks that imply to use it or as a simple low-cost contact switch, or with additional complex mechanical structures to realize force sensors. Moreover, QTC also takes considerable time to return to its original state after deformation. Tao Liu et al. [62] designed a 3-D tactile sensor by integrating four sensing cells, each composed of a pressure-sensitive electric conductive rubber (PSECR) and a fan-shaped pectinate circuit. Both the solutions use the measurements of four sensing cells to estimate the normal force component and the shear force components through a simple empiric relationship that expresses the force components with respect to the sensor information, i.e, sensor voltages. Also in [63, 64] two capacitive solutions able to estimates the force vector are presented.

1.5 Thesis contribution

The literature shows that the common goal of different works is to develop a tactile sensor with a soft surface that can be conformable, scalable and adaptable to smooth curved surfaces of the robot body. A fundamental property of a distributed sensor that should be used as an artificial skin is its spatial resolution. The spatial acuity of human skin is an important parameter that gives an idea of the spatial resolution that an artificial skin should possess. In particular, the method based on the two-points threshold [65] and the grating orientation [66] shows that the spatial acuity varies across the human body: the spatial resolution of a fingertip (1 mm [67]) is better than the spatial resolution of the palm (7 mm [68]) and of the torso ($20-30$ mm [69]). Therefore, also the artificial skin spatial resolution should be adaptable to different needs. Moreover, safety operations between humans and robots require that the sensor response time is as low as possible so as to change the robot behavior before causing serious damage.

According to the discussion reported before, a distributed force/tactile sensor should satisfy the following requirements:

- the sensor should be capable to provide a direct measure of the pressure map and of the contact force vector so as to avoid complex model-based estimation algorithms, even in the case of multiple contacts
- the sensor should be highly modular and scalable
- the sensor should have fast time response

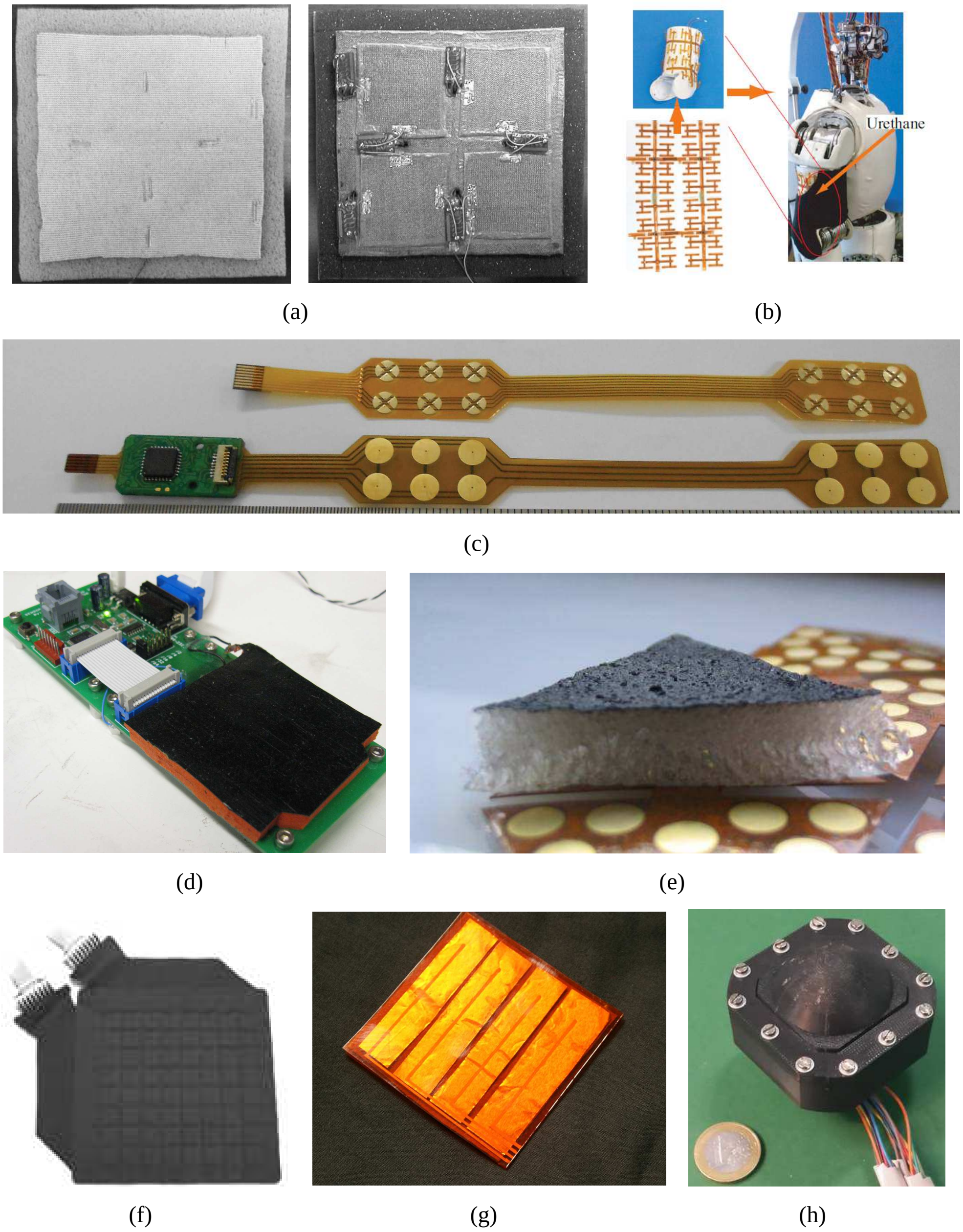


Figure 1.3: Pictures of force and tactile sensors proposed by: Hoshi and Shinoda (Fig. 1.3(a)), Ohmura et al. (Fig. 1.3(b)), Zhang et al. (Fig. 1.3(c)), Ulmen and Cutkosky (Fig. 1.3(d)), Cannata et al. (Fig. 1.3(e)), Elkmann et al. (Fig. 1.3(f)), Duchaine et al. (Fig. 1.3(g)), Palli et al. (Fig. 1.3(h)).

- the sensor should be characterized by light-weight, low-power consumption and low-cost, since it potentially could be used to cover the entire surface of a robot
- the sensor of the sensing elements should be adaptable to different spatial resolutions requested by the application site
- the sensing elements should be embedded into or covered with soft and/or elastic material
- the sensor should be easy to manufacture with a low number of wires
- the sensor measurements should be repeatable and with low hysteresis
- the sensor should be applicable to arbitrarily curved surfaces (conformability)

In this thesis the author deeply describes the design, the implementation and the use in robotics applications of a novel modular force/tactile sensor able to estimate both normal and shear contact force components. It has been demonstrated that the sensor can be exploited for all robotic and control application that require a force/-tactile feedback. Each sensing module consists of 4 taxels[2] covered by a silicone layer that transduces the external force into a mechanical deformation, measured by the four taxels. Each taxel consists of an infrared Light Emitting Diode (LED) and a Photo-Detector (PD) (also referred to as Photo-Transistor (PT)). The distributed sensor is realized by interconnecting several sensing modules with a suitable scanning strategy, used to obtain an highly modular and scalable solution. The sensor prototype has been calibrated, characterized and its main features have been highlighted with several tests. A flexible version of the developed sensor has been installed on several robotic arms and it has been used for manual guidance, intuitive programming, collision detection and reaction tasks.

In Section 2 the rigid sensor prototype is deeply described. Starting from the sensor working principle, the design and the development of a single sensing module are analyzed. It is explained how a distributed prototype can be obtained by interconnecting several sensing modules and the specific strategy adopted to interrogate the prototype is introduced and discussed. The procedure used to calibrate the sensor is reported and, then, the distributed sensor is characterized and its main features are highlighted with several tests.

[2]The word *taxel* derives from the union of the words “tactile element”.

Section 3 reports the design and the development of the conformable sensor prototype. The adopted scanning strategy allows a substantial reduction of the number of wires with respect to the sensing module number, which makes possible the use of a flexible PCB. A procedure for the integration of the sensor on a curved surface of a robot manipulator is defined, presented and discussed. The calibration procedure introduced for the rigid sensor prototype is adapted to the conformable sensor version in order to take into account the local curvature of the flexible version. Moreover, an alternative calibration technique is analyzed and compared with the previous one.

In Section 4 the use of the force/tactile sensor as Human-Machine Interface is described. Three algorithms for touch gesture recognition are introduced and assessed.

In Section 5 the developed sensor is introduced in the physical Human-Robot Interaction context. The sensor is installed on two lightweight manipulators for which it is possible to obtain information about the external forces acting on them through indirect estimation methods based on the theoretical approach of the residual method and on the use of joint torque sensors. An analysis of the mentioned methods, as well as a comparison with the proposed sensor in terms of estimation accuracy, are reported.

Section 6 describes the control algorithm adopted for the definition of the robotics tasks presented in this thesis. The mathematical formulation of the algorithm is reported and an analysis on the stability of the closed loop control algorithm is addressed through the Lyapunov method.

Section 7 addresses the problem of the sensor integration on different redundant manipulators. On the basis of the communication interfaces adopted by the specific robotics systems a proper sensor driver has been developed. The system architectures used for KUKA and YASKAWA 7-DOF robots are described as well as the sensor drivers. Finally, the force/tactile sensor is used for manual guidance, intuitive programming, collision detection and reaction tasks and a the experiments with their results are presented.

In Section 8 the conclusions are presented and possible future works and challenges are proposed.

Part I

The distributed force/tactile sensor

CHAPTER 2

The rigid sensor prototype

The sense of touch is an important means through which living creatures obtain external information and it plays an important role for the realization of direct interaction between robots and the environment. Tactile sensors are devices that measure parameters of interactions, regarding touch, pressure sensation, force, sliding feel, and heat sensation, between intrinsically sensitive areas and external objects. Distributed sensor devices include many tactile perception units, which are usually arranged in the form of tactile sensor arrays, to acquire the force distribution function of the contact between the sensitive area and an external object. Since the first development of a tactile sensor in the 1970s, the use in applications in the vast fields of smart robots, biomedicine, touch-screen technology, modern manufacturing, and modern services has grown rapidly. The functions of such sensors have expanded from the initial realization of single-dimensional normal-force measurement perpendicular to a surface to three-dimensional (3D) force measurement. With the rapidly growing number of applications and increasing number of user requirements, the distributed sensor design is focused on achieving flexibility, multidimensional force detection, miniaturization and multi-functionality.

The design of the described sensor passes through the definition of a rigid prototype, which has been deeply characterized and experimentally tested. The obtained results have encouraged further developments and evolutions of the rigid version that represented a good starting point for the design of a conformable sensor prototype. In the following sections both the rigid and conformable prototypes are described and particular interest is given to the calibration procedures and tests used to highlight the sensor capabilities. Part of the work described in this section is published in [70].

2.1 The basic idea

The distributed force/tactile sensor described in this thesis is based on the idea behind the tactile sensor concept introduced in [71], i.e., the use of optoelectronic devices to detect the local deformations, generated by an external contact force applied to a deformable layer that covers the optoelectronic layer. The tactile sensor presented in [71] has been designed as a stand-alone device to be integrated into anthropomorphic robotic fingers, capable of executing fine manipulation tasks. To this aim, the sensor consists of 16 taxels, with a dedicated low power 16-channels Analog-to-Digital Converter (ADC), with a resolution of 12 bit and a maximum throughput rate of 1 MSPS, directly integrated into the sensor itself. This configuration guarantees a high sensitivity to external stimuli. In particular, after a suitable calibration procedure, the 16 measurements allow, through a neural network, to estimate the three components of the force vector (estimation error less than 0.1 N) and the three components of the torque vector (estimation errorless than 1 Nmm). Furthermore, the measures from the 16 taxels allow also to reconstruct a pressure map on the whole fingertip, with a spatial resolution of about 2 mm, directly correlated to the external object shape, with a high sensitivity (minimum detectable force 0.05 N). As discussed in Chapter 1 for the design of a distributed force/tactile sensor, the features to be taken into account are different from those of a tactile sensor, since the main objective of a distributed sensor is not the fine manipulation but the human-robot interaction and human safety. In particular, some characteristics such as spatial resolution, accuracy of the force estimation and sensitivity can be relaxed in favor of additional features such as modularity, possibility to cover large areas with limited costs and power consumption, capability to discriminate multiple contact areas with the corresponding forces, ease of integration in different parts of the robot. The solution presented in this thesis addresses these aspects as detailed below. It uses four optoelectronic couples to realize a single sensing module able to estimate the three components of the force vector. The whole sensor consists of a matrix of sensing modules, suitably interconnected. This choice allows to estimate the three components of the force vectors wherever applied to the whole distributed sensor, differently from some author previous works [72, 73] where only the normal component of the force has been estimated, by guaranteeing all discussed features. The selected sensor architecture, differently from the tactile sensor described above, results scalable enough to be applied to robot surfaces such as torso, legs, arms: its spatial resolution can be properly adapted on the basis of the robot body part to cover by simply changing the distance

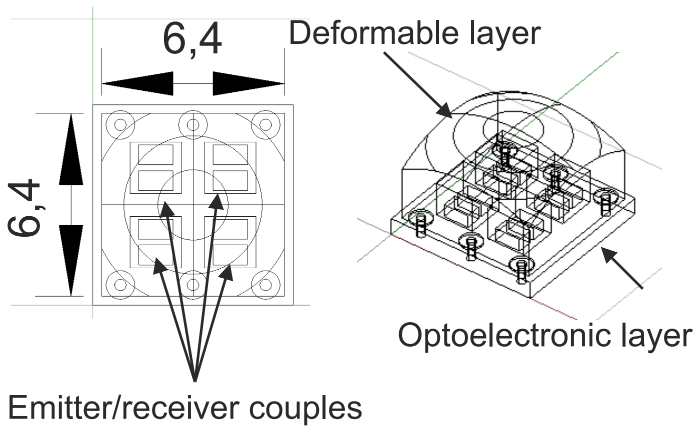
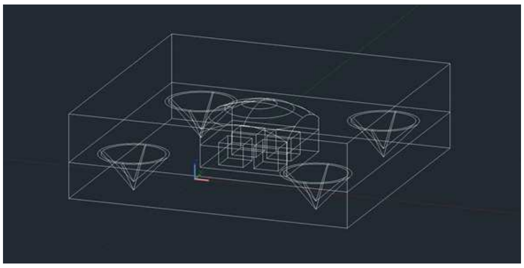


Figure 2.1: CAD model of a sensing module prototype: top view on left (dimensions are in mm) and perspective view on right.

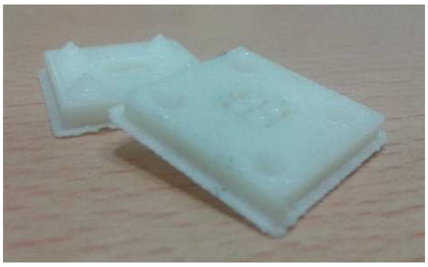
between two adjacent sensing modules.

2.2 Working principle

The sensor is obtained by interconnecting a number of identical sensing modules, each capable of measuring the three components of the contact force that acts on it. Each sensing module consists of four taxels organized in a 2×2 matrix. A single taxel consists of an optical emitter/receiver couple spectrally matched. A deformable elastic layer is positioned above the 4 optoelectronic couples (see Fig. 2.1). The deformable layer has a hemispherical shape on the top side, where the interaction with external objects occurs. On the bottom side it presents four empty cells into the material, with a parallelepiped shape, vertically aligned with the four optoelectronic couples. For each parallelepiped cell, the facet positioned in front of the optoelectronic couple must have optical properties able to guarantee a high reflectivity (reflective surface), while the lateral walls, which divide neighboring taxels, have to avoid optical cross-talk effects between taxels and also to ensure the immunity against external optical disturbances (absorbing surfaces). These properties can be guaranteed by using materials that allow to implement a molding of different layers with different properties (e.g., color, thickness, surface finishing). With this configuration the emitter illuminates the reflective surface of the corresponding parallelepiped cell and the reflected light is measured by the photodetector. An external force, applied to the deformable layer, produces displacement variations for all the four taxels constituting a sensing module. These displacement variations produce variations of the reflected



(a) CAD model.



(b) Printed unit.

Figure 2.2: Plastic molds for production of the silicone layer.

light and, accordingly, of the photocurrents measured by the photodetectors. Finally, the photocurrents are converted into voltage signals by using simple resistors. After a calibration procedure, detailed in Section 2.6.1, the external force components, acting on a sensing element, can be estimated with a suitable combination of the four measured signals. It is evident that the sensitivity and the full-scale of a sensing module depend on the hardness of the material used for the realization of the deformable layer.

2.3 The sensing module: enhancing the modularity

In order to realize the optoelectronic layer, the optical components have been selected on the basis of previous experiences, discussions and observations detailed in [71]. In particular, the realized prototype uses optoelectronic components manufactured by OSRAM. The LED (code SFH4080) is an infrared emitter with a peak wavelength of 880 nm, while the detector is a silicon NPN phototransistor (code SFH3010) with a peak sensitivity at 860 nm wavelength. Both the components have a viewing angle of $\pm 80^{\circ}$. The conditioning electronics is constituted by simple resistors without amplification and/or filtering stages, since the measured voltages are sufficiently high to be directly converted by using an ADC. The material selection for the deformable layer has been made on the basis of previous experiences detailed in [74]. In particular, a two layer plastic mold, suitably designed and realized by 3D printing (see Fig. 2.2), has been prepared in order to realize the deformable layer, by using black silicone for the absorbing walls and white silicone for the reflective surface. The black silicone guarantees the maximum absorption at all wavelengths and, as a consequence, to avoid cross-talk problems between taxels and light disturbances from the environment. The white silicone ensures the maximum reflection at all wavelengths, increas-

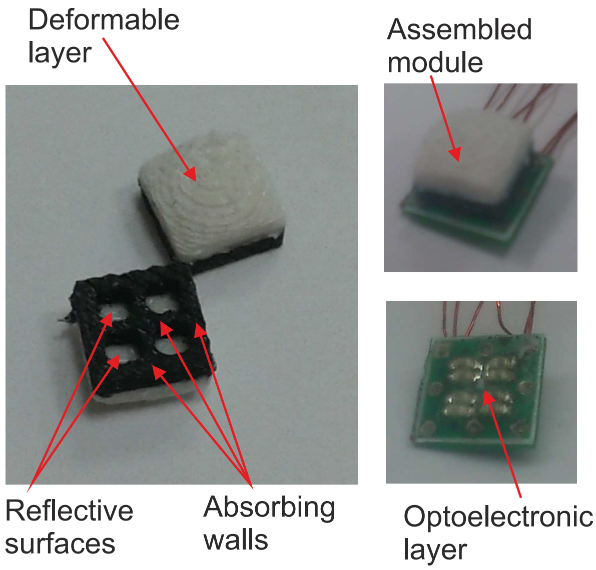


Figure 2.3: Pictures of sensing module.

ing the sensor sensitivity. Differently from a tactile sensor, a distributed force/tactile sensor requires a higher full-scale. To obtain these characteristics, a silicone with a higher hardness, with respect to the tactile sensor in [71, 74], has been chosen. The selected one is the MM928, provided by ACC Silicones Europe, with a Shore hardness of 28 A and a cure time of 24 h at room temperature. The aspect ratio of the black walls between taxels has been selected in order to reduce the horizontal deformations with respect to the vertical ones, by considering the Finite Element analysis reported in [74]. In particular, for the realized prototype, the thickness of the black walls is 0.8 mm, while the extension of the white reflecting surfaces is 1.8 mm×1.8 mm, which results in a total size for the deformable layer of 6 mm×6 mm. The height of the reflective surfaces from the electronic layer is 1.6 mm. The top of the deformable layer is a section of a sphere with a radius of 7 mm. The deformable layer is bonded on the electronic layer (of size 6.4 mm×6.4 mm) by using a cyanoacrylate glue. Figure 2.3 shows some pictures of the sensing module components and an assembled module.

2.4 From the sensing module to the distributed sensor: scanning strategy and power consumption

A distributed sensor prototype is realized by interconnecting several sensing modules. The version presented in this thesis consists of 36 modules, organized as a 6×6 matrix, for a total of 144 taxels. The sensor matrix has been developed as a shield that can be installed directly on a STM32F3 discovery board. The conformability property can be easily obtained by adopting a flexible PCB connected with the conditioning electronics by a thin wire, as described in Sec. 3. The STM32F303 MicroController Unit (MCU) provides sixteen ADC with a resolution of 12 bit: each voltage signal is digitized with two bytes and the selected MCU, with a system clock frequency of 72 MHz, represents the right trade-off between costs and performance. To ensure the scalability and the modularity of the system, a "scanning control" strategy, based on the same idea reported in [53], has been adopted to realize the module interrogation by using the MCU. The basic idea is to connect the sensing modules in groups which share 4 ADC channels, and to use the digital I/O of the MCU to switch on and off, with a cyclic pattern, the sensing modules, by ensuring that in each time instant, for each group, only one taxel is turned on, while all others, which share the same ADC, are turned off. This control logic is based on the fact that the switched off photodetectors behave as an open circuit that does not influence the A/D conversion of the voltage of the switched on photodetector. Differently from [53], the sensing modules can be directly driven by the MCU digital I/O, without using an external power supply, since each LED works with a forward current of about 1 mA and the voltage supply for all components is the 3.3 V, available from the MCU. Hence, since different groups use different A/D channels, sensing modules belonging to different groups can share the same digital I/O as power supply, by reducing also the number of digital I/O necessary to switch on and off the sensing modules during the interrogation. The described scanning strategy provides several advantages: a reduction of the whole sensor power consumption, since the number of modules simultaneously turned on is limited; a reduced number of ADC channels required to acquire the data; a simplification of the wiring. By generalizing the adopted interrogation technique, a total of n sensing modules (corresponding to $4n$ taxels) can be organized in m groups, each one constituted by p sensing modules. Since the sensing modules of each group share 4 A/D channels, the number of external wires needed to interrogate a sensor patch is equal to $4m + p$ (plus one for the ground). As a consequence, to minimize the number of wires needed for a sensor patch, the following constrained optimization problem can be solved

$$\min_{m,p} \; (4m + p) \tag{2.1}$$
$$\text{subject to: } mp = n, \quad m, n, p \in \mathcal{N}^{+}$$

with n, m, p positive integers. The developed force/tactile sensor has 144 taxels, divided into $n = 36$ sensing modules. By solving the optimization problem (2.1), the resulting minimum number of needed wires is $m = 3$ groups (corresponding to $4m$ A/D channels) and $p = 12$ digital I/O, for a total of 24 wires plus one for the ground.

Summarizing, the 144 total taxels, which constitute 36 sensing modules organized in 3 groups, are interrogated by using 12 ADC channels (4 ADC channels shared for each group) and 12 digital I/O used to implement the scanning strategy, for a total of 25 wires (the 25th signal is the ground) directly coming from a MCU (see the schematic diagram shown in Fig. 2.4). For applications where large surfaces have to be covered with a high number of taxels, the distributed force/tactile sensor proposed in this thesis presents very attractive properties from the power consumption point of view. Each taxel requires a voltage supply equal to 3.3 V with a current of about 1 mA, for an instantaneous power consumption of 3.3 mW. Since no additional ICs are necessary, with just a few watts of power, thousands of taxels can be driven at the same time. Generalizing, k taxels require a power consumption equal to $k \cdot 3.3$ mW. For the sensor patch proposed in this thesis, constituted by 144 taxels, a total instantaneous power consumption of 475, 2 mW would be needed if all taxels were always switched on. In this case, the power consumption would already be quite limited, but the interrogation technique described above allows a further power saving. In particular, at each time instant, only one sensing module is switched on for each group, corresponding to $4m$ taxels. With the optimal number of groups $m = 3$, only 12 taxels are switched on at the same time, with a total instantaneous power consumption of 39.6 mW, resulting in a reduction of one order of magnitude compared to the previous case. The only limitation can be the minimum sampling frequency necessary to interrogate the whole distributed sensor. For all the 144 taxels of the proposed patch, with the selected MCU, i.e., an ARM Cortex M4 STM32F303, a sampling frequency of 150 Hz was obtained. Therefore, the proposed solution is very attractive for battery-powered robotic systems. Finally, the 144 voltage signals are converted and transmitted via USB connection to a host PC. The microcontroller firmware is developed using the real-time embedded operating system ChibiOS/RT: a flow chart of the MCU operation is reported in Section 2.5. A Matlab script, running

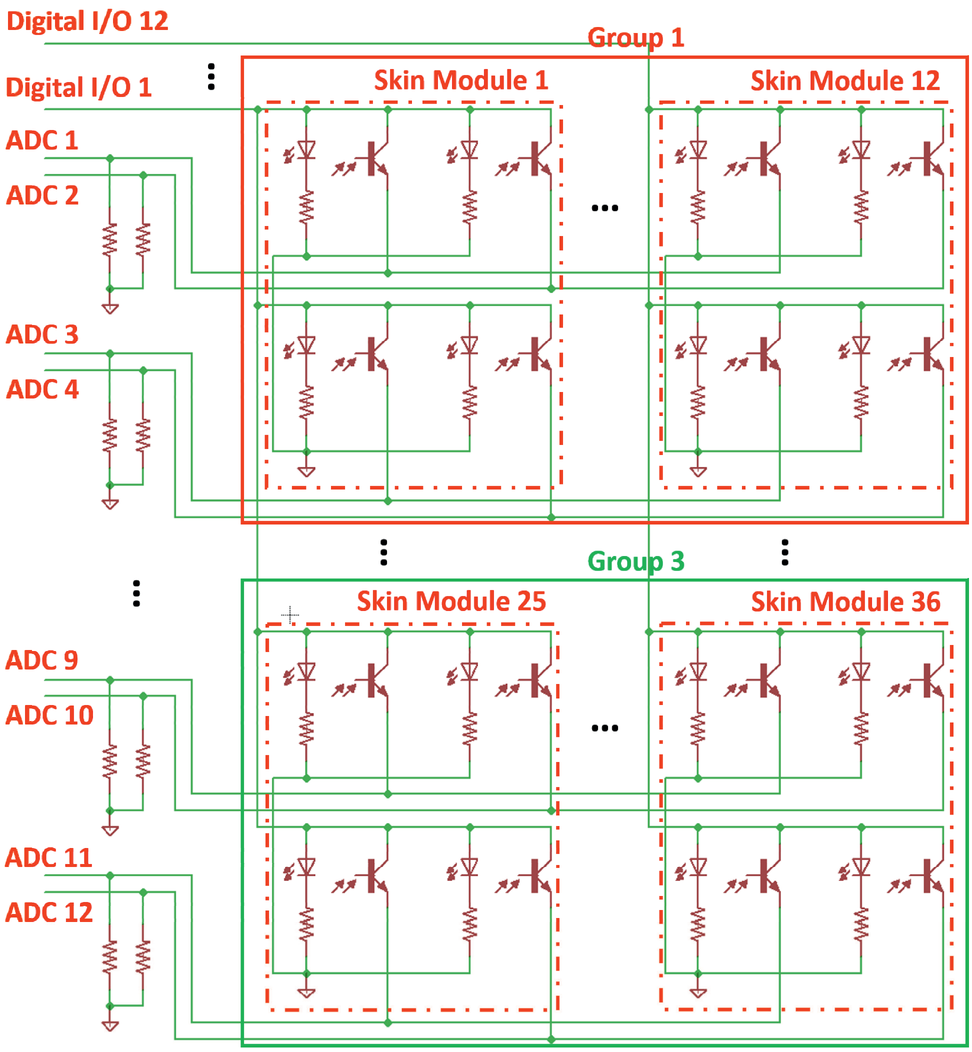


Figure 2.4: Electronic scheme of the interconnections between the sensing modules and microcontroller.

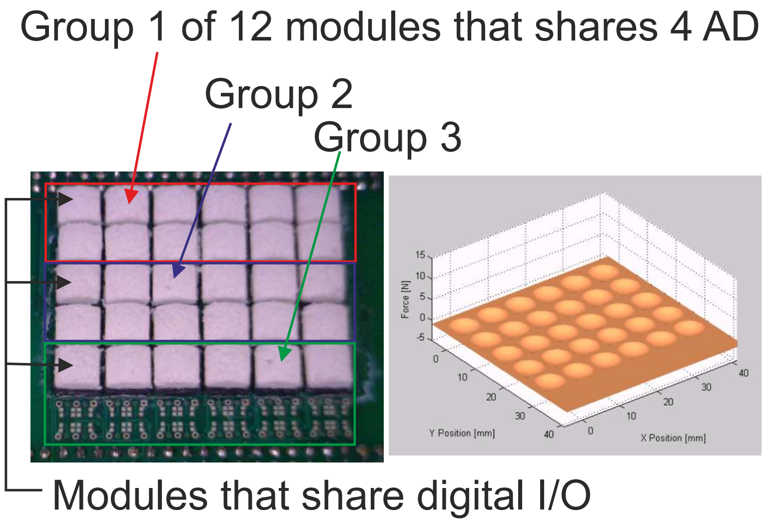
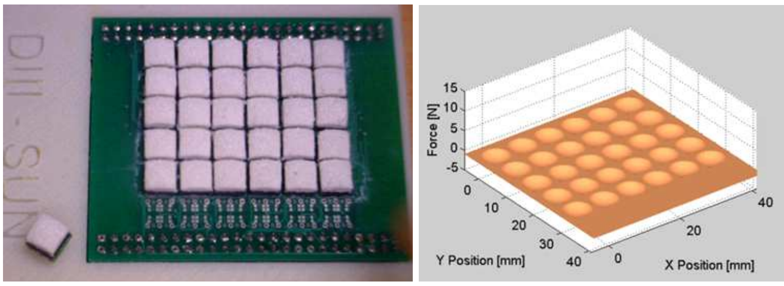
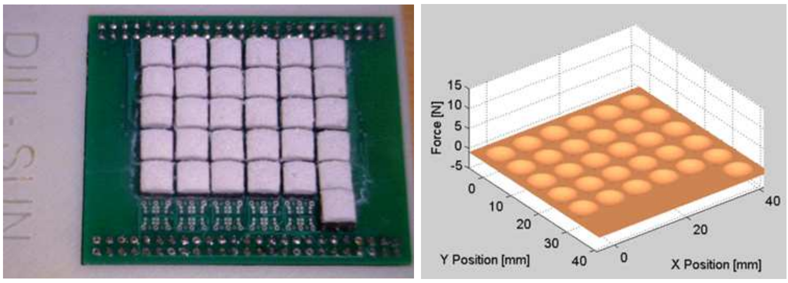


Figure 2.5: Rigid sensor prototype (left) and corresponding virtual sensor structure (right).

on the PC, acquires the data and computes the force components according to the algorithms reported in Section 2.6. The interrogation circuitry described so far has the great advantage that the sensor can work also if not all the sensing modules are actually connected or some connected modules are broken, since they appear only as open circuits. This improves the modularity of the proposed solution, since the number of the actually installed and/or working modules can be easily detected through an initialization phase. A virtual sensor structure can be reconstructed according to the detected modules, and used to show the information related to the estimated contact forces. Figure 2.5 shows a picture of the rigid prototype, together with the corresponding virtual sensor structure. Note how six modules on the third group are intentionally not mounted to show how the virtual structure automatically adapts. Such feature is better illustrated in Fig. 2.6, where a new single module has been connected to the last row of the sensor board and it is automatically recognized by the reading software module, which detects the number of the installed and/or working modules in the initialization phase. As reported in Section 2, the deformable layer of each sensing module was bonded on the electronic layer using a cyanoacrylate glue. In order to improve the reliability of the bond, to increase the loading cycles and in order to cover the whole sensor matrix with a single deformable layer, all sensing elements are connected together by an additional silicone molding. For a reliable contact force estimation, it is advised to have a negligible mechanical coupling between adjacent sensing elements. To avoid that such thin film could introduce a mechanical



(a) Disconnected module.



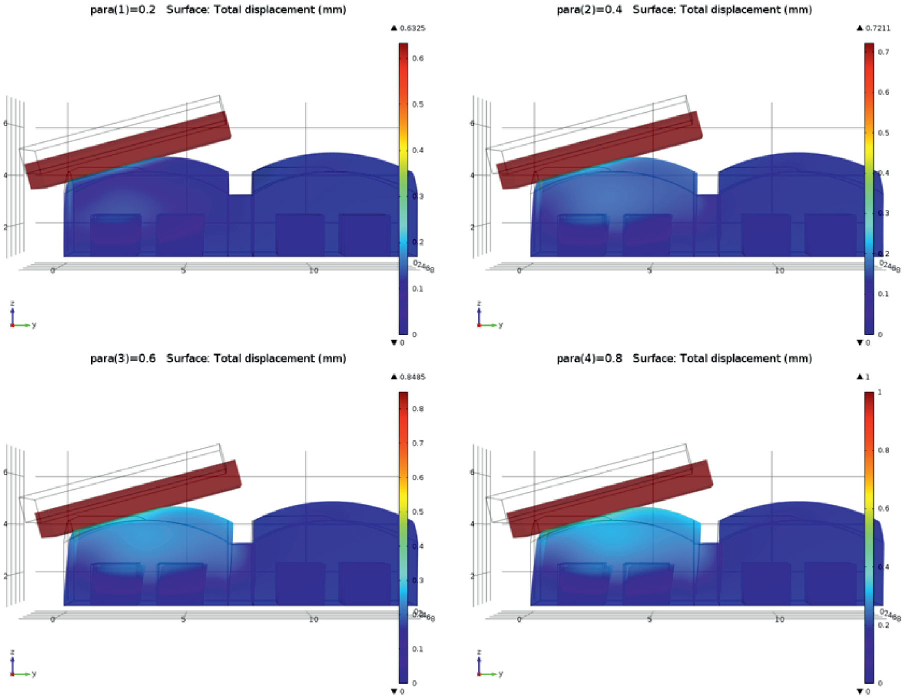
(b) Connected module.

Figure 2.6: Modularity of the scanning control strategy.

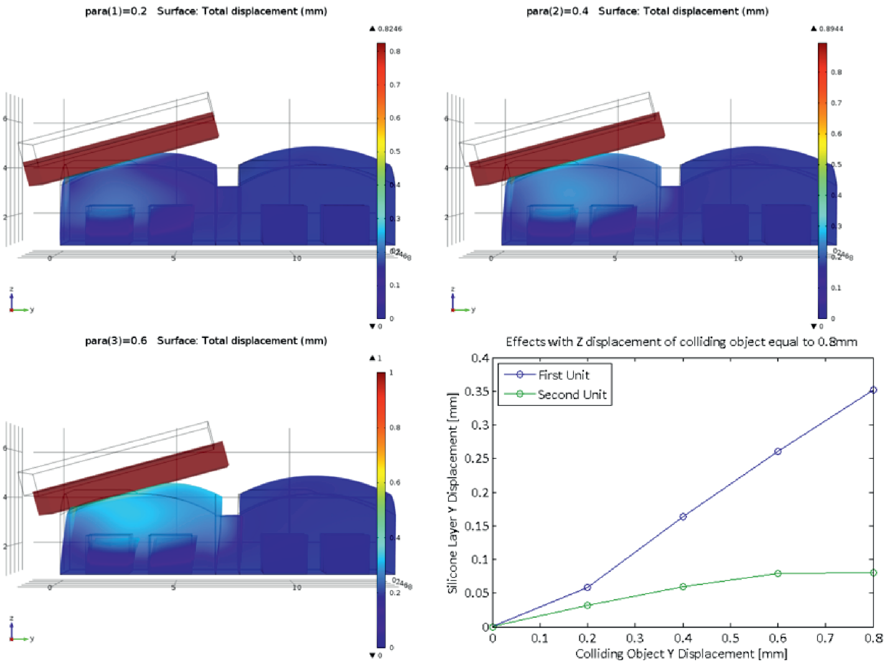
coupling, a FE analysis has been conducted (it has been presented in [75]) to verify that the use of a silicone with shore hardness of 6A (4-5 times more soft with respect to the one used to realize the deformable layer of the sensing elements) would not transmit significant stress from one cell to another. The silicone rubber behavior was modeled with the Mooney-Rivlin constitutive law:

$$\sigma = 2\left(\lambda - 1/\lambda^2\right)(\alpha_1 + \alpha_2/\lambda) \tag{2.2}$$

where λ is the elongation ratio. The model parameters α_1 and α_2 have been evaluated on the basis of the considerations reported in [74]. They have been chosen as $\alpha_1 = 3.96 \cdot 10^{-2}$ and $\alpha_2 = -3.37 \cdot 10^{-4}$ for the silicone with shore hardness 6A, while as $\alpha_1 = 0.16$ and $\alpha_2 = 0.13 \cdot 10^{-2}$ for the silicone with shore hardness 28A. In Figure 2.7, the results of the FEM simulations are reported. The 3D model is constituted by two sensing module deformable layers made of silicone with shore hardness 28A, by one layer of silicone with shore hardness 6A positioned between the two modules and by a plane of aluminium material that represents the colliding object. Latter is subjected to a prescribed displacement chosen in order to generate a mechanical contact between the plane and one deformable module and, then, a contact force. Referring to the reference system reported in Figure 2.7(a), the displacement is applied



(a) First simulation: prescribed displacement along z-axis of 0.6 mm.



(b) Second simulation: prescribed displacement along z-axis of 0.8 mm.

Figure 2.7: FEM analysis for the characterization of the second silicone molding.

only along the y- and z-axis. For the first simulation, the applied vertical displacement is 0.6 mm and the horizontal displacement ranges from 0 mm to 0.8 mm with a step size of 0.2 mm; for the second simulation, the applied vertical displacement is 0.8 mm and the horizontal displacement ranges from 0 mm to 0.6 mm with a step size of 0.2 mm. The mesh geometry is uniform for all the 3D model parts and it consists of 9469 tetrahedral elements. The last picture of Figure 2.7(b) reports a comparison between the displacements along the y-axis that affects the two sensor deformable layers: the results show that for a maximum deformation of the first sensing module equal to 0.35 mm, the corresponds maximum deformation on the second module is only 0.08 mm, that guarantees a negligible mechanical coupling. Figure. 2.8 shows the complete rigid sensor prototype. The figure highlights the sensing element deformable layers with the shore hardness of 28 A and the areas with the shore hardness of 6 A. It is evident how all the modules are now fully embedded into the silicone, which gives an improved mechanical robustness.

2.5 Sensor interrogation firmware

ChibiOS/RT provides a set of HAL functions that allow to easily manage the MCU peripherals. The two ADC units of the MCU are used to convert the sensor analog signals in different steps according to the adopted scanning control strategy and the USB unit is used for the MCU-PC communication. With a cyclic pattern, the signals related to the *i*th module of the three groups are digitized using 12 A/D channels. Figure 2.9 shows a simple flow chart that summarizes the MCU operations. The firmware starts with the ADC and USB peripherals configuration and, then, it waits the "start" command sent by the user/PC. So, the first sensing modules of the three groups are turned on, the corresponding analog signals are digitized and the data are sent to the PC via USB communication. The same operations are performed for the second modules of the three groups and so on.

2.6 Calibration, characterization and testing

The calibration of the sensor prototype is based on the hypothesis that the calibration of a single module can be used also for the other ones, since all modules are realized with the same components and they are mechanically separated. Actually, the assembly of the sensor, e.g., the soldering of the optoelectronic components, the positioning and the bonding of the deformable layer, could introduce differences in the

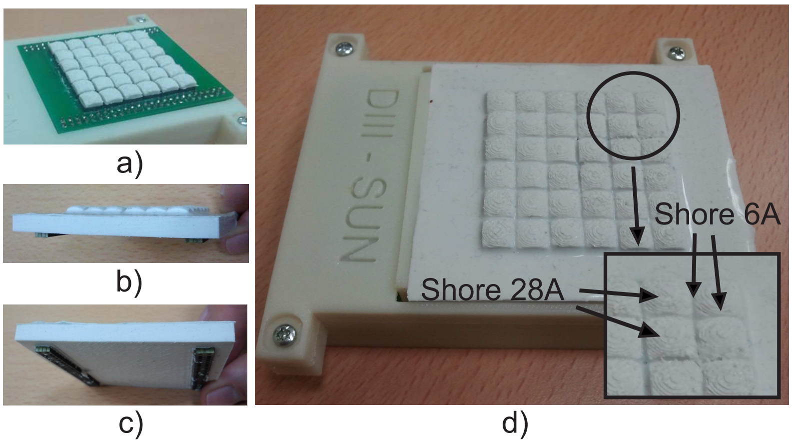


Figure 2.8: Perspective view of the 6×6 sensor matrix after the first silicone molding (a). Lateral view (b) and perspective view (c) of the 6×6 sensor matrix after the second silicone molding. The first complete sensor prototype connected to the conditioning electronics (d).

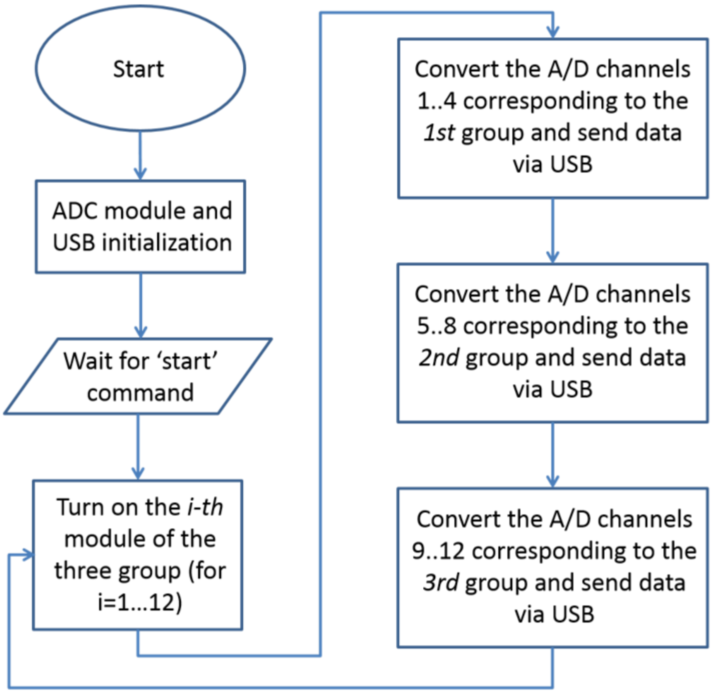


Figure 2.9: MCU firmware flow chart.

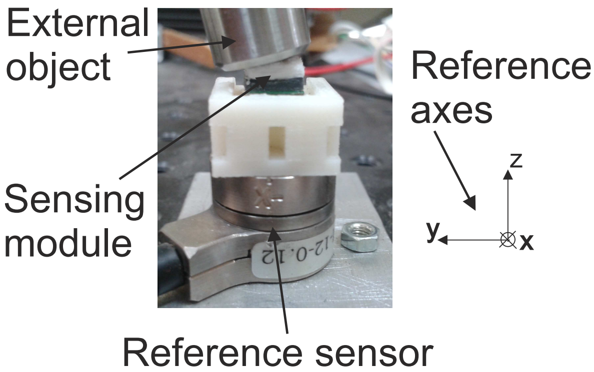


Figure 2.10: Setup for the calibration of a sensing module.

response of the sensing modules. So, in order to obtain the maximum accuracy in the force estimation, the identification of calibration parameters for each sensing module is advised. However, if the specific application does not require a high estimation accuracy, the same calibration parameters can be used for all the sensing modules of an entire sensor prototype. Accordingly, this section presents the calibration procedure and the characterization of a single module. The obtained calibration functions are applied to all sensing modules. Only the main results are reported in the following subsection.

2.6.1 Sensor calibration and characterization

The force components can be estimated as a suitable combination of the four voltages of a sensing module. A specific calibration setup has been prepared in order to acquire at the same time the module voltages and the actual force vector, measured by using a reference sensor. Figure 2.10 reports a picture of the setup with the corresponding reference axes. The sensing module is mounted on a six-axis load cell used as reference sensor. The model used is the FTD-Nano-17, manufactured by ATI, with a measurement range equal to ±12 N and ±17 N for horizontal and vertical force components, respectively. The measurement range for all torque components is equal to ±120 Nmm. An operator carried out various experiments using a stiff plane and by applying different external forces and, simultaneously, acquiring all the voltage variations on the phototransistors and all the forces components measured by the reference load cell. These data are acquired at a sample rate of 100 Hz. Considering the working principle described in Section 2.2, if the contact force is zero, each photodetector measures an initial voltage proportional to the light reflected by the white silicone when the deformable layer is in rest position. When an external force is applied to the deformable layer, each photodetector can present a positive or negative voltage variation with respect to the initial voltage, depending on the external force components. Figure 2.11 shows the voltage variations, measured by the sensing module, and the corresponding force components, measured by the reference sensor. It is evident that the sign of the voltage variations is related to the tangential force direction, and their amplitude to the force vector intensity. From the figure, it is also clear that the voltage variations are sufficiently high to be directly digitized without the introduction of additional amplification and/or filtering stages, as described in Section 2. So, if f_x, f_y, and f_z are the force components, and $\mathbf{V} = [V_1, V_2, V_3, V_4]^T$ is the vector that contains the voltage variations, the phenomenological model proposed to calibrate the sensing module is the following

$$f_x = \mathbf{k}_x^T \mathbf{V} \tag{2.3}$$

$$f_y = \mathbf{k}_y^T \mathbf{V} \tag{2.4}$$

$$f_z = \mathbf{k}_z^T \mathbf{g}(\mathbf{V}) \tag{2.5}$$

where the vector function $\mathbf{g}(\cdot)$ is simply the absolute value applied to each component of the vector $\mathbf{V}$ and the three 4×1 calibration vectors $\mathbf{k}_x$, $\mathbf{k}_y$ and $\mathbf{k}_z$ can be easily estimated with a simple least square algorithm by inverting Eqs. (3.3), (3.4) and (3.5), respectively, written for each point of the data set acquired as explained above (an example is reported in Fig. 2.11).

The accuracy of the calibration has been validated with a second data set, not used for estimated the calibration vectors. In particular, the estimated force components have been computed as

$$\widehat{f_x} = \mathbf{k}_x^T \mathbf{V} \tag{2.6}$$

$$\widehat{f_y} = \mathbf{k}_y^T \mathbf{V} \tag{2.7}$$

$$\widehat{f_z} = \mathbf{k}_z^T \mathbf{g}(\mathbf{V}) \tag{2.8}$$

and in Fig. 2.12 the estimated values $\widehat{f_x}$, $\widehat{f_y}$ and $\widehat{f_z}$ are compared to the actual force components f_x, f_y and f_z measured with the reference sensor, to evaluate the calibration performance. The results show a full-scale normal force of about 8 N and about ±2 N for the tangential components, with an estimation accuracy of about 0.5 N. The full-scale can be adapted to the requirement of a specific application, by changing the mechanical properties of the deformable layer (e.g., hardness, curvature radius of the hemispherical shape). The accuracy also depends on the full-scale and it could

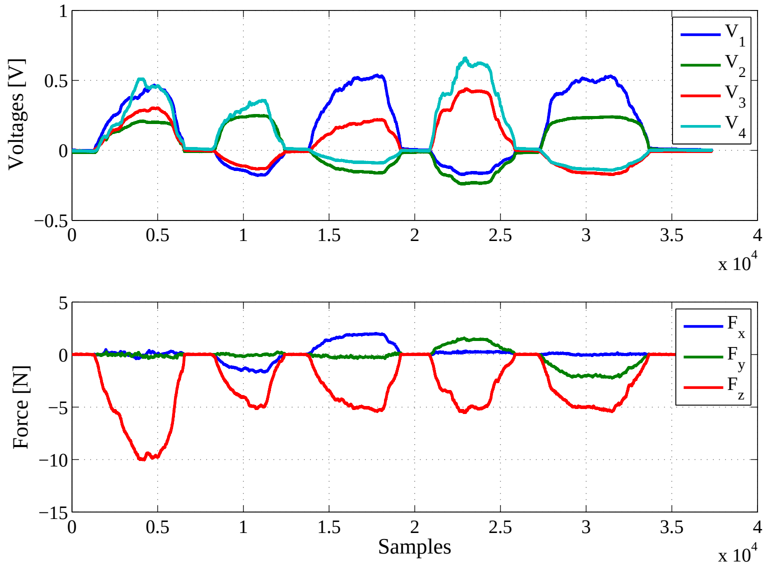


Figure 2.11: Voltage variations (top) measured by the sensing module and corresponding force components (bottom) measured by the reference sensor.

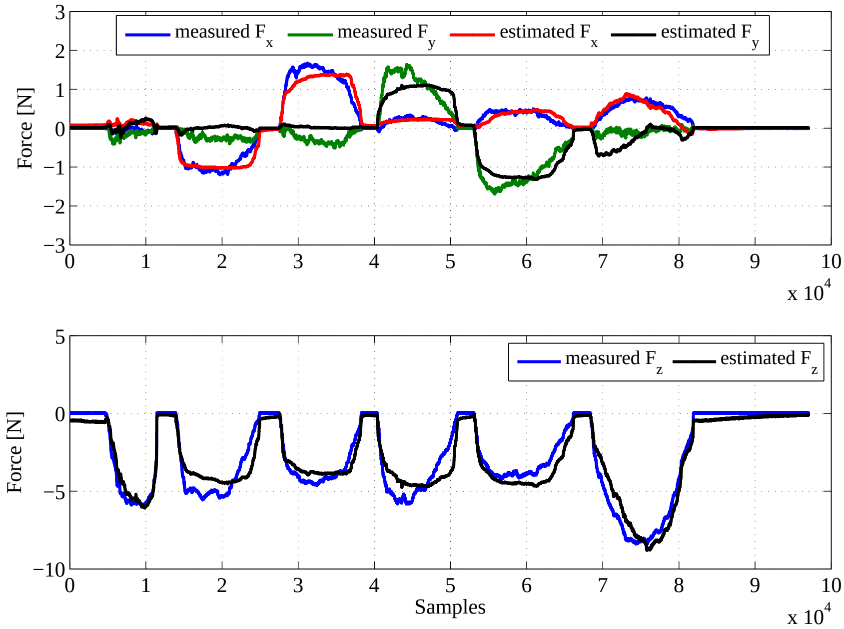


Figure 2.12: Example of force vector estimation: tangential force components (top) and normal force component (bottom).

Table 2.1: Example of calibration parameters.

$\mathbf{k}_x^T$	2.1519	−4.2440	−1.1642	0.1076
$\mathbf{k}_y^T$	−0.7988	−2.3203	1.0681	2.1013
$\mathbf{k}_z^T$	−2.0325	−6.2748	−9.1026	−8.1753

be improved by introducing a more complex model. Table 2.1 reports the calibration parameters used to estimate the force components shown in Fig. 2.12.

In order to assess both repeatability and hysteresis properties of the relationship between the external force applied to the deformable layer and the phototransistor signal variations, a few calibration experiments have been carried out. Using a micro-positioning stage, a known external force has been applied to the deformable layer that ranges from 0 N to 8 N with intervals of 0.2 N. The repeatability has been evaluated by acquiring more than once the calibration curve for a single taxel. In particular, Fig. 2.13 reports two measurements of the voltage variations for the same force applied to a single taxel, which denote a good repeatability with a maximum error of 6.77 %. To evaluate the hysteresis properties of the sensor due to the deformable layer, used to cover the electronic components, some measures have been carried out by increasing and decreasing the applied force. Defining the hysteresis error as the maximum difference between the output values of the sensor obtained for the same input value, then the maximum error is 10.27 %. The results are reported in Fig. 2.14. Both repeatability and hysteresis errors refer to the worst case that occurred during the different experiments.

Finally, a force pressure with a step change has been applied to the sensor to analyze the response time, here defined as the delay between the reference sensor signal and the voltage signal of a sensor taxel. The response time, generally, is influenced by the viscosity of the deformable layer material and by the characteristics of the selected transduction method. The optoelectronic technology combined with a silicone material provides a very low response time (about 1 ms), as shown in Fig. 2.15.

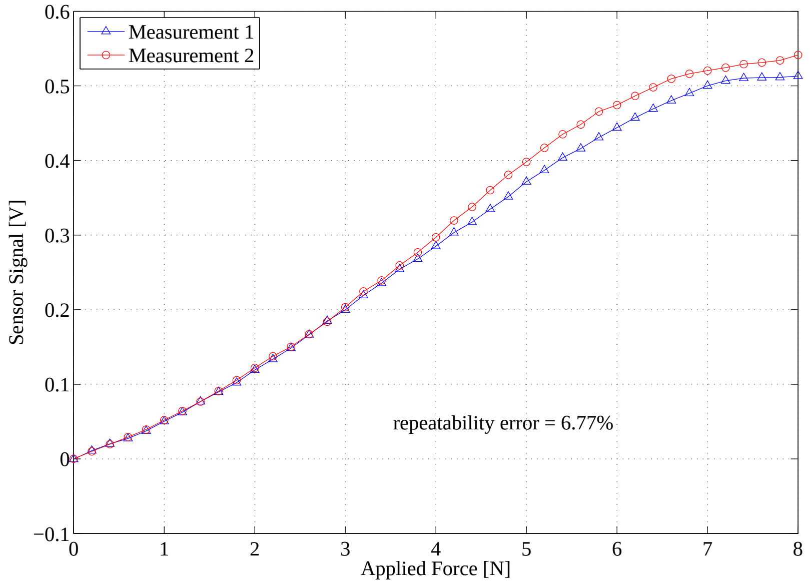


Figure 2.13: Evaluation of sensor repeatability: two measurements executed on the same taxel (the worst case has been reported).

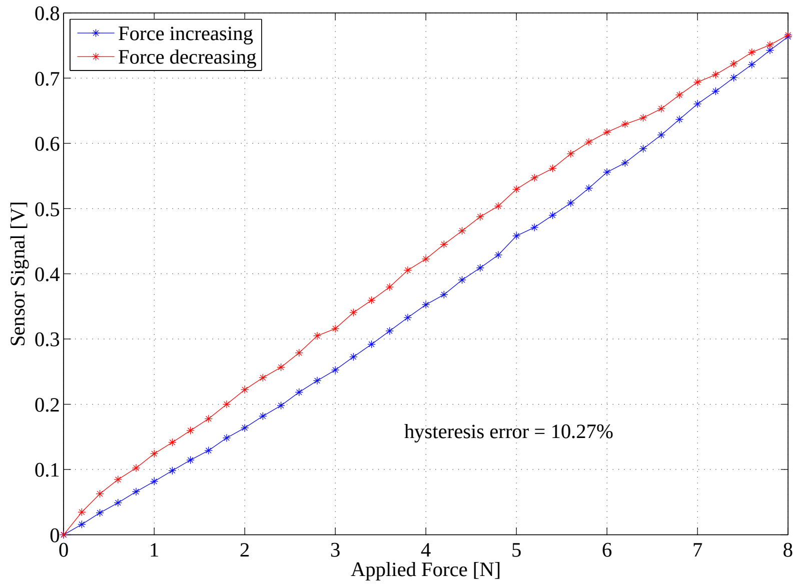


Figure 2.14: Evaluation of sensor hysteresis: two consecutive measurements executed with increasing and decreasing force on the same taxel (the worst case has been reported).

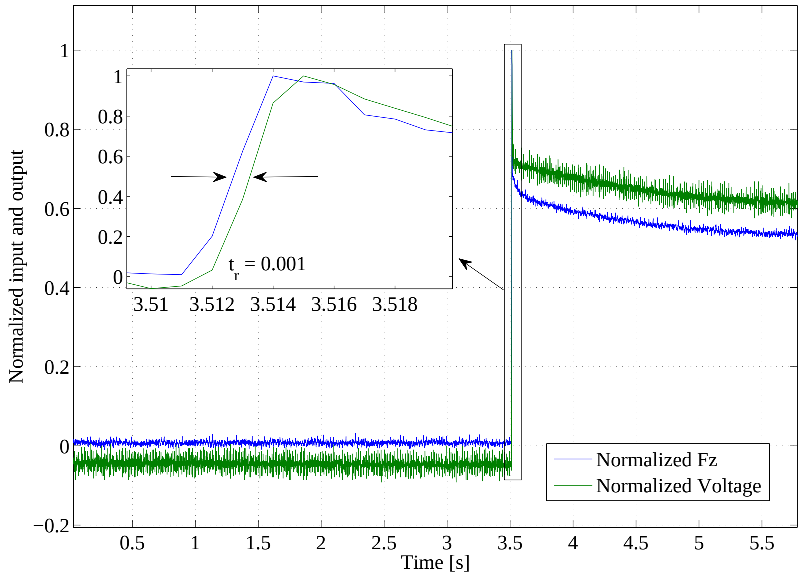


Figure 2.15: Evaluation of sensor response time: response of taxel to a step change in the applied force.

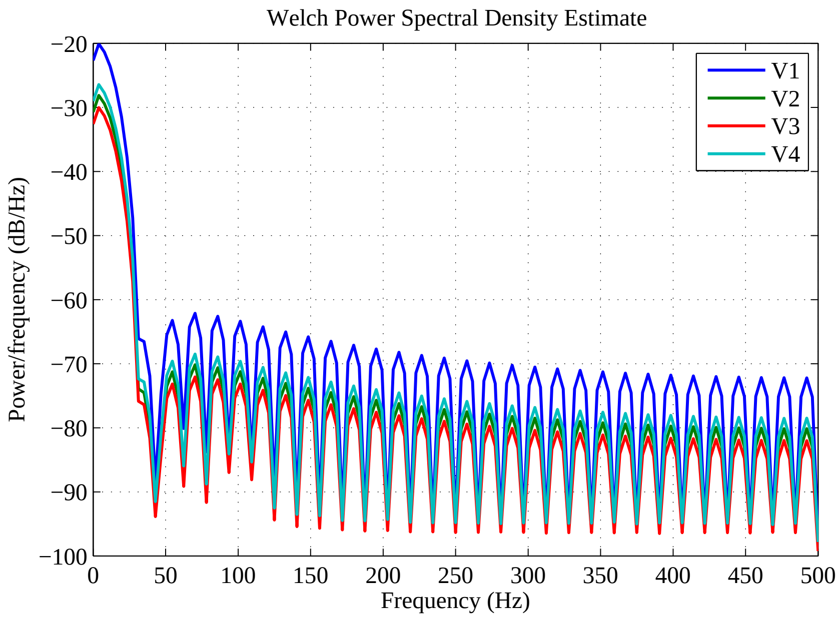


Figure 2.16: Evaluation of the PSD of four sensor voltage signals.

Table 2.2: Sensor comparison.

Year	Author	Transduction method	Miniaturization technique	Force vector estimation	No. of sensing elements	Spatial resolution	Signal conditioning electronic	Range of Force [N]/ Pressure [kPa]	Force/Pressure Sensitivity
2006	Hoshi et al.	Capacitive	Polymer	No	2×2	——	No	10 N	——
2006	Ohmura et al.	Optical	Flexible PCB	No	32	——	Yes	500 − 600 kPa	——
2008	Maggiali et al.	Capacitive	Flexible PCB	No	12	10 mm	Yes	128 kPa	——
2009	Duchaine et al.	Resistive	Polymer	No	16	10 mm	Yes	——	——
2013	Liu et al.	Resistive	PSECR	Yes	——	10 mm	Yes	100 N (Normal) 35 N (Shear)	——
2010	Ulmen et al.	Capacitive	Polymer	No	4×4	10 − 20 mm	Yes	~ 100 N	0.02 N
2011	Elkman et al.	Resistive	Polymer	No	8×8	——	Yes	——	0.25 N/cm^2
2013	Strohmayr et al.	Resistive	Polymer	No	——	2.5 mm	Yes	50 kPa	10 − 20 kPa
2013	Zhang et al.	Resistive	QTC	Yes	1	——	Yes	20 N (Normal) 7 N (Shear)	——
2014	Palli et al.	Optical	Rigid PCB	Yes	1	——	Yes	100 N (Normal) 50 N (Shear)	——
2014	Bekhti et al.	Capacitive	Multimaterial	Yes	1	20 mm	Yes	30 N (Normal) 20 N (Shear)	——
2014	Cirillo et al.	Optical	Flexible PCB	Yes	6×6 [12×12][(1)]	7.4 mm [2.6 mm][(2)]	Yes	10 N (Normal) 2 N (Shear)	~ 0.1 N

Notes:

(1) A 6×6 matrix of sensing elements has to be considered in the case of force vector estimation, while, in the case of contact point estimation, a 12×12 matrix of taxels can be considered.

(2) 7.4 mm refers to the distance between two adjacent sensing elements. So, it represents the spatial resolution in the case of force vector estimation. Considering the contact point estimation, the spatial resolution is determined by the distance between two adjacent taxels (2.6 mm for the presented implementation of the sensor).

Moreover, Fig. 2.16 shows the Power Spectral Density (PSD) of the voltage signals reported in Fig. 2.11, demonstrating that the noise level is below the signal level of about four orders of magnitude, since the signal bandwidth is limited to a few Hertz. In order to provide a comparison between the main sensors described in Section 1.4 and the proposed solution, the main characteristics, i.e., transduction method, spatial resolution, full-scale range, have been summarized in Table 2.2 on the basis of the comparison reported in [76]. To the author best knowledge no existing distributed sensing solutions are able to estimate the three components of the force vector; moreover, not all the sensors characteristics are available in literature and only a partial comparison was possible and reported in Table 2.2.

The calibrated force/tactile sensor has been tested in contact with different objects in various contact conditions and the results are discussed in the following sections.

2.6.2 Sensor testing

The rigid sensor has been used to perform several tests. First of all, since each taxel measures a local deformation, the 144 voltages, measured with the scanning strategy described in Section 2, can be used as a pressure map with a spatial resolution of 2.6 mm. This map allows to discriminate multiple contacts and to reconstruct the shape of the objects in contact with the force/tactile sensor. Figure 2.17 shows these features with two different experiments. The calibration functions (3.6), (3.7) and (3.8), have been used in the Matlab script to estimate the force vector for each module. The same function has been applied to the four voltage variations for all sensing modules, obtaining as many estimated force vectors as the sensing modules are. Figure 2.18 shows the estimated force vector on the virtual sensor structure, in the case of a single contact with a single sensing element.

If multiple contacts occur, the Matlab script first identifies all groups of sensing modules that constitute connected components of the whole contact area according to the algorithm detailed below. Then, it computes a force resultant for each connected component as the vector sum of the force vectors estimated by the single modules. The application point of a resultant force is computed as the centroid of the corresponding connected component. Figure 2.19 shows a typical example with two different contact areas, each consisting of 4 modules. The force vectors estimated by the single sensing modules that constitute the two contact areas are reported on the virtual sensor structure. Figure 2.20 reports another example with three different contact areas. In this case, the virtual sensor shows, in red, the force vectors estimated by the single sensing modules that belong to the two connected components of the

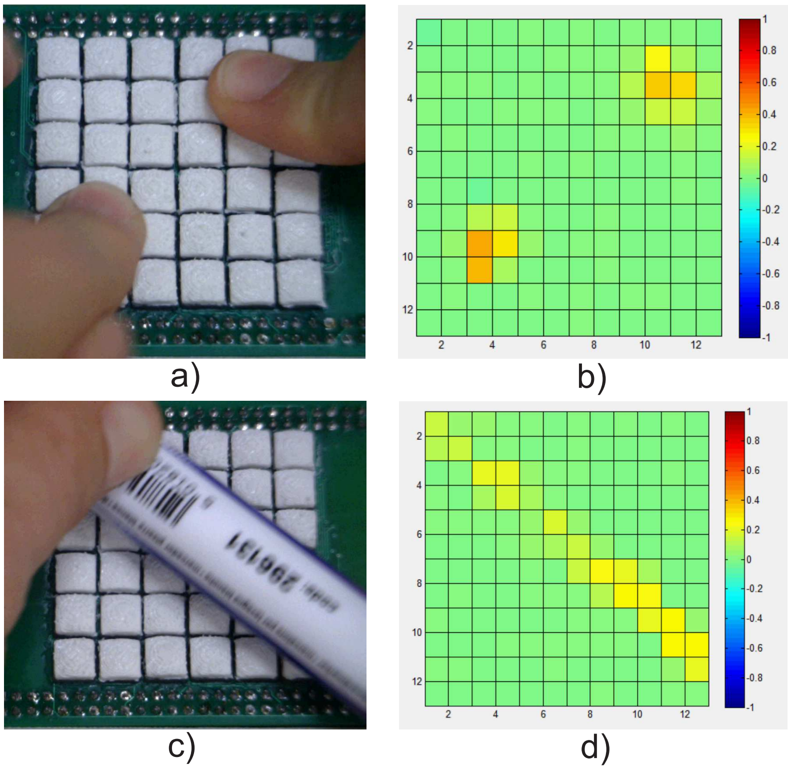


Figure 2.17: Force/tactile sensor as pressure sensor: multiple contacts (a) and their corresponding pressure map (b); distributed contact with a large object (c) and its corresponding pressure map (d).

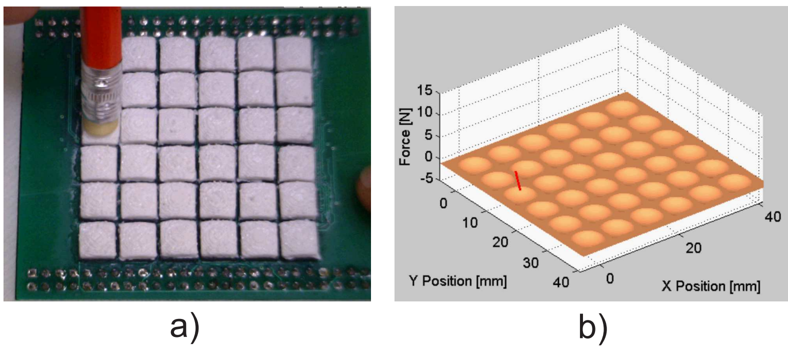


Figure 2.18: Estimation of the force vector, reported on the virtual sensor structure (b), for a single contact with a single sensing module (a).

contact area and, additionally, the corresponding resultant forces. Figure 2.21 reports the resultant force and the force vectors estimated by the single sensing modules in the case of a distributed contact, i.e. when a marker pen is in contact with the sensor patch.

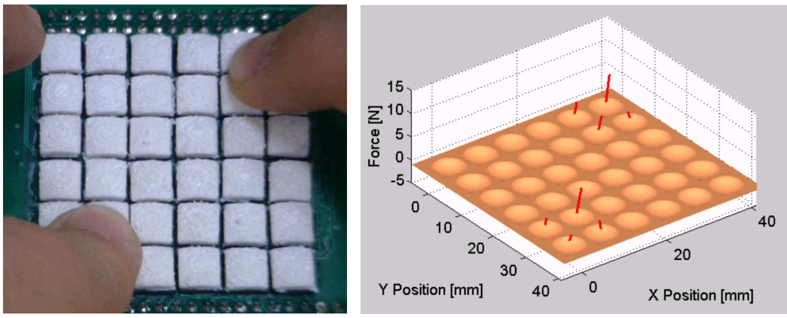
2.6.3 Algorithm for multi-point contact

A specific algorithm to identify multipoint contact conditions has been devised. It first identifies all groups of sensing modules that constitute a connected component of the whole contact area. Then, it computes a resultant force for each connected component as the vector sum of the estimated force vectors by the single modules belonging to the identified connected component. The application point of a resultant force is computed as the centroid of the corresponding connected component.

Let $\mathbf{M}$ be the matrix whose elements are equal to 1 in correspondence of the sensing modules on which a contact occurs. In order to avoid false contact detections, the element (i, j) of $\mathbf{M}$ is set to 1 only if the magnitude of the estimated contact force on the module (i, j) is greater than a threshold value f_t. The threshold value of 0.5 N has to be equal to the average calibration error of the sensor. To detect the sets Ω_i of adjacent cells on which a distributed contact is applied, an iterative algorithm has been designed. Starting from the first module of the sensor (considering a prototype of 6×6 modules, the first module is in the upper-left corner of the matrix) the algorithm verifies if on the 8 adjacent modules a force is applied and, then, the corresponding element of the matrix $\mathbf{M}$ is set to 1. In this case, a connected component Ω_i is created and the matrix element related to the new modules inserted in it, is set to 0. The algorithm is iteratively executed on the new elements of the connected component until no new adjacent modules result activated by a contact force. In practice, a tree is associated to each connected component Ω_i, where each entry (the node of the tree corresponds to a sensing module) can have a number of branches up to eight. The pseudo-code of the implemented algorithm is reported below as the main algorithm (Algorithm 1) and the subroutine (Algorithm 2) that finds the connected components.

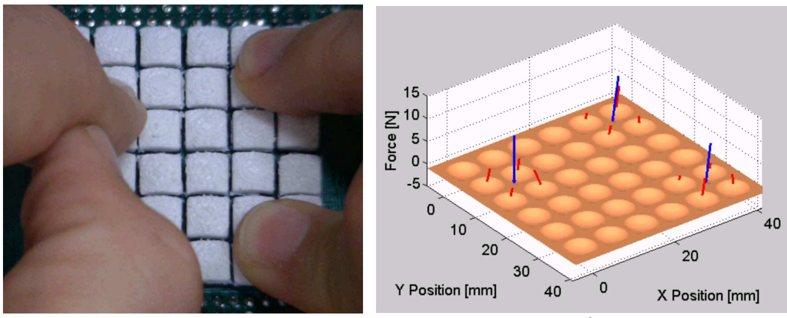
Algorithm 1 Main algorithm

```
for all elements of M do
    if element is equal to 1 then
        Ω_i ← find_set(M, element_index)
        F_i ← vectorial sum of f_j where j is the identifier of the Ω_i elements
    end if
end for
```



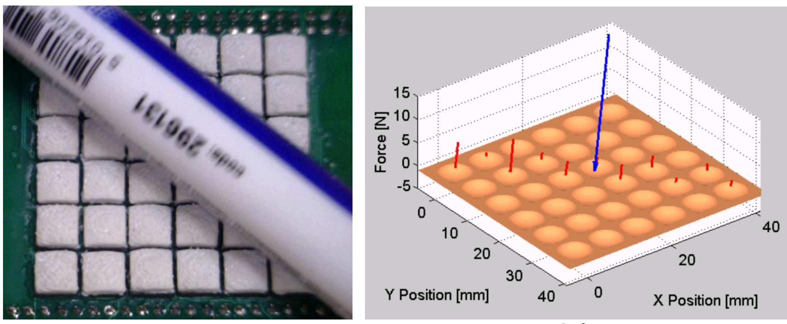
a) b)

Figure 2.19: Estimation of the force vectors, reported on the virtual sensor structure (b), in a multiple contact case (a).



a) b)

Figure 2.20: Estimation of the force vectors in a multiple contact case (a): on the virtual sensor structure (b) are reported both the force vectors estimated by the single modules (red arrows) and the net force vectors for each contact area (blue arrows).



a) b)

Figure 2.21: Estimation of the force vectors (b) for a distributed contact (a).

Algorithm 2 Subroutine

Function **find_set** (**M**, *initial_element*) return Ω, **M**
if **M** (*initial_element*) is equal to 1 **then**
  add initial_element to Ω
  M (*initial_element*) $\leftarrow 0$
else
  compact set does not exist
  return
end if
while set not found **do**
  for all element k of Ω not analysed **do**
    find all adjacent elements l to k-th element so as **M** (l) is equal to 1
    if l is not empty **then**
      add l to Ω
      M (l) $\leftarrow 0$
    else
      set not found
    end if
  end for
end while
return Ω, **M**

CHAPTER 3

The flexible and conformable sensor prototype

The design of the conformable sensor patch, based on flexible PCB technology, is an evolution of the rigid prototype design made possible by two main characteristics of the proposed sensor, namely the smart scanning control strategy and the low power consumption of a sensing module. The scanning strategy allows a substantial reduction of the number of wires with respect to the taxel number, which makes possible the use of a flexible PCB with a limited number of layers corresponding to a reduced thickness, which guarantees high conformability for the optoelectronic board and low production cost.

3.1 Design of the conformable sensor

The design of the flexible PCB affects the maximum achievable flexibility of the sensor patch, hence some observations are in order. Firstly, the installation of the electronic components on the flexible PCB reduces the flexibility property, depending both on the number and the dimensions of the components. Secondly, the flexibility depends also on the number of layers necessary for the wiring, thus a proper routing of the PCB should be carried out. This requires a suitable re-design of the optoelectronic layer of the original rigid prototype to maximize conformability of the new force/tactile sensor version. First of all, note that the sensing modules are only constituted by the optoelectronic components (SFH4080 and SFH3010), that have a SmartLED package 0603 (with dimensions $1.3 \times 0.8 \times 0.65$ mm), and additional resistors to drive the LEDs (a resistor for each LED), with package 0402 (with dimensions $1 \times 0.5 \times 0.32$ mm). By using the scanning strategy described in Section 2, each group of sensing modules can share the resistors in series with the PTs. With this choice, the

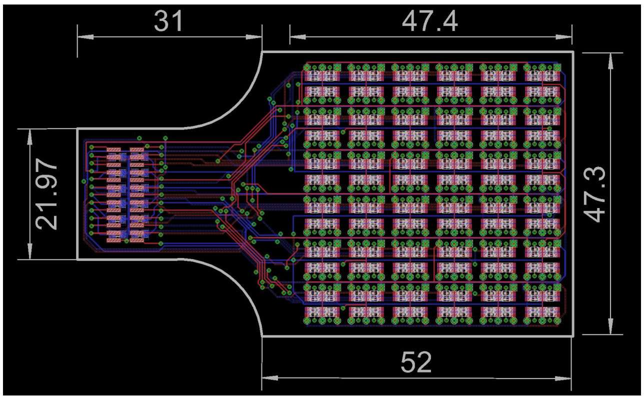


Figure 3.1: Routing of the flexible PCB (dimensions are expressed in millimeters).

number of resistors needed to convert the photocurrents into the voltages acquired by the A/D channels is reduced from the number of PTs to the number of A/D channels used during the scanning. Furthermore, these resistors can be mounted directly near the A/D channels, by avoiding to add components on the conformable part of the PCB. So, by adopting the presented scanning strategy, the routing of a whole conformable force/tactile sensor patch can be completed by using a flexible PCB with only 4 layers. Since no additional Integrated Circuits (ICs) with cumbersome package are used for the conditioning electronics, the types of components to mount on the flexible PCB, for each taxel, are only three and small enough to maintain a high flexibility of the PCB.

Design of the routing has been carried out by using a semi-automatic routing algorithm. The layout of the obtained PCB is reported in Fig. 3.1, including the dimensions. The active surface of the sensor patch (corresponding to the sensing elements) is about $47 \times 47\,\text{mm}^2$, while the 25 wires, needed to interrogate the patch, are carried to a standard connector positioned on the left side. Figure 3.2, on the top, reports a picture of the realized PCB, where the high flexibility is highlighted. The solution, after soldering of all the components, maintains a high flexibility that allows the sensor patch to be conformable to a surface with minimum curvature radius of about 3 cm, which is sufficient for covering robot surfaces such as arms, legs and torso.

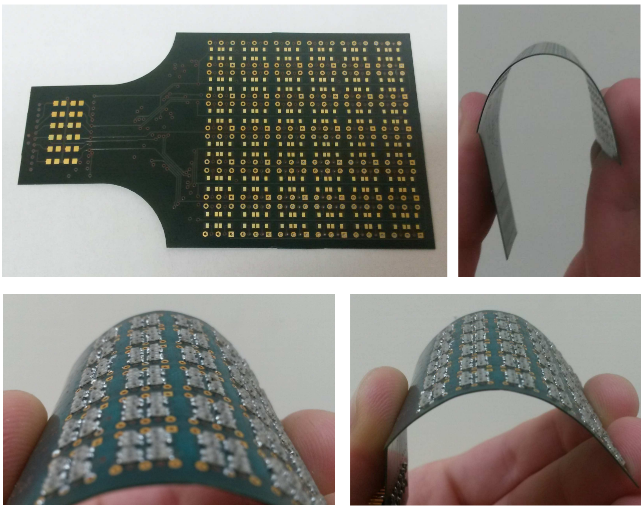


Figure 3.2: Pictures of the realized flexible PCB before (top) and after (bottom) soldering of optoelectronic components.

3.2 Integration of the conformable sensor on a robot arm

The first step towards the integration of the sensor patch on a robot link is the bonding of the deformable layer to the flexible PCB. In order to ensure correct operation of the sensor, the flexible PCB has to be conformed to the surface selected for the final assembly of the sensor patch before bonding of the silicone layer. Since the force estimation depends on the deformations of the silicone layer, if the flexible PCB were conformed to the target shape after bonding of the deformable layer, a residual strain would affect the sensing module causing a wrong force estimation. Therefore, reduction of these undesired deformations is achieved by first curving the PCB and then by bonding the silicone caps to it. To this aim, a mechanical support, designed on the basis of the shape of the surface selected for the final mounting, has to be realized. For the experiments presented in this thesis, the conformable sensor patch has to be mounted on a KUKA LWR 4+. The support has been designed on the basis of a 3D CAD model of the robotic arm and it has been 3D printed in ABS. The 3D CAD model provides a simple mechanism to extract a part of the robot surface where to collocate the distributed sensor. Figure 3.3 shows how the sensor patch support has been designed in the 3D CAD software. Starting from the CAD model of the whole arm (see Fig. 3.3-a), the area identified for the final mounting of the sensor patch is selected (see Fig. 3.3-b). From the selected surface the sensor patch support has been extruded as a filled solid (see Fig. 3.3-c). To obtain the final

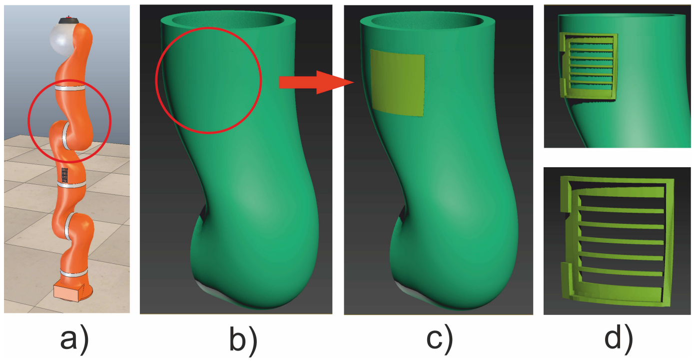


Figure 3.3: Sensor patch mechanical support: a) KUKA LWR 4+ 3D model, b) selected surface, c) extruded shape d) designed support.

sensor patch support, the filled solid has been completed with side edges designed to mechanically block the sensor patch on it and also several holes for inspection of the bottom side of the electronic layer (see Fig. 3.3-d). Once the flexible PCB has been fixed to the mechanical support by epoxy resin, the silicone caps are bonded to each sensing modules on the optoelectronic layer, by obtaining a fully assembled patch. Figure 3.4 shows some pictures of the conformable sensor prototype during the assembly phases.

Note that the final obtained conformed patch, differently from the rigid version, has not an uniform spatial resolution for the force detection, which depends from the local curvature. Let r_{flat} denote the sensing modules distance before the bonding of the PCB on the conformed mechanical support, that is equal to the spatial resolution of the rigid version, namely 7.4 mm. Moreover, let R denote the local curvature radius of the mechanical support and h_c the height of the silicone caps bonded on the PCB. Then, the spatial resolution of the conformed sensor patch locally varies in the range $\pm(h_c * r_{flat})/R$ from the flat value r_{flat}. In particular, by considering a curvature radius of 3 cm, being $h_c = 4$ mm and $r_{flat} = 7.4$ mm, it is possible to estimate that the spatial resolution of the conformed sensor patch can locally vary of ±1 mm from the spatial resolution of the rigid version, by resulting in a non uniform resolution equal to 7.4 ± 1 mm.

3.3 Conformable sensor calibration

In the following, an extension of the calibration procedure originally presented in 2.6.1 is proposed to handle the complex shape of the tactile sensor patch. In fact, to provide

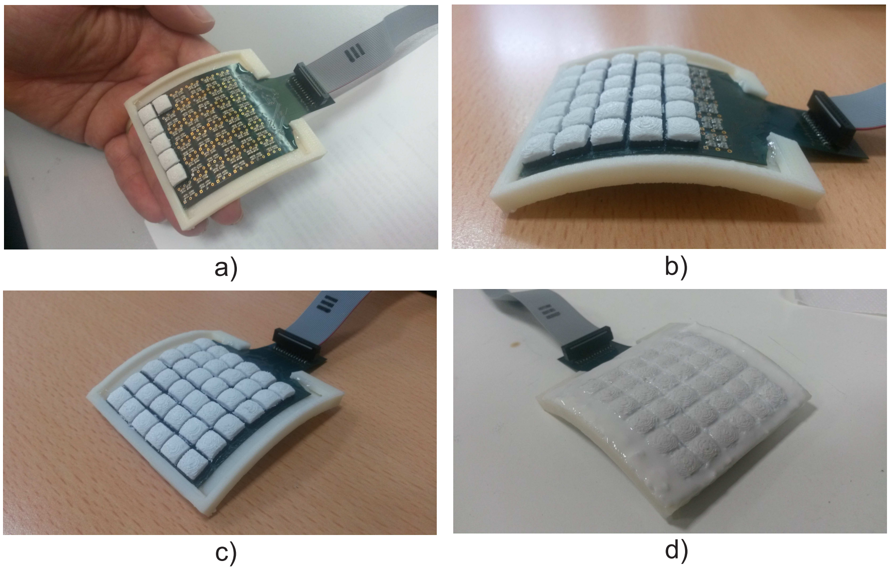


Figure 3.4: Conformable sensor patch during assembly phases: a) perspective view, b) side view, c) perspective view of the complete sensor prototype, d) completed sensor prototype.

the user with a contact force vector and contact position expressed in a single reference frame fixed to the sensor, the calibration procedure has to take into account the orientation of suitable reference frames attached to each module with respect to the given sensor frame. Furthermore, a comparison between two calibration algorithms with different computational complexity will be reported: one based on an Artificial Neural Network (ANN) and the other one based on a simple linear mapping.

The calibration procedure has been implemented by using, as reference sensor, a six-axis load cell, manufactured by ATI. The model used is the FTD-Nano-17, with a measurement range equal to ± 12 N and ± 17 N for horizontal and vertical force components, respectively. The measurement range for all torque components is equal to ± 120 Nmm. First of all, in order to install the conformable sensor patch on the reference sensor a second support with the same shape of the one realized for the robot arm has been 3D printed. Figure 3.5 shows the calibration setup with the sensor patch mounted on the mechanical adapter, fixed to the reference sensor. The computation of the calibration parameters for each sensing module requires that the axis of the reference frame of the ATI sensor and the axis of the reference frame of the sensing module, on which the external force is applied, are aligned. Let Σ_{s_i} be the reference frame of the ith sensing module and Σ_{ATI} the reference frame of the ATI sensor. On the basis of the 3D CAD models, the rotation matrix $\mathbf{R}_{s_i}^{ATI}$ for each sensing module

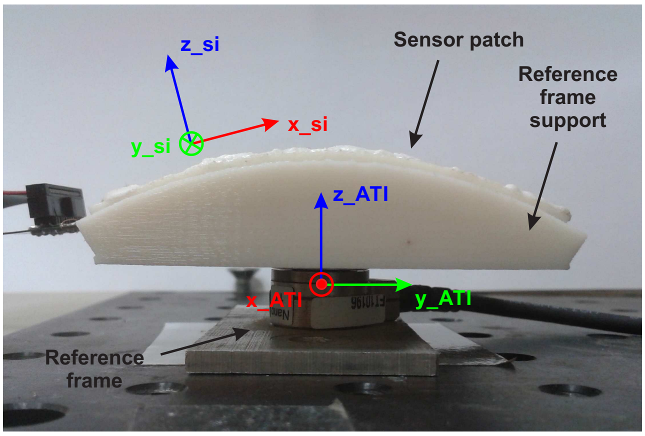


Figure 3.5: Calibration setup.

can be defined. Then, with the choice of the reference frames reported in Fig. 3.5, the following equation, that allows to rotate the force vector measured by the ATI sensor in the ith sensing module reference frame, can be written:

$$\mathbf{f}^{s_i} = \left(\mathbf{R}_{s_i}^{ATI}\right)^{\mathrm{T}} \mathbf{f}^{ATI} \tag{3.1}$$

where $\mathbf{f}^{ATI} = (f_x^{ATI} \; f_y^{ATI} \; f_z^{ATI})^{\mathrm{T}}$ is the vector that contains the force components expressed in the ATI sensor frame, $\mathbf{f}^{s_i} = (f_x^{s_i} \; f_y^{s_i} \; f_z^{s_i})^{\mathrm{T}}$ is the vector that contains the force components expressed in the ith sensing module frame and the rotation matrix $\mathbf{R}_{s_i}^{ATI}$ is defined as

$$\mathbf{R}_{s_i}^{ATI} = \mathbf{R}_z\,(\pi/2)\,\mathbf{R}_y\,(\theta_i)\,, \tag{3.2}$$

where θ_i is the angle between the y_{ATI} e x_{s_i} axes, positively defined for counter-clockwise rotation about the y_{ATI} axis, that can be extracted from the 3D CAD model of the mechanical support.

To collect data for the calibration of the sensor patch, an operator carried out at least two experiments for each sensing module, by using a stiff flat object to apply different external forces. In particular, the operator manually interacted with each sensing module being careful to apply forces with components along all the directions, defined by the frame of the sensing module itself, and with amplitudes varying from the value 0 N to the sensor full scale. For each experiment, all the voltage variations $\mathbf{V}^{s_i} = (V_1^{s_i} \; V_2^{s_i} \; V_3^{s_i} \; V_4^{s_i})^{\mathrm{T}}$ measured by the PTs belonging to the sensing module, and the force components $\mathbf{f}^{ATI}$ measured by the reference sensor have been acquired. Then, for each experiment the measured $\mathbf{f}^{ATI}$ have been rotated according to (3.1) in order to obtain the correct force components $\mathbf{f}^{s_i}$. The collected data have been divided in two sets: a training set used to identify the calibration parameters and a validation set used to validate the accuracy of the calibration. With these data two phenomenological models have been calibrated and compared.

The first model considers the force components as a linear combination of the measured voltages as

$$f_x^{s_i} = (\mathbf{k}_x^{s_i})^{\mathrm{T}} \mathbf{V}^{s_i} \tag{3.3}$$

$$f_y^{s_i} = (\mathbf{k}_y^{s_i})^{\mathrm{T}} \mathbf{V}^{s_i} \tag{3.4}$$

$$f_z^{s_i} = (\mathbf{k}_z^{s_i})^{\mathrm{T}} \mathbf{g}(\mathbf{V}^{s_i}), \tag{3.5}$$

where the vector function $\mathbf{g}(\cdot)$ is the absolute value applied to each component of the vector $\mathbf{V}^{s_i}$ and the three 4×1 vectors $\mathbf{k}_x^{s_i}$, $\mathbf{k}_y^{s_i}$ and $\mathbf{k}_z^{s_i}$ represent the calibration parameters to identify. Starting from the training set data, these parameters have been identified with a simple least square algorithm by inverting Eqs. (3.3), (3.4) and (3.5), respectively, written for each point of the training data set. Then, the identified parameters have been used to evaluate the accuracy of the calibration phase, by computing the estimated force components for the validation data set, as

$$\widehat{f_x^{s_i}} = (\mathbf{k}_x^{s_i})^{\mathrm{T}} \mathbf{V}^{s_i} \tag{3.6}$$

$$\widehat{f_y^{s_i}} = (\mathbf{k}_y^{s_i})^{\mathrm{T}} \mathbf{V}^{s_i} \tag{3.7}$$

$$\widehat{f_z^{s_i}} = (\mathbf{k}_z^{s_i})^{\mathrm{T}} \mathbf{g}(\mathbf{V}^{s_i}), \tag{3.8}$$

where $\widehat{f_x^{s_i}}$, $\widehat{f_y^{s_i}}$ and $\widehat{f_z^{s_i}}$ are the estimated force values for the *i*th sensing module.

As second model a $\mathbf{f}_{\mathbf{NN}}^{s_i}(\cdot)$ ANN operator has been considered, namely, for the *i*th sensing module

$$\mathbf{f}^{s_i} = \mathbf{f}_{\mathbf{NN}}^{s_i}(\mathbf{V}^{s_i}). \tag{3.9}$$

In particular, a standard two-layer feed-forward neural network, trained with the Levenberg-Marquardt method, has been used to fit the training data. Different number of neurons of the hidden layer for the ANN have been tested, and the solution providing a good trade-off between the training error and complexity has been found with 6 neurons. Thus, for each sensing module, an ANN with these characteristics has been trained and then its performance has been evaluated by using the corresponding validation data set.

The force components estimated by using both models have been compared to the measured ones. In Fig. 3.6, as an example, for a sensing module the estimated

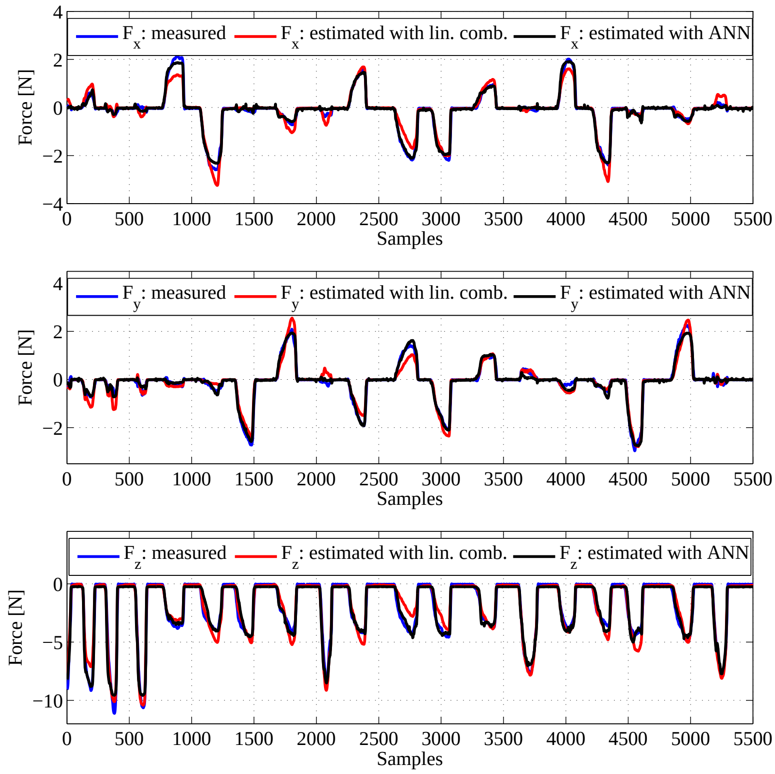


Figure 3.6: Force components for a training data set: X component (top), Y component (middle), Z component (bottom).

force components are compared to the actual force components measured by the reference sensor, for a training data set, just to verify the convergence of the training algorithm. Instead, Fig. 3.7 shows the accuracy of the calibration, by reporting, for the same sensing module, the estimated and the measured force components by using the validation data set. The estimation appears satisfactory for all force components with both calibration models.

In order to evaluate the calibration performance in a quantitative way, a synthetic index has been computed for each force component and for both models. The quality index is defined as

$$e_k = \frac{1}{N}\sum_{i=0}^{N}\left|f_k^{s_i}(i) - \widehat{f_k^{s_i}}(i)\right|, \tag{3.10}$$

where $k = x, y, z$ indicates the force component, N is the number of samples, $f_k^{s_i}$ is the force component k of the ith sensing element and $\widehat{f_k^{s_i}}$ the corresponding estimated value. Figures 3.8 and 3.9 report the mean errors computed as in Eq. (3.10) and evalu-

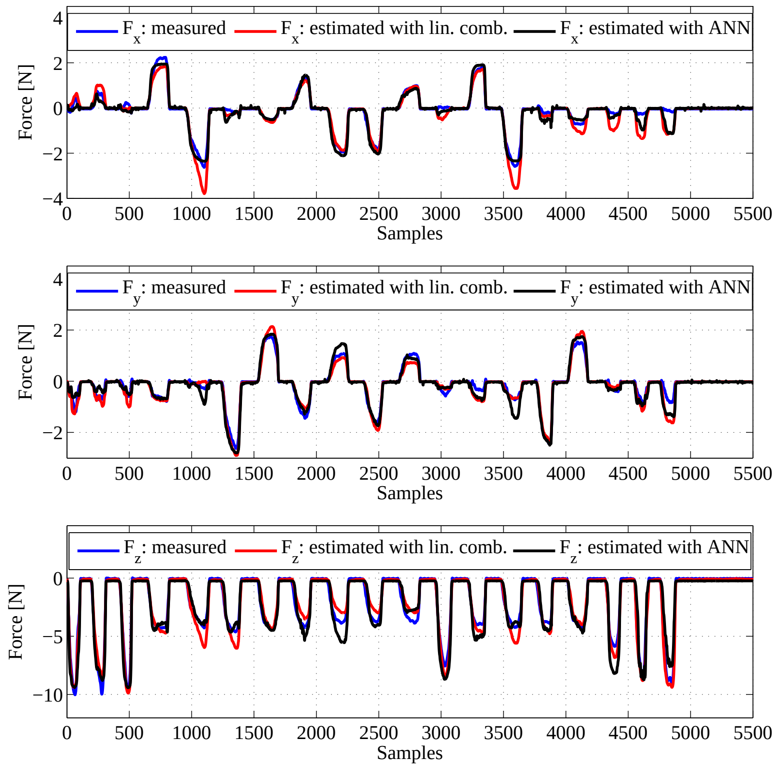


Figure 3.7: Force components for a validation data set: X component (top), Y component (middle), Z component (bottom).

ated for the entire 6×6 sensor matrix considering the two calibration approaches. The use of the calibration approach based on the ANN model provides a better accuracy for the shear force components for most of the sensing elements; but, it introduces a greater mean error for the normal force component. Moreover, the better accuracy of the ANN model is not enough to justify its computational complexity compared to the linear model. By considering the full scales of each sensing element, which are ±4 N for the shear components and 14 N for the normal component, the maximum mean error is less than 7.5%.

Up to now, all modules have been calibrated one by one. Instead, by using the same calibration parameters for all sensing modules, estimated on a module located in the middle of the patch, a degradation of the accuracy is expected. To quantify it, the linear model is adopted for all sensing modules and the results are reported in Fig. 3.10. The maximum mean error is less than 15%, which is the double of the former calibration approach, but using a significantly less time consuming calibration procedure. It is evident that for whole body applications, a good trade off can be: the

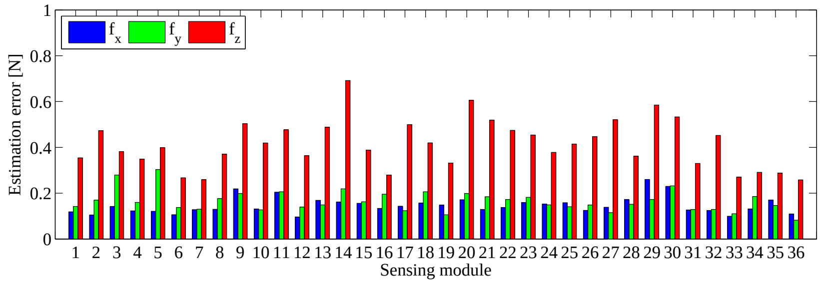


Figure 3.8: Estimation errors with calibration of each sensing module using the Linear Combination approach.

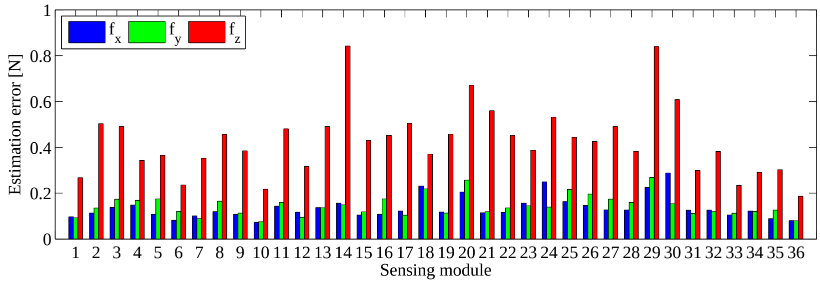


Figure 3.9: Estimation errors with calibration of each sensing module using the ANN approach.

use of the first strategy (more accurate and time-consuming) for parts (e.g., arms, hands) where the interactions with the environment are many and intentional; the use of the second approach (less accurate but very time-saving) for parts (e.g., torso, legs) where typically only unintentional interactions can occur. Table 3.1 provides a comparison of the three calibration approaches. It reports the mean estimation error and the standard deviation for each calibration approach computed considering the 36 sensing modules. The absolute accuracy of the shear components is less than the one of the normal component, but taking into account the force range previously described, the relative accuracy results to be equivalent for all the force components.

The process needed to provide conformability to the described sensor has been carefully designed and tuned in order to do not introduce significant alterations to the sensor characteristics, already studied for the rigid prototype of the tactile sensor. Properties such as repeatability, hysteresis, time response and signal to noise ratio have been again analyzed with the same methodology reported in Section 2.6.1 and they result to be very close to those obtained for the rigid sensor prototype:

- sensitivity: $\approx 0.2\,\text{N}$

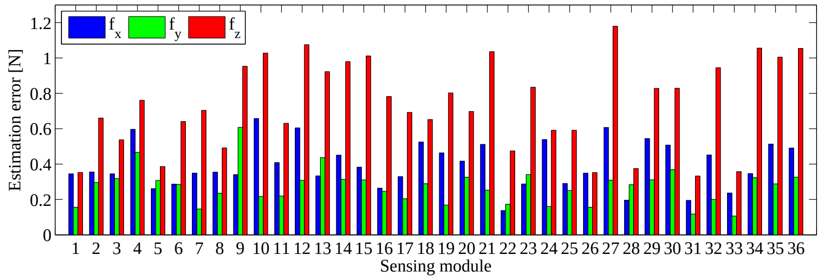


Figure 3.10: Estimation errors with the same calibration matrix for all sensing modules using the Linear Combination approach.

Table 3.1: Comparison of the three calibration approaches in terms of mean estimation error and standard deviation.

Approach	Mean error [N]			Standard deviation [N]		
	f_x	f_y	f_z	f_x	f_y	f_z
Lin. comb.	0.1469	0.1657	0.4139	0.0363	0.0460	0.1052
ANN	0.1357	0.1445	0.4290	0.0485	0.0457	0.1497
Lin. comb. (one calib. matrix)	0.3964	0.2728	0.7388	0.1293	0.1011	0.2483

- repeatability error: $\approx$ 6%
- hysteresis error: $\approx$ 10%
- response time: ≈ 0.001 s

An important remark concerns the hysteresis error. Its limited value allows to use the sensor information without introducing further compensation algorithms differently from other solutions based on soft foam materials [77].

Part II

Control: manual guidance, intuitive programming, collision detection and reaction

CHAPTER 4

The force/tactile sensor as Human-Machine Interface

How users will interact with robots of the future came out of the factories? This is still an open question, certainly not through a keyboard and a mouse or through a heavy teach pendant. Someone say that speech will be the preferred interaction modality, but some decades ago this was envisioned for the personal computers too, and this did not happen. While, nowadays touchpads are by far the most widespread interface of both PC's and other digital devices, from smartphones and tablets to car on board computers. Tactile interaction is becoming the preferred way to provide commands to our digital assistants and ask them to do something for us. Imagine that such a modality were available also for interacting with robots, then it would be quite natural to command robots by simply touching them, not only for teaching them new movements but also for asking them to carry out the task we need in a certain moment. Robots are 6-dimensional machines that can move themselves and objects in their world, so simple touches might not be enough to teach them complex movements or to ask them the large variety of tasks they are able to perform. More complex haptic gestures could be needed and motion commands in specific directions in space are surely needed to learn new movements.

4.1 What is a human-machine interface

To the aim of a close collaboration between humans and robots the use of human-machine interfaces (HMI) is exploited to enable the perception of the users, including many of their important communication cues, such as speech, gestures, head orientation, and to allow robust interaction between the human and the robot. A huge number of HMI solutions exist and most of them exploit more than one perception system (multi-modal perception). A large portion is constituted by vision-based system [78, 79, 80, 81, 82, 83] in which the main drawbacks are the background variability, the bad lighting conditions and the computational time. Computer video camera, Microsoft Kinect [84], RGB-D camera, is used to detect human motions, i.e., face and hand gesture, head orientation, arm posture. Another smaller part is constituted by more complex systems in which different perceptional and communicative cues are fused together in order to build multi-modal dialogue components that enable the robot to engage in task-oriented dialogue with their users in a more natural way. They include systems for spontaneous speech recognition, multi-modal dialogue processing, and visual perception of a user, e.g., localization, tracking and identification of the face and hand of the user, recognition of pointing gestures [85, 86, 87, 88]. However, all the mentioned approaches use different sensors, i.e., vision camera or Kinect, microphone, IMUs and they use computationally expensive frameworks to fuse the data acquired from all the sensors and then take a decision. So, integrating such systems in a real-time task in which there is a physical collaboration between the human and the machine, represents, by now, an important challenge. An intuitive and very fast way for interaction with people is offered by the tactile interaction. Haptic cues can usually be interpreted very quickly as demonstrated in [89, 90] and tactile sensor can be used to classify different types of touch [91, 92]. The KUKA LWR 4+ has been used in [93] for executing complex tasks in collaboration with humans; switching between task segments and control modalities has been implemented through simple haptic gestures that the user had to apply to the last robot link, e.g., pushing or pulling in a certain Cartesian direction. Such approach has only a limited number of gestures due to the limited accuracy in the estimation of contact force vector based on the sole residuals. In case a distributed tactile map were available, the number of haptic gestures could be greatly enlarged owing to the richness of the captured information, but still with a fast response. Imagine if a notebook-like touch pad would be available on one or more robot links, collaboration with a robot could become very intuitive and complex at the same time. Imagine also that the same device is able to provide contact force vector estimates on many points, then it could be exploited also to move the robot links not only for programming but also during task execution to dynamically reconfigure a redundant arm in a more natural and comfortable posture for the user. Naturally, a standard touch pad would not be suitable for mounting on a robot link since the rigid device would be damaged quite easily by the first accidental collision. Also capacitive touch pads are not easily conformable to curved surfaces and cannot estimate contact forces.

This Section shows how the force/tactile sensor can be actually used as an input device for sending commands to the robot, e.g., commands for changing control modality or selecting a task to execute. Different recognition methods, e.g., Finite State Machine, Artificial Neural Network [94, 95], Hidden Markov Model [96, 97], Features extrapolation [98], that differ in complexity and performance are described in the literature [99], but most of them are applied to inertial and camera systems.

In this thesis three algorithms have been designed taking into account the sensor transduction principle and the sensor data collection. The first one is used to recognize gestures that are applied with static contact on the sensor surface, while, the other two methods are used to recognize dynamic and more complex touch gestures. The sensor provides 288 bytes corresponding to the voltage signals of the 144 taxels. Starting from the idea behind the classic features extrapolation techniques, different features are computed with the sensor raw data according to the complexity of the gestures to recognize. Moreover, a suitable preprocessing stage and a classifier have been proposed considering the specific feature adopted for each recognition algorithm.

4.2 Static gesture recognition

The first method is presented to simply show how the sensor information can be exploited to recognize tactile gestures using a simple algorithm. The sensor signals are organized in a 12×12 matrix corresponding to the sensor pressure map. The latter is used as recognition feature. Since only a small set of gestures has been considered, a simple algorithm such as the dot product-based recognition [100] is used to recognize static tactile gestures. This can be achieved by defining an elementary set of tactile gestures (codebook), i.e., a set of modalities to touch the sensor patch by a human hand. A part of the selected set of tactile gestures is reported in the left side of Fig. 4.1.

A tactile map corresponds to each tactile gesture that can be represented with a 12×12 matrix constituted by the signals from all the taxels, thus a recognition can be performed by resorting to algorithms typically used for image processing applications. In fact, for each time instant a static representation of the tactile map, i.e, an image of 12×12 pixels can be obtained by properly pre-processing the acquired raw data. In a pre-elaboration stage an image of boolean values (“0” and “1”) is obtained by thresholding the sensor voltage signals. Moreover, a bounding box that contains the detected gesture, depicted in the bit-map image as a group of “1” elements, is identified and it is translated in the upper-left corner of the bit-map image. The elaborated gesture can now be used in the recognition process. Let $\mathbf{x}_i$ be the 144×1 vector that contains the 12 columns of the bit-map image corresponding to the ith gesture to be recognized and $\mathbf{y}$ the 144×1 vector that contains the columns of the bit-map image corresponding to the acquired tactile image. The dot product is calculated as in Eq. (4.1), and the result provides a likelihood measure between the vectors $\mathbf{x}_i$ and $\mathbf{y}$, i.e.,

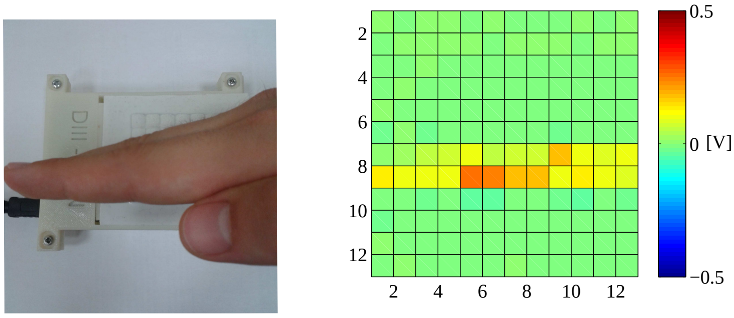
$$s_i = \sum_{j=1}^{144} x_{i_j} y_j \tag{4.1}$$

The higher s_i, the closer (in the Hamming sense) the two vectors are and the more alike the corresponding gestures are. The dot product-based recognition is by far the fastest and easiest gesture recognition method and it is able to recognize letters and digits. However, this method is not universal, it will often have a problem separating circles and squares, but this is the price for simplicity and speed. In Figure 4.1 four gestures and the corresponding tactile maps are shown. Gestures like vertical line, horizontal line, line along the main and secondary diagonal are considered. It is evident how the raw data provide a complete information about the contact that occurs on the deformable layer of the sensor.

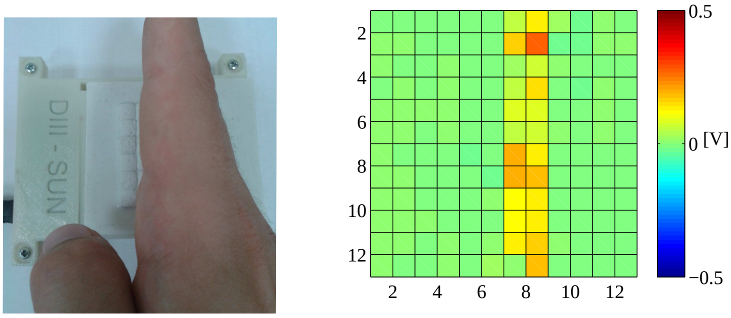
4.3 Dynamic gesture recognition

Two different methods used to recognize dynamic gestures are presented. For the first one, the pressure map obtained reorganizing the 144 tactile sensor signals in a 12×12 matrix has been chosen as recognition feature, while, the second one exploits the information about the force contact point in order to recognize more complex gestures. Figure 4.2 reports a scheme that highlights the training pipeline (right branch) and the recognition pipeline (left branch). The sensor starts acquiring the gesture applied by the user as soon as a contact on the deformable layer is detected. The gesture data are collected until the contact ends. In order to make the recognition process independent from the particular sensor contact area on which the gesture is applied, the data pass through a pre-elaboration/normalization stage. The preprocessed gesture is, then, compared to each gesture contained in a training set, which is preliminarily collected. The gesture selection is made on the basis of a maximum likelihood criteria. The pre-elaboration/normalization phase and the error index computation depend on the specific recognition feature used in each implemented method.

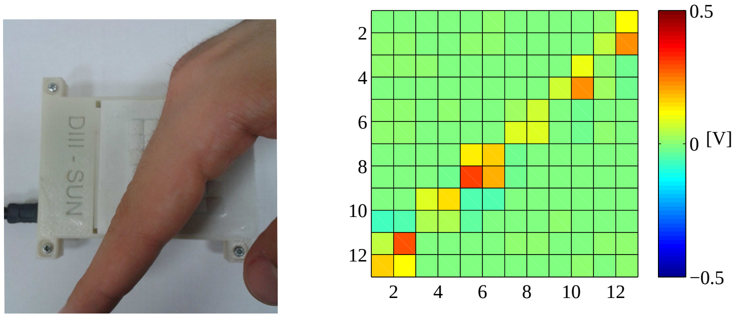
For the sake of completeness, the Nearest-Neighbor Interpolation algorithm [101]



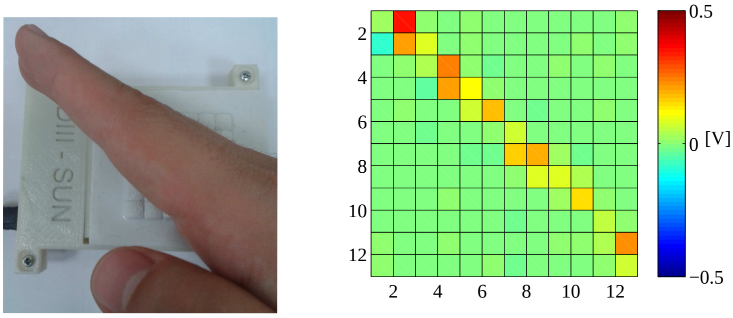
(a) Horizontal line.



(b) Vertical line.



(c) Main diagonal.



(d) Secondary diagonal.

Figure 4.1: Applied gestures and corresponding tactile maps.

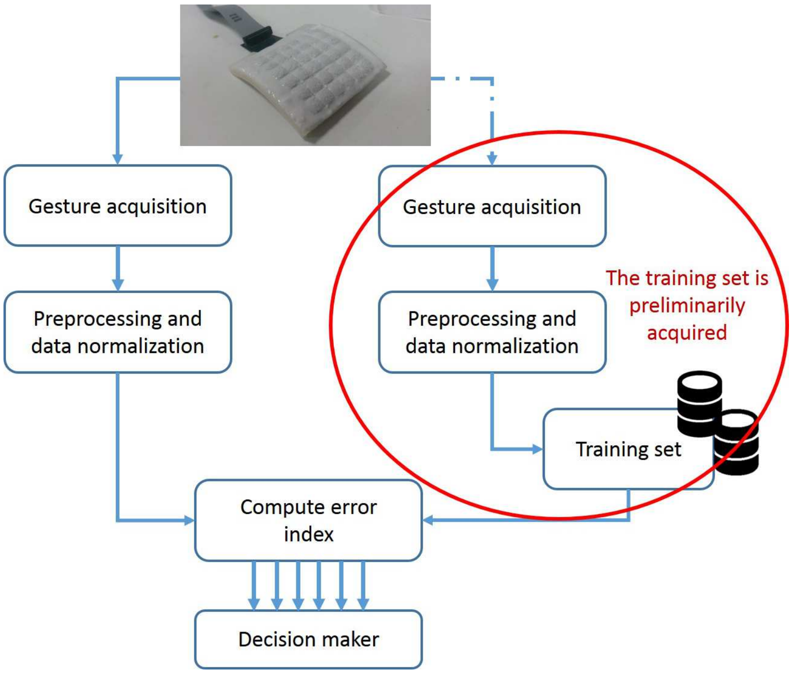


Figure 4.2: Schematization of training and recognition pipelines for haptic gesture recognition.

(NNA) used in the recognition methods is briefly recalled below. Let consider a generic interpolation algorithm in the following linear form

$$f(\mathbf{x}) = \sum_{\mathbf{z}\in\mathbb{Z}^q} f_k \phi(\mathbf{x} - \mathbf{k}), \qquad \forall \mathbf{x} = \left(x_1, x_2, \ldots, x_q\right) \in \mathbb{R}^q, \tag{4.2}$$

where an interpolated value $f(\mathbf{x})$ at some coordinate $\mathbf{x}$ in a space of dimension q is expressed as a linear combination of the samples f_k evaluated at integer coordinates $\mathbf{k} = \left(k_1, k_2, \ldots, k_q\right) \in \mathbb{Z}^q$, being the value of the function $\phi(\mathbf{x} - \mathbf{k})$ the interpolation weight. Typical values of the space dimension correspond to bidimensional images (2D), with $q = 2$ and tridimensional volumes (3D), with $q = 3$. In the specific case when all coordinates of $\mathbf{x} = \mathbf{k}_0$ are integer, the following formulation can be considered

$$f_{k_0} = \sum_{\mathbf{z}\in\mathbb{Z}^q} f_k \phi(\mathbf{k}_0 - \mathbf{k}), \qquad \forall \mathbf{k}_0 \in \mathbb{Z}^q, \tag{4.3}$$

which represents a discrete convolution. On the basis of the specific synthesis function ϕ used in the interpolation process, several interpolation algorithms that differ in complexity and accuracy can be identified [102]. The Nearest-Neighbor algorithm is the simplest interpolation technique from a computational point of view used in image processing for image scaling. The synthesis function associated to it is the simplest of all, since it is made of a square pulse. For simplicity its expression for a space of dimension $d = 1$ is reported

$$\phi(x) = \begin{cases} 1, & \text{if } 0 \leq |x| < 0.5 \\ 0, & \text{if } 0.5 \leq |x| \end{cases} . \tag{4.4}$$

The main interest of this synthesis function is its simplicity, which results in the most efficient of all implementations. In fact, for any coordinate $\mathbf{x}$ where it is desired to compute the value of the interpolated function f, there is only one sample f_k that contributes, no matter how many dimensions q are involved. The price to pay is a severe loss of quality. The algorithm performs image magnification by pixel replication and image reduction by sparse point sampling, and it derives its primary use as a tool for real-time magnification.

4.3.1 Map-based recognition algorithm

For the first recognition algorithm the sensor tactile map, suitably adapted and elaborated, represents the recognition feature. As described in Section 4.2, to each tactile gesture corresponds a tactile map that can be represented with a 12×12 matrix constituted by the signals from all the taxels. For each time instant a static representation of the tactile map, i.e, an image of 12×12 pixels can be obtained by properly processing the acquired raw data and in a preliminary stage an image of "0" and "1" values is obtained by thresholding the sensor signals. During the gesture acquisition, maps obtained in each time instant are element-wise multiplied. At the end, a representation, in terms of an image of 12×12 pixels, of the route traced by the user finger on the contact surface of the sensor is available. Given that the gesture could be generally traced anywhere on the available sensor contact area, a pre-elaboration/normalization phase is necessary so that the recognition algorithm can properly work independently from that area. Starting from the map provided at the end of the acquisition phase, a bounding box that contains the detected gesture (see Fig. 4.3), depicted in the bit-map image as a group of "1" elements, is identified. The reduced image, which represents the detected gesture, is rescaled in order to obtain a new image of 12×12 pixels by applying the NNA. The elaborated gesture can now be used in the recognition process. The decision is made by evaluating $n \times m$ error indexes obtained by comparing the elaborated gesture to the n gestures, which are preliminarily acquired for

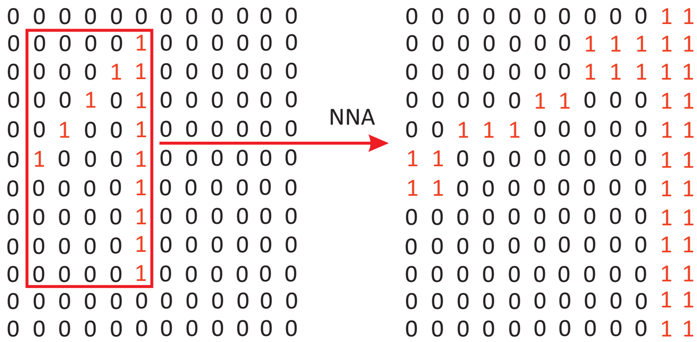


Figure 4.3: Pre-elaboration/Normalization phase adopted for the first recognition method.

m times, collected in the training set and choosing the gesture corresponding to the lowest error index. The error indexes are computed according to the Hamming distance[3] between the bit-map matrices. The described algorithm has been summarized with the pseudocode Algorithm 3.

Algorithm 3 Pseudocode of the Map-based recognition algorithm.

Require: 144 tactile sensor signals
Ensure: Recognized gesture

initialization
while TRUE **do**
    TactileMap=extractTactileMap(sensorSignals)
    TactileMap+=TactileMap (element-wise sum)
    if sensorNotTouched **then**
        ClippedTactileMap=getBoundingBox(TactileMap)
        scaledTactileMap=NNA(ClippedTactileMap)
        for each *i*-th gesture in the codebook **do**
            H_i=compare(scaledTactileMap,codebook$_i$) (in terms of Hamming distance)
        end for
        makeDecision(H)
        clear(TactileMap)
    end if
end while

---

[3]The Hamming distance between two matrices of equal size is the number of positions at which the corresponding elements are different.

4.3.2 Centroid-based recognition algorithm

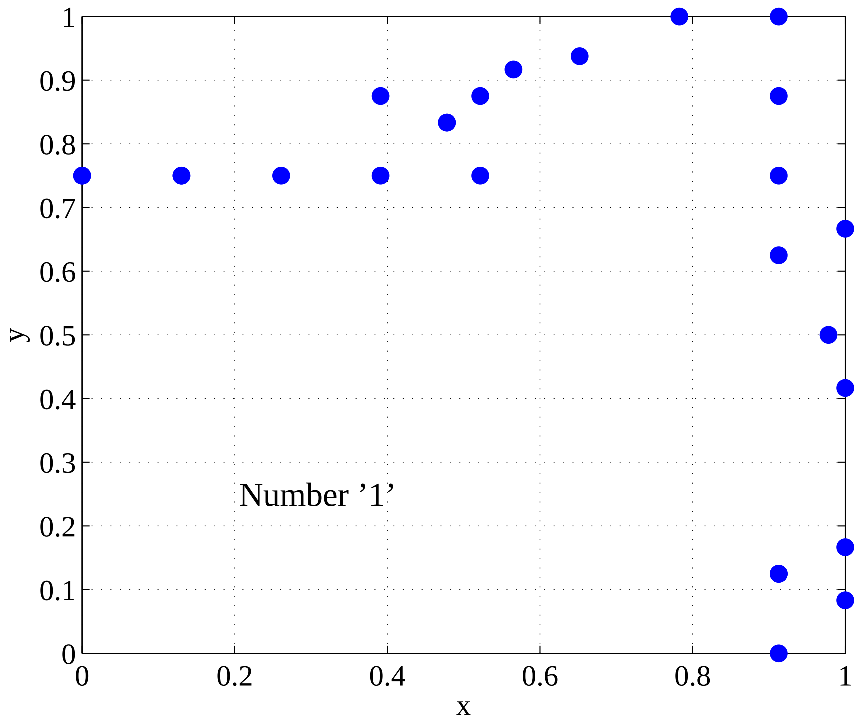
The method of the previous section is able to recognize in an efficient way different touch gestures, e.g., numbers, chars, geometric primitives. However, the use of a bit-map image as recognition feature does not allow to discriminate the direction with which the gesture is made, e.g., a line from left to right and vice versa. This second method intends to overcome this disadvantage exploiting the force contact point provided by the sensor, which brings information concerning both the area on which the touch gesture is applied and the direction with which it is traced. By properly processing the sensor raw data, it is possible to estimate the spatial coordinate of the force contact point w.r.t. a reference frame fixed on the tactile sensor (refer to Sec. 2.6.2 for more details). Let define $\mathbf{g}_x$ and $\mathbf{g}_y$ as the vectors that contain the x and y components of the contact point, respectively, whose size depends on the time needed by the user to trace the gesture on the sensor surface. The couple $\left(\mathbf{g}_x, \mathbf{g}_y\right)$ represents the gesture feature. The normalization stage foresees two successive feature elaborations. First, the vectors $\mathbf{g}_x$ and $\mathbf{g}_y$ are resampled exploiting the NNA in order to produce a time-independent gesture feature, $\left(\mathbf{g}_x^t, \mathbf{g}_y^t\right)$. The latter is, then, normalized to obtain a gesture feature independent from the area of the sensor on which the gesture is traced, i.e.,

$$\overline{\mathbf{g}}_i = \frac{\mathbf{g}_i^t - \min \mathbf{g}_i^t}{\max \mathbf{g}_i^t - \min \mathbf{g}_i^t} \text{ with } i = x, y. \tag{4.5}$$

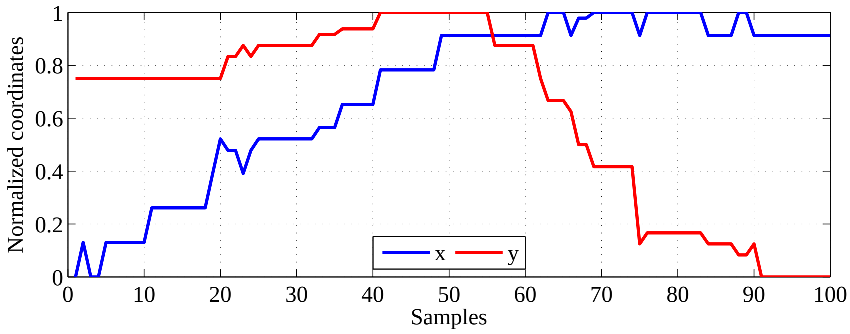
Figure 4.4 shows an example of a touch gesture feature after the normalization stage. Figure 4.4(a) reports the touch gesture traced on the sensor surface, e.g., the number *1*, while Figure 4.4(b) reports the gesture feature considered during the recognition process. As in the map-based method, the decision is made by evaluating $n \times m$ error indexes obtained by comparing the elaborated gesture to the n gestures, which are preliminarily acquired for m times, collected in the training set and choosing the gesture corresponding to the lower error index. In this case, the error indexes are calculated as the Euclidean distance between the feature of the acquired gesture $\left(\overline{\mathbf{g}}_x, \overline{\mathbf{g}}_y\right)$ and the $n{\times}m$ features contained in the codebook $\left(\hat{\mathbf{g}}_x^j, \hat{\mathbf{g}}_y^j\right)$ with $j = 1, \ldots, n{\times}m$, i.e.,

$$E_{d_j} = \frac{1}{2}\left[\sqrt{\sum_{k=1}^{l}\left(\overline{\mathbf{g}}_x(k) - \hat{\mathbf{g}}_x^j(k)\right)^2} + \sqrt{\sum_{k=1}^{l}\left(\overline{\mathbf{g}}_y(k) - \hat{\mathbf{g}}_y^j(k)\right)^2}\right] \tag{4.6}$$

where $j = 1, \ldots, n{\times}m$ and l is the number of elements of the vectors $\overline{\mathbf{g}}_x$ and $\overline{\mathbf{g}}_y$ chosen in the resampling phase.



(a) Gesture: number *1*.



(b) Gesture feature: contact point *x*-,*y*-coordinates.

Figure 4.4: Example of a touch gesture and of the corresponding feature after the normalization step.

Table 4.1: Confusion matrix of the dot product-based gesture recognition algorithm.

Recognized/Traced	Horizontal line	Vertical line	Main diagonal	Secondary diagonal
Horizontal line	86.6%	0.0%	6.7%	6.7%
Vertical line	0.0%	83.3%	8.3%	8.3%
Main diagonal	6.7%	8.3%	85.0%	0.0%
Secondary diagonal	6.7%	8.3%	0.0%	86.6%

The described algorithm has been summarized with the pseudocode Algorithm 4.

Algorithm 4 Pseudocode of the Centroid-based recognition algorithm.

Require: 144 tactile sensor signals

Ensure: Recognized gesture

initialization
while TRUE **do**
    cp=extractContactPoint(sensorSignals) (as described in [70])
    if sensorNotTouched **then**
        resampledCP=NNA(cp)
        normalizedCP=normalize(resampledCP) (as defined in Eq.(4.5))
        for each *i*-th gesture in the codebook **do**
            E_i=compare(normalizedCP,codebook$_i$) (as defined in Eq.(4.6))
        end for
        makeDecision(E)
        clear(cp)
    end if
end while

4.4 Algorithm assessment

In order to assess the recognition algorithms a set of 30 trials for each gesture have been performed by 20 performers and the performance is assessed in terms of the recognition rate, namely ratio between the number of correctly recognized gestures and the total number of trials.

Table 4.1 reports a confusion matrix for the dot product-based algorithm used to recognize static touch gesture: a recognition rate higher than 80% has been obtained for each gesture. The algorithm results to be extremely simple, but it is able

Table 4.2: Confusion matrix of the map-based algorithm.

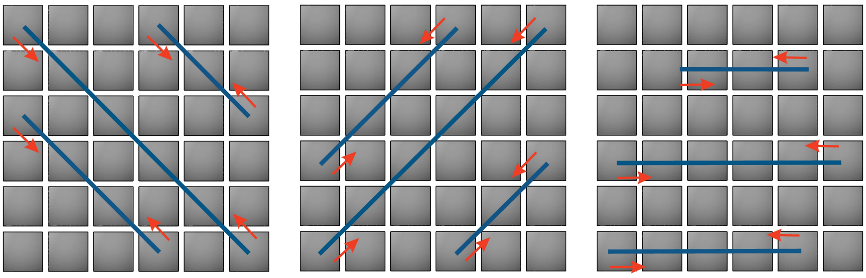
Recognized/Traced	a	b	c	d	e
a	95.3%	3.3%	0.0%	2.3%	2.3%
b	3.3%	93.3%	2.7%	6.7%	9.7%
c	0.0%	2.3%	95.3%	5.7%	2.7%
d	1.3%	1.0%	0.7%	85.3%	2.3%
e	0.0%	0.0%	1.3%	0.0%	83.0%

Table 4.3: Confusion matrix of the centroid-based algorithm.

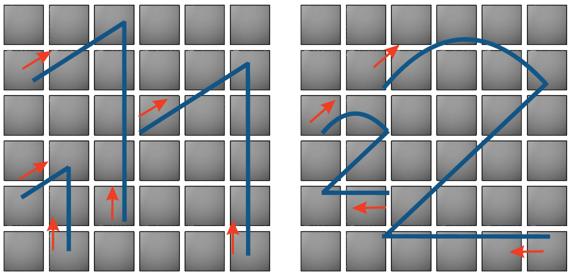
Recognized/Traced	a	b	c	d	e
a	100.0%	0.0%	11.3%	0.0%	2.3%
b	0.0%	100.0%	0.0%	11.3%	0.3%
c	0.0%	0.0%	86.0%	0.0%	2.7%
d	0.0%	0.0%	0.0%	88.0%	0.0%
e	0.3%	0.0%	2.7%	0.0%	94.7%

to recognize only simple gestures applied with static contacts on the sensor surface. This characteristic represents a critical disadvantage that limits the touch gestures applicable with a human hand to the ones showed in Fig. 4.1.

Table 4.2 and Fig. 4.5 report the results for the map-based algorithm used in the recognition of the dynamic touch gestures in terms of recognition rate and the set of analyzed gestures, respectively. Let define the five analyzed gesture as *a* for the diagonal, *b* for the secondary diagonal, *c* for the horizontal line, *d* for the number 1 and *e* for the number 2. As said previously, the decision making is not influenced by the direction with which the gesture is traced on the sensor surface, so, a diagonal traced from the upper-left corner to the bottom-right corner of the sensor is equivalent to a diagonal traced from the right-bottom corner to the upper-left corner and both are recognized as gesture *a*. In order to demonstrate it, the performers have executed the tests tracing the gestures into the two directions as shown in Fig. 4.5 by the red arrows. The same analysis has been carried out for the centroid-based algorithm. Table 4.3 and Fig. 4.6 report the confusion matrix and the analyzed gestures. Let be *a* the diagonal traced from the upper-left corner to the bottom-right corner, *b* the diagonal traced from the bottom-right corner to the upper-left corner, *c* the horizontal line traced from left to right, *d* the horizontal line traced from right to left and *e* the

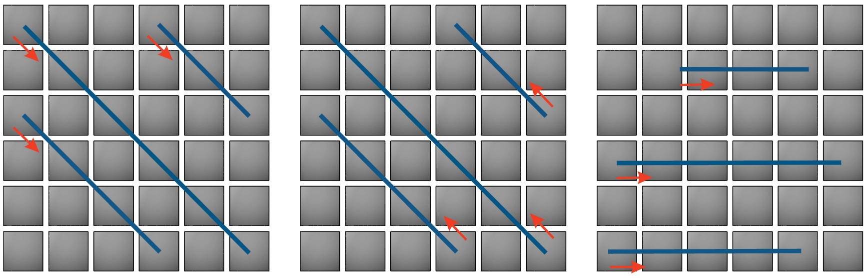


(a) Diagonal. (b) Secondary diagonal. (c) Horizontal line.

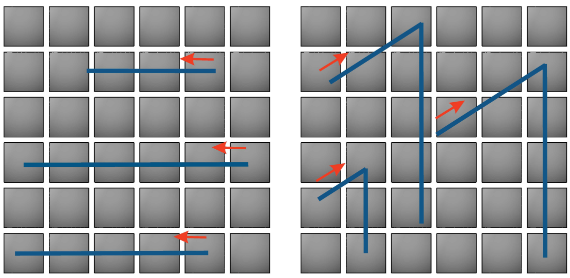


(d) Number 1. (e) Number 2.

Figure 4.5: Gestures analyzed for the map-based recognition algorithm.



(a) Diagonal from Upper-Left corner to Bottom-Right corner. (b) Diagonal from Bottom-Right to Upper-Left corner. (c) Horizontal line from Left to Right.



(d) Horizontal line from Right to Left. (e) Number 1.

Figure 4.6: Gestures analyzed for the centroid-based recognition algorithm.

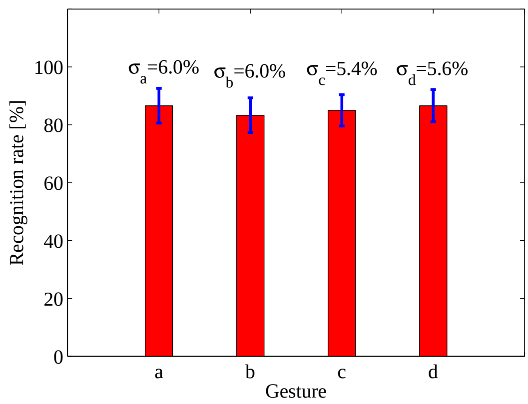
number 1 traced from left to right. It is evident, from the confusion matrix reported in Table 4.3, that the algorithm is able to discriminate the direction of the gesture. In order to evaluate the dependency of the algorithm on the specific performer, the standard deviation of the recognition rate has been computed considering the results obtained with the 20 performers for each gesture following Eq. (4.7)

$$\sigma_h = \left(\frac{1}{N} \sum_{i=1}^{N} (R_{i,h} - \mu_h)^2 \right)^{\frac{1}{2}}, \quad h = a, b, \ldots, \tag{4.7}$$

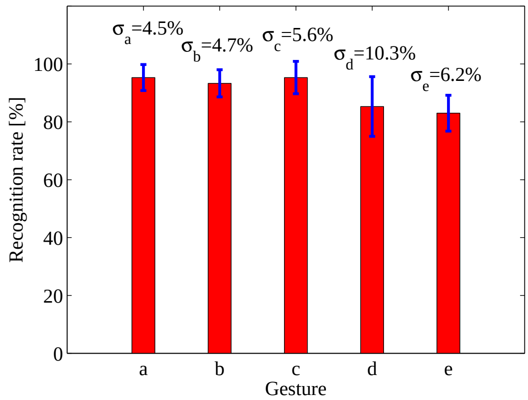
where N is the number of performers, $R_{i,h}$ is the average recognition rate achieved by the ith performer for the hth gesture and μ_h is the average recognition rate of hth gesture achieved by all performers computed as

$$\mu_h = \frac{1}{N} \sum_{i=1}^{N} R_{i,h}, \quad h = a, b, \ldots. \tag{4.8}$$

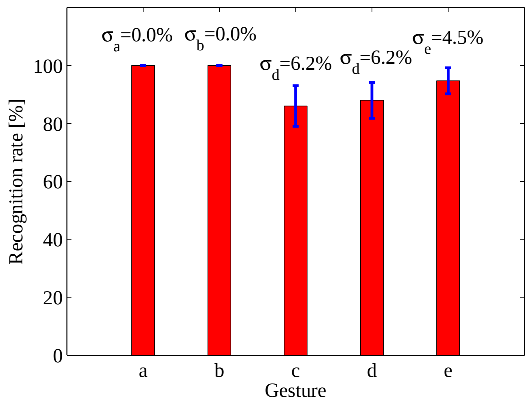
The results are reported in Fig. 4.7. The centroid-based algorithm shows a higher recognition rate for both simple and complex gestures, i.e., diagonals and numbers, and it is proven to be more independent from the codebook preliminarily acquired. Moreover, given the discrete nature of the features involved in the recognition process, i.e., the bit-map image and the coordinates of the contact point depends on the spatial resolution of the sensor, gestures such as horizontal lines, in some cases, are bad recognized for the difficulty to trace a really straight line. Finally, the low values of the standard deviations compared to the high value of the average recognition rates demonstrate that almost all algorithms are fairly independent from the performers. The centroid-based method is totally independent from the performer for the diagonal gestures that result easy to recognise, i.e., those with a 100% recognition rate. This feature is quite important since it allows the algorithms to be used effectively without any special training of the user.



(a) Dot product-based algorithm.



(b) Map-based algorithm.



(c) Centroid-based algorithm.

Figure 4.7: Assessment of algorithm sensitivity to the performer: average and standard deviation of gesture recognition rates for the varius performers (see Eqs. (4.7) and (4.8)).

CHAPTER 5

The force/tactile sensor for physical Human-Robot Interaction

The same device used so far to recognize haptic gestures can be adopted as a distributed force sensor to handle both intentional and unintentional contacts. In both cases, the accuracy in the detection of the contact force direction is crucial for ensuring a safe interaction with the human operator. In applications where the contact is intentional (advanced programming methods, interaction with the environment, manipulation), contact points are typically located on the links and an accurate estimation of both the location and contact force vector is needed for a proper motion of the robot, e.g., allowing the user to move the robot links along specific Cartesian directions and improving the intuitiveness of the collaboration. For handling unintentional collisions on other parts of the robot body a more rough information on contact location, but an accurate detection of the direction is needed for ensuring a safe behaviour of the robot, i.e., quickly moving the arm away from the operator who touched the robot.

5.1 Comparison between direct and indirect contact force sensing on a real robot

The aim of this section is to provide a validation of the calibration procedure described in Section 3.3 by testing the force/tactile sensor when mounted on a real robot as well as a comparison with two indirect techniques adopted by KUKA robots for providing information about the external forces exerted on the manipulator rigid links in terms of the estimation accuracy of the force vector. In particular, the more accurate, even though time consuming, calibration procedure has been selected for comparison purposes. A KUKA LWR4+ and a KUKA iiwa have been used for the estimation accuracy analysis. In both cases the sensor is connected to an acquisition board with a flat cable long enough to avoid to install it on the robot and the sensor voltages are acquired through the “scanning strategy” detailed in Section 2.4. The board sends over a USB connection the acquired raw data to a PC running a sensor library (see Section 7 for detailed information) for interfacing with the data acquisition board. The sensor library is able to provide, on the basis of the sensor voltages, the estimated force vectors applied to the 36 sensor modules, the contact points and the contact frames. The information are sent via a UDP socket to a second PC used to execute the robot control algorithm.

5.1.1 Residual-based method

In the first comparison the KUKA LWR4+ is considered. Two sensor patches are installed on the robot, one on the end effector and the other one on the third link (see Fig. 5.1). Several times, a force is applied by an operator to the force/tactile sensor on a single contact point while the robot is fixed in a given joint configuration, so as to allow a fair comparison with the contact force estimated using the residual-based method proposed in [103]. For the sake of completeness, such approach is briefly illustrated below. Consider the robot dynamic model, neglecting joint friction torques,

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau} + \boldsymbol{\tau}_{ext}, \tag{5.1}$$

where $\mathbf{q} \in \mathbb{R}^7$ is the vector of the generalized coordinates, $\mathbf{M}(\mathbf{q})$ is the symmetric, positive definite inertia matrix, $\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})$ is the matrix of Coriolis and centrifugal terms, $\mathbf{g}(\mathbf{q})$ is the vector of gravity torques, $\boldsymbol{\tau}$ is the vector of control torques, $\boldsymbol{\tau}_{ext}$ is the vector of torques due to external contact forces acting on the robot. The contact force estimation has been computed considering an approximate dynamic model of the KUKA LWR 4+ robot by neglecting friction torques.

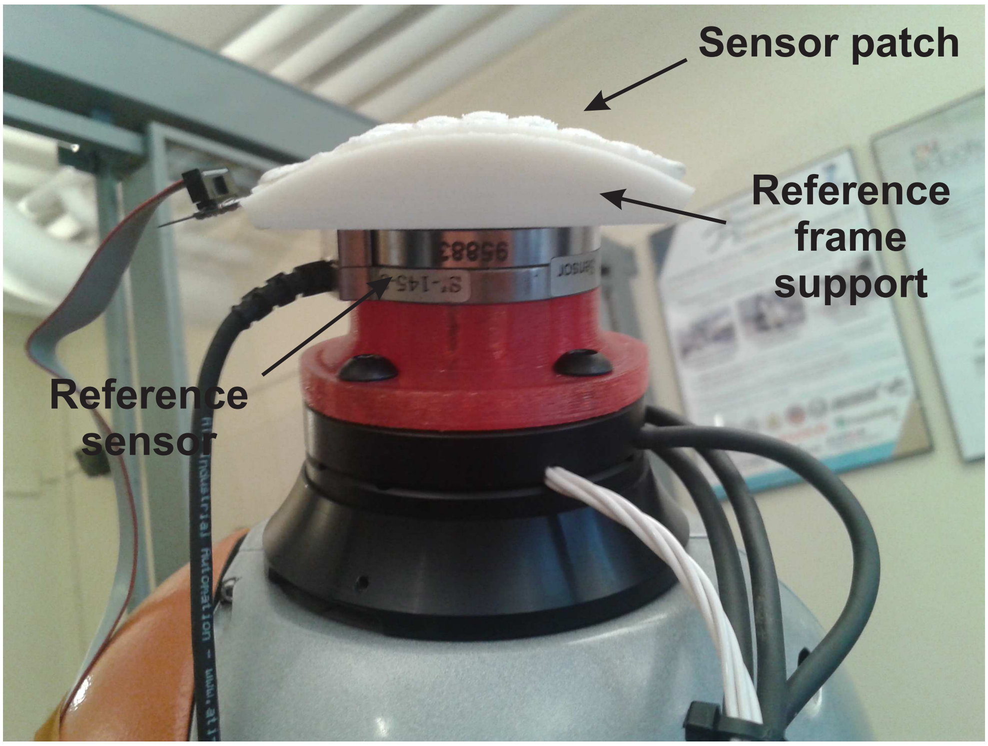
By defining the generalized momentum of the robot as

$$\mathbf{p} = \mathbf{M}(\mathbf{q})\dot{\mathbf{q}} \tag{5.2}$$

the residual vector $\mathbf{r} \in \mathbb{R}^7$ can be expressed as

$$\mathbf{r} = \mathbf{K}_I\left(\mathbf{p} - \int_0^t \left(\boldsymbol{\tau} - \mathbf{C}^T(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} - \mathbf{g}(\mathbf{q}) + \mathbf{r}\right)\mathrm{d}s\right), \tag{5.3}$$

with $\mathbf{r}|_{t=0} = 0$ and $\mathbf{K}_I$ a diagonal positive definite matrix. From Eqs. (5.1) and (5.2),



(a) The sensor patch installed on the end effector.

(b) The sensor patch installed on the third link.

Figure 5.1: The force/tactile sensor on the KUKA LWR 4+.

the dynamic of $\mathbf{r}$ can be expressed as

$$\dot{\mathbf{r}} = \mathbf{K}_I \left(\boldsymbol{\tau}_{ext} - \mathbf{r}\right). \tag{5.4}$$

By choosing a gain $\mathbf{K}_I$ large enough, assuming perfect knowledge of the dynamic model parameters, the asymptotic solution of (5.3) is

$$\mathbf{r} = \boldsymbol{\tau}_{ext}. \tag{5.5}$$

If $\mathbf{F}_k \in \mathbb{R}^3$ is an external force applied to a generic point of the robot and $\mathbf{J}_k(\mathbf{q})$ is the 3×7 Jacobian matrix associated to the contact point on the robot, the external torques, and thus the residual computed as in Eq. (5.5), are related to $\mathbf{F}_k$ as

$$\boldsymbol{\tau}_{ext} = \mathbf{J}_k^{\mathrm{T}}(\mathbf{q})\,\mathbf{F}_k. \tag{5.6}$$

By solving Eq. (5.6), the component of the contact force vector not laying in the null space of $\mathbf{J}_k^{\mathrm{T}}(\mathbf{q})$ can be computed as

$$\widehat{\mathbf{F}_k} = \left(\mathbf{J}_k^{\mathrm{T}}(\mathbf{q})\right)^{\#} \boldsymbol{\tau}_{ext}, \tag{5.7}$$

where $\left(\mathbf{J}_k^{\mathrm{T}}(\mathbf{q})\right)^{\#}$ is the generalized inverse of the Jacobian transpose.

The first experiment involves the force/tactile sensor installed on a ATI F/T Mini45 sensor mounted on the robot end effector. The force measured with the ATI F/T sensor has been used as ground-truth. Figure 5.2 reports the components of the force measured with the two sensors and the force components estimated with the residual-based method, while Table 5.1 reports the mean errors computed by means of Eq. (3.10). By observing the maximum value measured with the reference sensor for both the shear and normal components, the mean error for the tactile sensor is less than the 5%, while for the force estimated with the residual method it is about 30%. The force estimated with the residual-based method is clearly less accurate than the force measured by the tactile sensor. The greater estimation error likely depends on the use of an uncertain and approximate dynamic model available in the KUKA FRI Library.

The second experiment involves the force/tactile sensor installed on the third link of the KUKA LWR 4+. In this case study, the force measured with the proposed sensor has been compared only with the force estimated with the residual-based method since it was impossible to install the ATI sensor on the robot link. The results are

Table 5.1: Estimation errors with respect to the ATI F/T Mini45 sensor expressed in Newtons.

Sensor	f_x err	f_y err	f_z err
Tactile sensor	0.1412	0.1136	0.5721
Residual	1.1818	0.7100	4.9900

reported in Fig. 5.3, where large errors on the residual-based estimation of the x component of the contact force are experienced. This is easily explained since the x axis of the base frame, in which the force vectors are expressed, lies along the direction parallel to the third, which is in a configuration such that in this direction external forces are balanced by the mechanical structure of the arm and not by the joint torques. Also the y component is not perfectly estimated by the residual-based method even tough this component does not belong to the null space of the Jacobian transpose. From a safety point of view, this is a nice feature of the force/tactile sensor, since it allows the robot to detect contacts that cannot be detected by the other method. Moreover, since the sensor provides a direct measurement of the contact force, its installation on different parts of the robot structure does not affect the estimation accuracy. Differently, as reported in [104], the closer to the robot base is the contact point, the greater is the error of the residual estimation. Moreover, if three ore more contacts occur on the structure of a robot with seven or less DOF it is not possible to estimate the corresponding external forces, but just information on the external joint torques can be obtained. On the other hand, if a specific application required a higher full scale range for the tactile sensor the mechanical structure of the latter, e.g., the deformable layer, would have to be redesigned by using a silicone with a higher shore hardness. This issue does not affect the residual-based method since the full scale of the estimated residual depends on the limits of the robot structure and sensors.

5.1.2 Joint torque sensor-based method

In the second comparison the KUKA iiwa is considered. A sensor patch is installed on the third link, while, an ATI F/T Sensor Mini45 is mounted on the robot end effector as show in Fig. 5.4. The KUKA Sunrise.Connectivity FRI Library allows to obtain information about the external joint torques related to a contact force applied to the robot structure by simply calling a software routine. Such data is provided

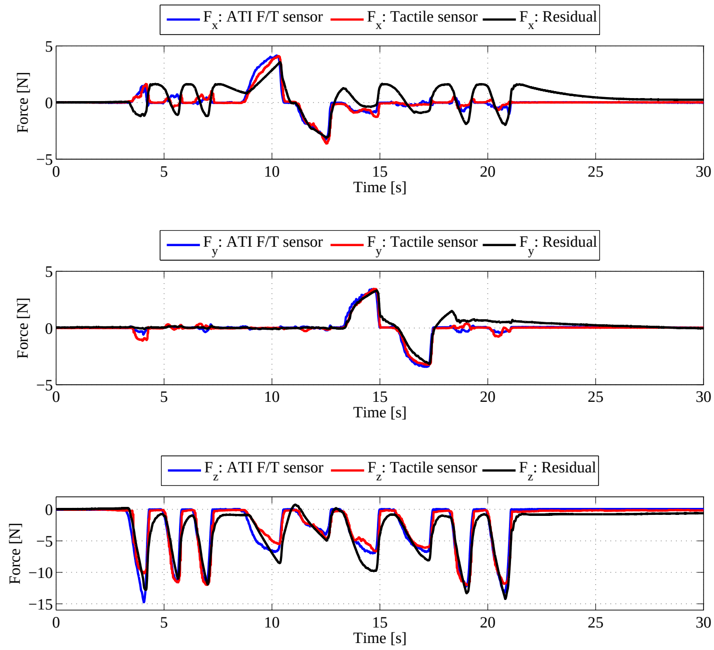


Figure 5.2: Components of the force applied to the robot end effector expressed in base frame.

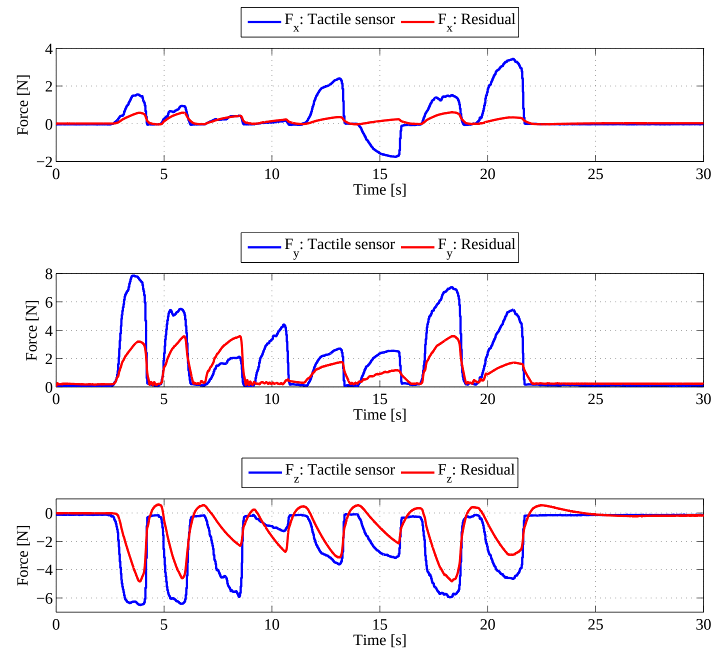
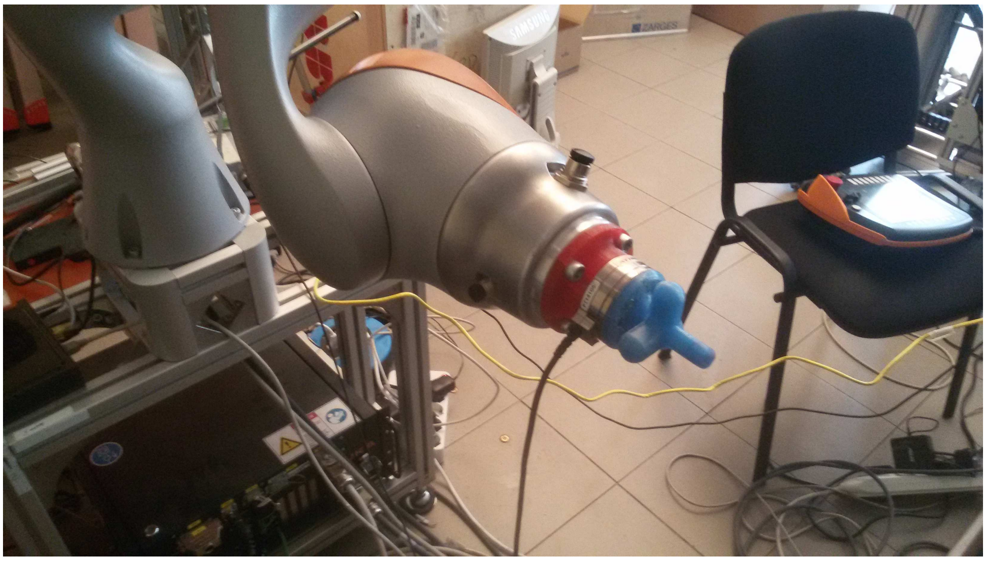


Figure 5.3: Components of the force applied to the third link of the robot expressed in the base frame.

(a) The sensor patch installed on the third link.



(b) The ATI F/T Sensor Mini45 installed on the end effector.

Figure 5.4: The force/tactile sensor on the KUKA iiwa.

on the basis of both the joint torque sensors measurements and the knowledge of the commanded joint torques. Suppose that the robot has rigid joints, so as to simplify the system model by neglecting the elastic dynamic related to the KUKA iiwa actuators. If $\boldsymbol{\tau}_{meas}$ is a 7×1 vector that contains the seven measured joint torques and $\boldsymbol{\tau}_{com}$ is a vector 7×1 that contains the seven commanded joint torques, the joint torques corresponding to an external contact can be simply obtained as

$$\boldsymbol{\tau}_{ext} = \boldsymbol{\tau}_{meas} - \boldsymbol{\tau}_{com}. \tag{5.8}$$

In the hypothesis that one external contact is applied to the robot and that the contact point $\boldsymbol{p}_k$ is known, an estimation of the force vector $\boldsymbol{F}_k$ can be computed with the following equation

$$\widehat{\mathbf{F}_k} = (\boldsymbol{J}_k^T)^{\#}\boldsymbol{\tau}_{ext}. \tag{5.9}$$

where $\boldsymbol{J}_k$ is the $3 \times n$ Jacobian matrix computed in the point where the force $\boldsymbol{F}_k$ is acting on the robot. Note that the estimate will be limited to only those components of $\boldsymbol{F}_k$ that can be detected by the external toques. In particular, all forces $\boldsymbol{F}_k \in \mathcal{N}(\boldsymbol{J}_k^T(\boldsymbol{q}))$ will not be recovered in $\widehat{\mathbf{F}_k}$. As reported in [103], Eq. 5.9 can be extended to the case of multiple simultaneous contact points. On the basis of the observations reported in Sec. 5.1.1, let consider two contact forces ($k = 1, 2$) acting on two different contact points $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$. Then, the equation become

$$\begin{pmatrix} \widehat{\mathbf{F}_1} \\ \widehat{\mathbf{F}_2} \end{pmatrix} = \left(\boldsymbol{J}_1^T(\boldsymbol{q})\boldsymbol{J}_2^T(\boldsymbol{q})\right)^{\#} \boldsymbol{\tau}_{ext}. \tag{5.10}$$

As before, the estimation will be intrinsically limited to the components of each contact force not lying in the kernel of the respective Jacobian transpose.

Again, a force is applied by an operator to the robot structure and the measured force is compared with the contact force estimated using the joint torque sensor-based method described above. In the first experiment, the robot has been position controlled commanding a constant joint trajectory (joint velocities $\dot{\boldsymbol{q}} = \mathbf{0}$), so as to considered a static condition. Figures 5.5 and 5.6 report the comparison results. In the first part of the experiment contact forces along different directions are applied to one point of the robot structure, first on the robot end effector, then, on the force/-tactile sensor (from 0 s to 41 s). On the basis of the accurate information of the point where the external force is acting, which is provided by the force/tactile sensor, the Jacobian computed in the contact point can be calculated and the external force has been estimated by Eq. (5.9). In the second part of the experiment (from 41 s to 60 s),

Table 5.2: Estimation errors with respect to the ATI F/T Mini45 sensor and to the force/tactile sensor expressed in Newtons.

/	$t <= 41$ s			$t > 41$ s		
Contact point	f_x err	f_y err	f_z err	f_x err	f_y err	f_z err
End effector	0.1310	0.1041	0.1182	0.9870	0.1166	0.8861
Third link	0.2956	0.3947	0.5084	1.8266	1.0903	2.1195

a multi-point contact is considered. Several forces are applied to both the robot end effector and the force/tactile sensor at the same time and the external forces acting on the two points have been estimated by using the extended formulation reported in Eq. (5.10). As expected, the estimation of the forces applied to the end effector is more accurate than the estimation of the forces applied to the robot third link, since the seven measured joint torques are all used in the estimation process. Although, a degradation in terms of the estimation accuracy has been observed when a force is simultaneously applied to two contact points. Table 5.2 reports the mean errors, which have been computed by means of Eq. (3.10), between the external forces estimated with the joint torque sensor-based method and the sensor measurements for both the end effector, using the ATI F/T Sensor Mini45, and robot third link, using the force/tactile sensor.

In the second experiment the forces applied to the two contact points are used to move the robot according to the force direction in a task of manual guidance. Figures 5.7 and 5.8 show the results. As in the previous experiment, in the first part a force is applied to a single contact point, then, the operator acts simultaneously on the robot end effector and on the third link. This time the estimation accuracy obtained with the joint torque sensor-based method are very poor due to the high noise affecting the torque measurements during the robot joint movements. Moreover, the use of a simplified model, which neglects the elasticity of the robot joints, significantly affects the estimation procedure. This analysis encourages, again, the use of a dedicated distributed force/tactile sensor in robotics applications where an accurate measure of the magnitude and of the direction of the external interaction force is needed.

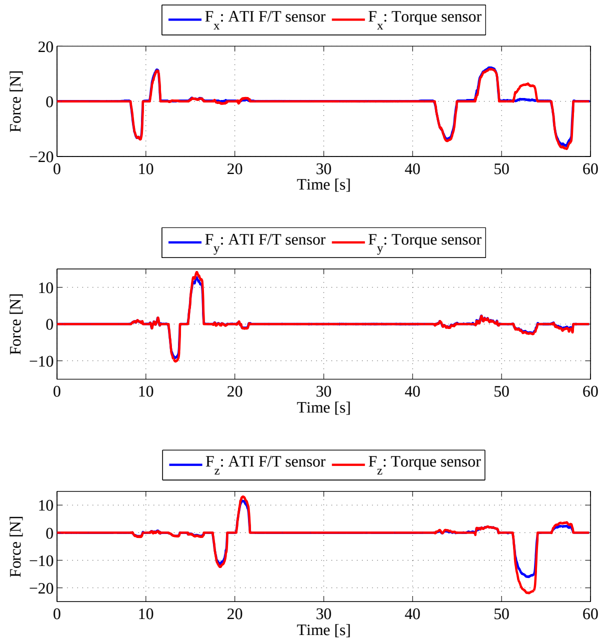


Figure 5.5: Static: components of the force applied to the robot end effector expressed in base frame.

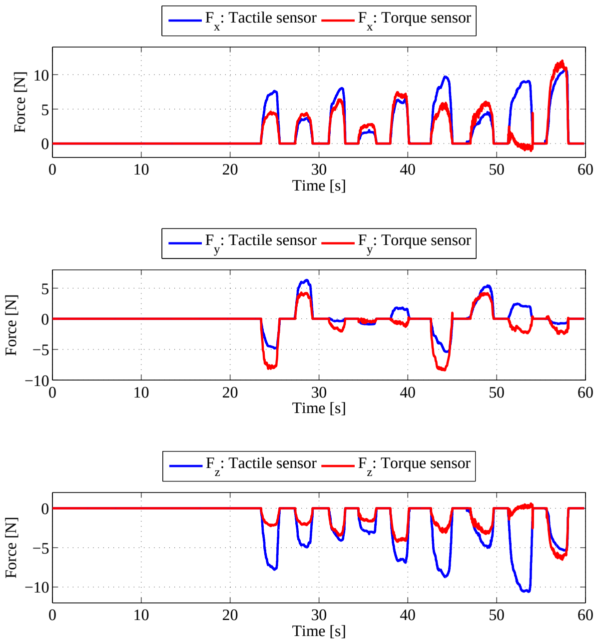


Figure 5.6: Static: components of the force applied to the third link of the robot expressed in the base frame.

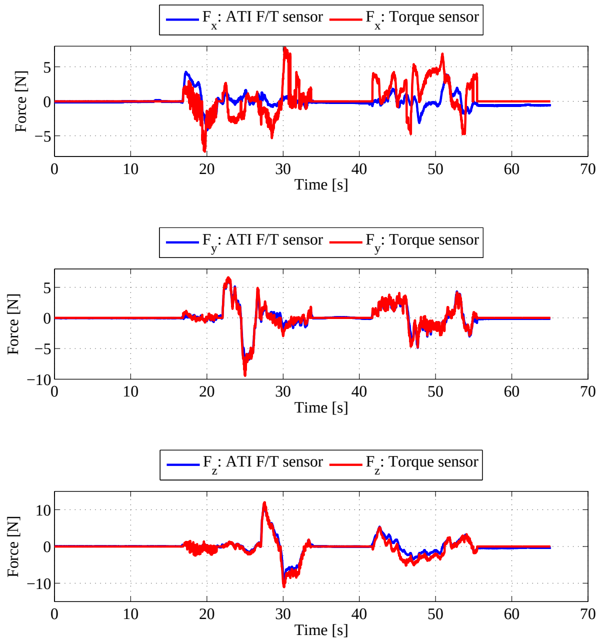


Figure 5.7: Dynamic: components of the force applied to the robot end effector expressed in base frame.

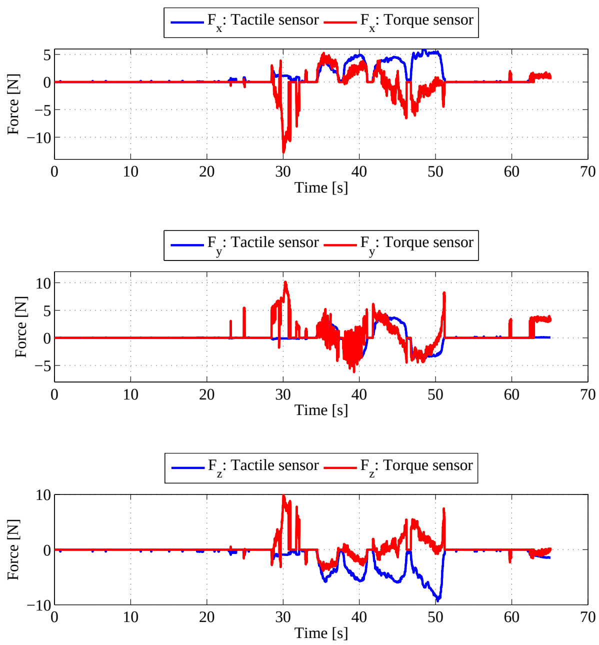


Figure 5.8: Dynamic: components of the force applied to the third link of the robot expressed in the base frame.

CHAPTER 6

The control strategies

The sensing capabilities of the force/tactile sensor can be used in a wide range of robotic applications. In this thesis, two standard control strategies have been implemented, in order to highlight all sensor capabilities. In particular, both the cases of intentional and unintentional contacts have been considered. In the case of intentional contacts, the forces measured by the sensor are used to manually guide the robot through multiple contact points. In the case of unintentional collisions detected by the sensor, the measured forces are used to achieve a safe reaction strategy.

6.1 Admittance control

The choice of admittance control is motivated by the safety requirement to ensure a robot motion in the Cartesian space coherent with the direction of the forces applied by humans. Two contact points are considered: one ($\boldsymbol{p}_e \in \mathbb{R}^3$) is located on the robot end effector and the other ($\boldsymbol{p}_b \in \mathbb{R}^3$) is located on link 3 of a 7-DOF robot. Let $\boldsymbol{f}_e \in \mathbb{R}^3$ and $\boldsymbol{f}_b \in \mathbb{R}^3$ be the corresponding contact forces.

The robot control law is a standard position control in the joint space, which allows to track a suitable reference joint trajectory $\boldsymbol{q}_r(t) \in \mathbb{R}^7$. This reference trajectory is computed according to a multi-priority algorithm for manual guidance or according to a collision reaction algorithm, depending on the magnitude of the sensed force $\boldsymbol{f}_b$, i.e.,

$$\boldsymbol{q}_r(t) = \begin{cases} \textit{manual guidance}, & \text{if } \|\boldsymbol{f}_b\| \leq f_{th}, \\ \textit{collision reaction}, & \text{if } \|\boldsymbol{f}_b\| > f_{th}, \end{cases} \tag{6.1}$$

being $f_{th} > 0$ a suitable threshold.

In the case of *manual guidance*, $\boldsymbol{q}_r(t)$ is computed on the basis of suitable dynamic relationships, or admittances, between the sensed contact forces and the displacements of the contact points, as explained below.

For a generic contact point $\boldsymbol{p}_c$, with $c = e, b$, the reference acceleration $\ddot{\boldsymbol{p}}_{c,r}$, velocity $\dot{\boldsymbol{p}}_{c,r}$ and position $\boldsymbol{p}_{c,r}$ are computed from the force $\boldsymbol{f}_c$ measured in the contact point by integrating the admittance equation:

$$\ddot{\boldsymbol{p}}_{c,r} = \boldsymbol{M}_c^{-1}(\boldsymbol{f}_c - \boldsymbol{D}_c\dot{\boldsymbol{p}}_{c,r}), \tag{6.2}$$

where $\boldsymbol{M}_c, \boldsymbol{D}_c \in \mathbb{R}^{3\times3}$ are suitable positive definite matrix gains, with the meaning of mass and damping respectively. In other words, the quantities $\ddot{\boldsymbol{p}}_{c,r}$, $\dot{\boldsymbol{p}}_{c,r}$, $\boldsymbol{p}_{c,r}$ represent the desired compliant motion of a virtual body located at point $\boldsymbol{p}_c$ with mass $\boldsymbol{M}_c$ and damping $\boldsymbol{D}_c$ under the action of the contact force $\boldsymbol{f}_c$.

Since the two contact points belong to the same kinematic chain, their motion cannot be assigned arbitrarily and conflicting situations may occur. These conflicts can be managed by the control through a suitable task priority strategy. Depending on the specific situation, the motion of one of the two contact points is considered as the main task, while the motion of the other point is considered as a secondary task. Only the motion components of the secondary task that are not conflicting with the main task, i.e., those projected in the null space of the Jacobian of the main task, will be executed.

The main task can be defined, for example, at the point which is touched first. Therefore, when the human applies a force $\boldsymbol{f}_e$ to the end effector (point $\boldsymbol{p}_e$) first, and then applies a force $\boldsymbol{f}_b$ to the robot's body (at point $\boldsymbol{p}_b$), the latter will cause a reconfiguration of the robot's body that does not affect the motion of $\boldsymbol{p}_e$, which depends only on $\boldsymbol{f}_e$. Vice versa, if the human applies first a force $\boldsymbol{f}_b$ to the robot's body at point $\boldsymbol{p}_b$, then the motion of point $\boldsymbol{p}_b$ will depend only on $\boldsymbol{f}_b$, also in the case that another force will be applied at the end effector.

The tasks priorities are managed with the Null Space-based Behavioural approach [105, 106, 107, 108] and they are handled at kinematic level in the joint space through IK (inverse kinematics). The joint space reference acceleration $\ddot{\boldsymbol{q}}_r(t)$ can be computed as

$$\ddot{\boldsymbol{q}}_r = \boldsymbol{J}_e^{\#}(\boldsymbol{r}_e - \dot{\boldsymbol{J}}_e\dot{\boldsymbol{q}}_r) + \dot{\boldsymbol{J}}_e^{\#}\boldsymbol{J}_e\dot{\boldsymbol{q}}_r + (\boldsymbol{I} - \boldsymbol{J}_e^{\#}\boldsymbol{J}_e)\left[\boldsymbol{J}_b^{\#}(\boldsymbol{r}_b - \dot{\boldsymbol{J}}_b\dot{\boldsymbol{q}}_r) - \boldsymbol{D}\dot{\boldsymbol{q}}_r\right], \tag{6.3}$$

where $\boldsymbol{J}_e^{\#}$ is the generalised inverse of the robot end effector Jacobian $\boldsymbol{J}_e \in \mathbb{R}^{3\times7}$, $\boldsymbol{J}_b^{\#}$ is the generalised inverse of the contact point Jacobian $\boldsymbol{J}_b \in \mathbb{R}^{3\times7}$, $(\boldsymbol{I} - \boldsymbol{J}_e^{\#}\boldsymbol{J}_e) \in \mathbb{R}^{7\times7}$ is the null space of $\boldsymbol{J}_e$, $\boldsymbol{D} \in \mathbb{R}^{7\times7}$ is a positive definite matrix with the meaning of a virtual damping, while the resolved acceleration vectors $\boldsymbol{r}_c \in \mathbb{R}^3$, with $c = e, b$ are computed as

$$\boldsymbol{r}_c = \ddot{\boldsymbol{p}}_{c,r} + k_d(\dot{\boldsymbol{p}}_{c,r} - \boldsymbol{J}_c\dot{\boldsymbol{q}}_r) + k_p(\boldsymbol{p}_{c,r} - \boldsymbol{k}_c(\boldsymbol{q}_r)), \tag{6.4}$$

being k_d, k_p strictly positive gains. The reference vectors $\ddot{\boldsymbol{p}}_{c,r}$, $\dot{\boldsymbol{p}}_{c,r}$ and $\boldsymbol{p}_{c,r}$ are computed using the admittance equation (6.2), while the vector $\boldsymbol{k}_c(\boldsymbol{q}_r)$ is the contact point position computed from $\boldsymbol{q}_r$ using the forward kinematics mappings $\boldsymbol{k}_c(\boldsymbol{q}_r)$, with $c = e, b$.

Equation (6.3) assumes that the motion of point $\boldsymbol{p}_e$ has higher priority with respect to the motion of point $\boldsymbol{p}_b$. The change of priority can be achieved using the same equation, by replacing the subscript e with the subscript b and viceversa. Notice that, when the Jacobians are close to a singularity, high joint acceleration and speed can be generated yielding high tracking errors and possibly dangerous situations. To mitigate such effects the generalised inverse can be robustly calculated using the damped least squares pseudo-inverse, with the method proposed in [109]. The same admittance strategy can be also used to manage unexpected collisions, i.e., a *collision reaction* when the threshold f_{th} in (6.1) is overcome. In this case, a safe reaction is commanded to the robot according to the following criterion. The primary task is interrupted and a motion of the detected contact point is commanded still according to the admittance equation (6.2), but $\boldsymbol{M}_c$ and $\boldsymbol{D}_c$ are suitably selected so as the reaction time and the magnitude of the repulsive acceleration generate a quick reflex motion of the robot. In particular, the point where the collision is detected by the sensitive skin, moves in the same direction of the applied force. Hence, the reference joint space acceleration becomes

$$\ddot{\boldsymbol{q}}_r = \boldsymbol{J}_c^{\#}(\ddot{\boldsymbol{p}}_{c,r} - \dot{\boldsymbol{J}}_c\dot{\boldsymbol{q}}_r + k_d(\dot{\boldsymbol{p}}_{c,r} - \boldsymbol{J}_c\dot{\boldsymbol{q}}_r) + k_p(\boldsymbol{p}_{c,r} - \boldsymbol{k}_c(\boldsymbol{q}_r))). \tag{6.5}$$

In turn, the reference joint space trajectory $\boldsymbol{q}_r(t)$ in (6.1) can be computed by integrating (6.3) for manual guidance or (6.5) for collision reaction. Notice that, since the Jacobian $\boldsymbol{J}_c$ has null columns from 4 to 7, the accelerations of the joints from 4 to 7 computed using Eq. 6.2 are null. Therefore, the corresponding robot links freeze just after the collision.

6.2 Control law stability

To the author's best knowledge, the literature does not provide an analysis of the stability of a kinematic control law for redundant manipulator based on a second-order Closed-Loop Inverse Kinematic (CLIK) algorithm. In [110] the authors describe a

CLIK algorithm to control a robot manipulator and they report the theoretical analysis of the algorithm convergence on the basis of a Lyapunov argument. However, the algorithm convergence has been demonstrated by considering a non-redundant six-joint manipulator. Starting from the work presented in [111], in which spatial impedance control with redundancy resolution has been considered, this section reports an analysis of the control law stability described in Sec. 6.1.

Let consider the control law

$$\ddot{\boldsymbol{q}}_r = \boldsymbol{J}_e^{\#}(\boldsymbol{q}_r)(\boldsymbol{r}_e - \dot{\boldsymbol{J}}(\boldsymbol{q}_r)\dot{\boldsymbol{q}}_r) + \boldsymbol{\phi}_n, \tag{6.6}$$

where $\boldsymbol{J}_e$ is $(6 \times n)$ Jacobian matrix relating joint velocities $\dot{\boldsymbol{q}}_r$ to the velocities of the end effector $\boldsymbol{v}_e$, with n equal to the number of the robot joints. Premultiplying both sides by $\boldsymbol{J}_e$, given that $\dot{\boldsymbol{v}}_e = \boldsymbol{J}_e(\boldsymbol{q}_r)\ddot{\boldsymbol{q}}_r + \dot{\boldsymbol{J}}_e(\boldsymbol{q}_r)\dot{\boldsymbol{q}}_r$ and observing that $\boldsymbol{J}_e\boldsymbol{J}_e^{\#} = \boldsymbol{I}$, $\boldsymbol{J}_e\boldsymbol{\phi}_n = \boldsymbol{0}$ yields

$$\dot{\boldsymbol{v}}_e = \boldsymbol{r}_e, \tag{6.7}$$

that is a resolved end-effector acceleration that can be computed with the CLIK algorithm as in (6.4). The null-space joint accelerations $\boldsymbol{\phi}_n$ have to be chosen so as to ensure stabilization of the null-space motion.

Consider the matrix $(\boldsymbol{I} - \boldsymbol{J}_e^{\#}\boldsymbol{J}_e)$ projecting a vector in the null space of $\boldsymbol{J}_e$. Then, let

$$\boldsymbol{e}_n = (\boldsymbol{I} - \boldsymbol{J}_e^{\#}(\boldsymbol{q}_r)\boldsymbol{J}_e(\boldsymbol{q}_r))(\boldsymbol{\gamma} - \dot{\boldsymbol{q}}_r) \tag{6.8}$$

denote the null-space velocity error where $\boldsymbol{\gamma}$ is a joint velocity vector which is available for redundancy resolution. The goal is to make $\boldsymbol{e}_n$ asymptotically converge to zero. Taking the time derivative of (6.8) and using (6.6) gives the null-space dynamics

$$\dot{\boldsymbol{e}}_n = (\boldsymbol{I} - \boldsymbol{J}_e^{\#}(\boldsymbol{q}_r)\boldsymbol{J}_e(\boldsymbol{q}_r))(\dot{\boldsymbol{\gamma}} - \boldsymbol{\phi}_n) - (\dot{\boldsymbol{J}}_e^{\#}(\boldsymbol{q}_r)\boldsymbol{J}_e(\boldsymbol{q}_r) + \boldsymbol{J}_e^{\#}(\boldsymbol{q}_r)\dot{\boldsymbol{J}}_e(\boldsymbol{q}_r))(\boldsymbol{\gamma} - \dot{\boldsymbol{q}}_r) \tag{6.9}$$

where $\dot{\boldsymbol{J}}_e^{\#}$ is the time derivative of $\boldsymbol{J}_e^{\#}$. Consider the Lyapunov function candidate

$$V = \frac{1}{2}\boldsymbol{e}_n^T\boldsymbol{e}_n. \tag{6.10}$$

Computing the time derivative of (6.10) along the trajectories of system (6.9) yields

$$\dot{V} = \boldsymbol{e}_n^T(\dot{\boldsymbol{\gamma}} - \boldsymbol{\phi}_n - \dot{\boldsymbol{J}}_e^{\#}\boldsymbol{J}_e(\boldsymbol{\gamma} - \dot{\boldsymbol{q}}_r)), \tag{6.11}$$

where the dependence on $\boldsymbol{q}_r$ has been dropped off and the following identities have been exploited

$$\boldsymbol{e}_n^T \boldsymbol{J}_e^{\#} = \mathbf{0}^{\mathbf{T}} \tag{6.12}$$

$$\boldsymbol{e}_n^T (\boldsymbol{I} - \boldsymbol{J}_e^{\#} \boldsymbol{J}_e) = \boldsymbol{e}_n^T. \tag{6.13}$$

Let choose

$$\boldsymbol{\phi}_n = (\boldsymbol{I} - \boldsymbol{J}_e^{\#} \boldsymbol{J}_e)(\dot{\boldsymbol{\gamma}} - \dot{\boldsymbol{J}}_e^{\#} \boldsymbol{J}_e (\boldsymbol{\gamma} - \dot{\boldsymbol{q}}_r) + \boldsymbol{D}\boldsymbol{e}_n) \tag{6.14}$$

where $\boldsymbol{D}$ is a positive definite matrix. By combining (6.11) and (6.14) yields

$$\begin{aligned} \dot{V} &= \boldsymbol{e}_n^T (\dot{\boldsymbol{\gamma}} - \dot{\boldsymbol{J}}_e^{\#} \boldsymbol{J}_e (\boldsymbol{\gamma} - \dot{\boldsymbol{q}}_r) - \dot{\boldsymbol{\gamma}} + \dot{\boldsymbol{J}}_e^{\#} \boldsymbol{J}_e (\boldsymbol{\gamma} - \dot{\boldsymbol{q}}_r) - \boldsymbol{D}\boldsymbol{e}_n + \\ &\quad + \boldsymbol{J}_e^{\#} \boldsymbol{J}_e (\dot{\boldsymbol{\gamma}} - \dot{\boldsymbol{J}}_e^{\#} \boldsymbol{J}_e (\boldsymbol{\gamma} - \dot{\boldsymbol{q}}_r) + \boldsymbol{D}\boldsymbol{e}_n)) \\ &= -\boldsymbol{e}_n^T \boldsymbol{D}\boldsymbol{e}_n < 0, \end{aligned} \tag{6.15}$$

where the identity 6.12 has been used again. It can be concluded that the choice (6.14) gives a negative definite $\dot{V}$ with a positive definite $\boldsymbol{D}$, and thus $\boldsymbol{e}_n \rightarrow \mathbf{0}$ asymptotically. With the choice of such $\boldsymbol{\phi}_n$, by considering $\boldsymbol{\gamma} = \mathbf{0}$, the joint space reference acceleration $\ddot{\boldsymbol{q}}_r$ become

$$\ddot{\boldsymbol{q}}_r = \boldsymbol{J}_e^{\#}(\boldsymbol{r}_e - \dot{\boldsymbol{J}}_e \dot{\boldsymbol{q}}_r) + \dot{\boldsymbol{J}}_e^{\#} \boldsymbol{J}_e \dot{\boldsymbol{q}}_r + (\boldsymbol{I} - \boldsymbol{J}_e^{\#} \boldsymbol{J}_e) \left[\boldsymbol{J}_b^{\#}(\boldsymbol{r}_b - \dot{\boldsymbol{J}}_b \dot{\boldsymbol{q}}_r) - \boldsymbol{D}\dot{\boldsymbol{q}}_r \right]. \tag{6.16}$$

CHAPTER 7

The integration on redundant manipulators

The author worked on the integration of the proposed force/tactile sensor on four 7-DOF robots of different brands. The robots chosen for this scope are:

- a KUKA LWR 4+ kindly provided by the PRISMALab of the Università di Napoli Federico II
- a KUKA LWR 4+ kindly provided by the Dynamic Human-Robot Interaction (DHRI) laboratory of the Technische Universität München (TUM)
- a KUKA iiwa kindly provided by the PRISMALab of the Università di Napoli Federico II
- a YASKAWA SIA5F provided by DIII of the Seconda Università degli Studi di Napoli (SUN)

In all the cases, it has been shown that the technique used to install the sensor on a robot link, which was described in Section 3.2, results to be an effective, reliable and quick solution. As shown in Figure 7.1, the sensor has easily been conformed to the surface of the four robots. The force/tactile sensor has been exploited in the execution of many collaborative and safety tasks, i.e., manual guidance, intuitive programming, collision detection and reaction. In the following sections, a brief description of the HW/SW architectures and the system interfaces used with the four robots are reported.

(a) KUKA LWR 4+ at PRISMALab.

(b) KUKA LWR 4+ at TUM.

(c) KUKA iiwa at PRISMALab.

(d) YASKAWA SIA5F at DIII of the SUN.

Figure 7.1: Force/tactile sensor installed on the redundant robots.

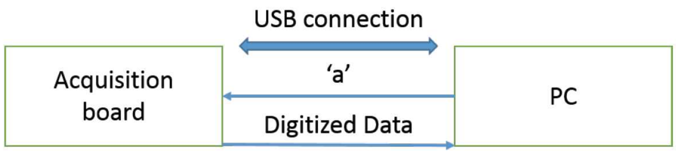


Figure 7.2: Communication protocol between the sensor acquisition board and the PC.

7.1 Sensor SW drivers

As explained in the first part of this thesis, the force/tactile sensor provides 4 voltages for each sensing module and, then, it provides 144 voltages for a patch of 36 sensing elements. On the basis of the adopted scanning strategy, the acquisition board exploits 12 A/D channels with a resolution of 12 bit to digitize the 144 voltage signals. The acquisition board chosen to digitize the sensor voltages is the STM32F3 Discovery board, provided by STMicroelectronics, based on a STM32F3 ARM Cortex-M4 microcontroller. It is constituted of a 32-bit CPU with a clock frequency of 72 MHz, a FPU unit, a 12-channel DMA controller and 16 A/D channels with a maximum resolution of 12 bits. The digitized data (288 bytes) are sent through an USB connection to a PC with a simple communication protocol (see Fig. 7.2) with a maximum frequency of about 150 Hz. The PC sends the char *a* to the acquisition board and, then, it receives the 288 bytes corresponding to the digitized voltage signals. To support the data acquisition and the data elaboration two software drivers have been developed. Firstly, a C++ Sensor Library for both Microsoft Windows and Linux O.S. has been developed and tested with the KUKA LWR4+ and KUKA iiwa provided by PRISMALab. Secondly, a sensor driver compliant with the ROS[4] specifications has been developed in order to make available the data provided by the sensor to a generic robotic system. Latter has been tested with the KUKA LWR4+ provided by TUM and with the YASKAWA SIA5F available at DIII. The sensor drivers are able to

- communicate with the acquisition board,
- compute the signals offset,
- provide the sensor Tactile Map,

[4]The Robot Operating System: http://www.ros.org/

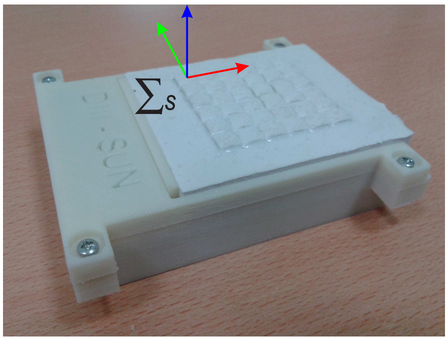


Figure 7.3: Sensor reference frame: rigid sensor.

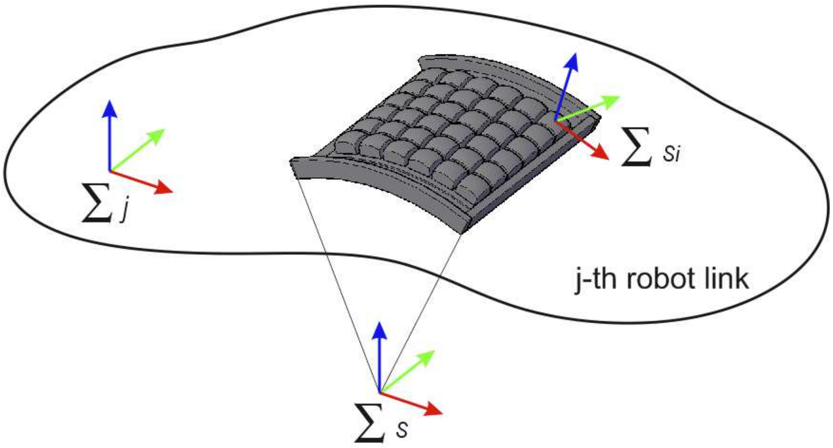


Figure 7.4: Sensor reference frame: conformable sensor.

- compute the contact point and the force vector applied to each sensing module w.r.t. the sensor reference frame Σ_s reported in Figs. 7.3 and 7.4,
- estimate the contact point(s) and the net force(s) applied to compact area(s),
- recognize touch gestures applied on the sensor surface.

On the basis of the specific control system and communication interface available with each robot a proper system architecture has been defined and tested.

7.1.1 C++ sensor library

This section is aimed at providing a clear documentation to support the C++ Sensor Library, by describing in detail all library methods. The library has been developed for both Microsoft Windows and Linux O.S. in order to have a full compatibility

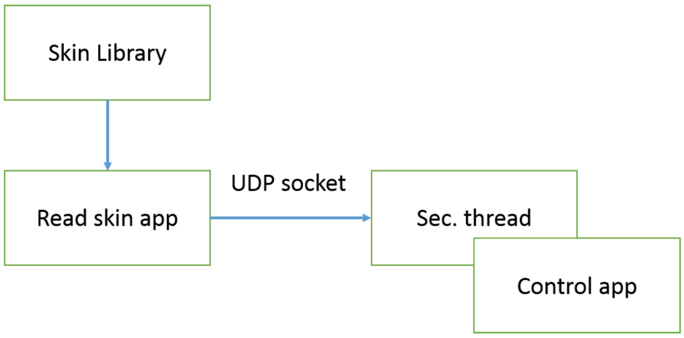


Figure 7.5: C++ Sensor Library SW architecture.

with such robotics systems that exploit Ethernet based communication interface, e.g., KUKA LWR4+ and KUKA iiwa FRI Library. To work correctly with the two O.S. the C++ Library uses different support libraries for the serial and socket communication. In any case, the Linux version needs that the *Boost* library has been correctly installed. Note that the serial communication under Linux O.S. has been found to be more reliable than the serial communication under Microsoft Windows O.S.; with Windows O.S. different errors during the data acquisition were observed. The acquisition board provides the data of a skin patch of 36 sensing elements with a maximum frequency of 150 Hz. Since most of the control algorithms need a faster execution frequency, the SW architecture reported in Fig. 7.5 has been chosen. The *ReadSkin* application communicates with the acquisition board by using the C++ Sensor Library. It sends the sensor information, e.g., Tactile Map, the three components of the force exerted on the 36 sensing modules, contact point(s), recognized gesture, through an UDP socket to the *Control* application. The latter uses a secondary thread to asynchronously acquire the UDP packet. The data are organized in a customized data structure, *skin_socket_struct*, that contains:

- *skinType*: integer value. It can be equal to RIGID_SKIN (0) or CONFORMABLE_SKIN (1)
- *calibrationMethod*: integer value. It can be equal to NOT_FINE_CALIBRATION (0) or FINE_CALIBRATION (1)
- *tactileMap*: a 12×12 matrix of float values. It contains the voltage signals of the 36 sensing modules organized in a matrix of 12×12 elements
- *fx_modules*: a 6×6 matrix of float values. It contains the x-component of the

force vector applied to the 36 sensing modules organized in a matrix of 6×6 elements

- *fy_modules*: a 6×6 matrix of float values. It contains the y-component of the force vector applied to the 36 sensing modules organized in a matrix of 6×6 elements

- *fz_modules*: a 6×6 matrix of float values. It contains the z-component of the force vector applied to the 36 sensing modules organized in a matrix of 6×6 elements

- *fx_resultant*: a vector of 36 float values. It contains the x-component of the net force(s) applied to the connected component(s) detected in the tactile map (the number of the detected components can dynamically changes)

- *fy_resultant*: a vector of 36 float values. It contains the y-component of the net force(s) applied to the connected component(s) detected in the tactile map (the number of the detected components can dynamically changes)

- *fz_resultant*: a vector of 36 float values. It contains the z-component of the net force(s) applied to the connected component(s) detected in the tactile map (the number of the detected components can dynamically changes)

- *contact_points*: a 36×6 matrix of float elements. It contains the:
  - x-coordinate of the *i*th compact centroid in the element *contact_points[i][0]*
  - y-coordinate of the *i*th compact centroid in the element *contact_points[i][1]*
  - z-coordinate of the *i*th compact centroid in the element *contact_points[i][2]*
  - ϕ angle (in XYZ Euler angles representation) of the reference frame related to the *i* compact centroid in the element *contact_points[i][3]*
  - θ angle (in XYZ Euler angles representation) of the reference frame related to the *i* compact centroid in the element *contact_points[i][4]*
  - ψ angle (in XYZ Euler angles representation) of the reference frame related to the *i* compact centroid in the element *contact_points[i][5]*

- *numContacts*: integer value. It is equal to the number of the detected contact regions.

- *recogGesture*: integer value. It is equal to an integer identifier associated to the recognized touch gesture.

The C++ Sensor Library has been implemented as a C++ Class. In Figures 7.6 and 7.7 the Class Diagrams for the Microsoft Windows and Linux version, respectively, are reported. They differ for the attributes used to establish the serial and socket connection. In Appendix A the library, the methods and their input/output parameters are described.

7.1.2 A ROS-based sensor driver

The C++ Sensor driver described in the previous section results to be a complete communication interface for the proposed tactile/force sensor. It provides simple sensor information, e.g., tactile map, or more complex sensor information, e.g., recognized touch gesture, that allow to use the sensor in the execution of different type of robotics tasks, i.e., manual guidance, intuitive programming. However, the sensor data are sent through an UDP connection that needs, in general, an Ethernet Switch where the robot control cabinet, the PC used to control the robot and the PC used to acquire the sensor data are connected and, often, non-trivial modifications to the control algorithm code. In general creating robust, general-purpose robot software is very hard. In the last five years the community of robotics and computer science researchers worked on a flexible, distributed and modular framework for writing robot software, namely Robot Operating System (ROS). ROS is a collection of tools, libraries, and conventions that aim to simplify the task of creating complex and robust robot behavior across a wide variety of robotic platforms. The distributed and modular nature of ROS encourages the integration of software modules developed by researchers that work on different robotics topics that, this time, becomes faster and easier. Given that the idea is using the force/tactile sensor with heterogenous robotics system a sensor driver compliant with the ROS specifications has been developed following the same software requirements using in the design of the C++ Sensor Library described before. Figure 7.8 reports a schematization of the ROS nodes net. The ROS driver network consists of four nodes, dedicated to the data acquisition and elaboration, and four ROS topics. The nodes involved in the data elaboration are:

- *skin_serial* is the main node that acquires the sensor data. It communicates with the sensor through an USB interface: the data are transferred with a baudrate up to 921600 bps. The binary values of the sensor signals are properly converted in 144 voltage signals and the latter are published in the *raw_data* topic with a maximum frequency of 150 Hz. A reorganization of the data in a 12×12 matrix allows to obtain a distributed map of the pressure applied on the

Skin

- offsetMap: float[12][12]
- gesture: int[12][12]
- g_cp1: vector<float>
- g_cp2: vector<float>
- gesture_thr: float
- stop_gesture: int
- gesture_timeIst: int
- si_other: struct sockaddr_in
- s: int
- skin_sensor: skin_socket_struct
- calibrationMethod: int
- skinType: int
- gestureType: int
- normalizationMatrix: float[12][12]
- pressureMap: float[12][12]
- calibrationMatrix: float[3][4][36]
- cTmod: float[4][4][36]
- gestureDB: gestureStruct
- gestureDB2: gestureStruct2
- COMport: string
- SP: Serial*

+ Skin(int skin_type, int calibration_method, int gesture_type, char* ip, int port) : void
+ setCOM(char* port) : void
+ connect(void) : void
+ disconnect(void) : void
+ getNormalizationMatrix(void) : void
+ update(void) : void
+ removeOffset(int numSamples) : void
+ getForces(float fx[6][6], float fy[6][6], float fz[6][6], float V[12][12]) : void
+ findCompact(int y_comp[36], int x_comp[36], int* index, int Mret[6][6], int M[6][6], int xi, int yi) : void
+ getResultantForces(float Fx[36], float Fy[36], float Fz[36], float contactPoints[36][6], int* numContacts, float fx[6][6], float fy[6][6], float fz[6][6]) : void

Figure 7.6: Skin Class diagrams: Windows O.S. version.

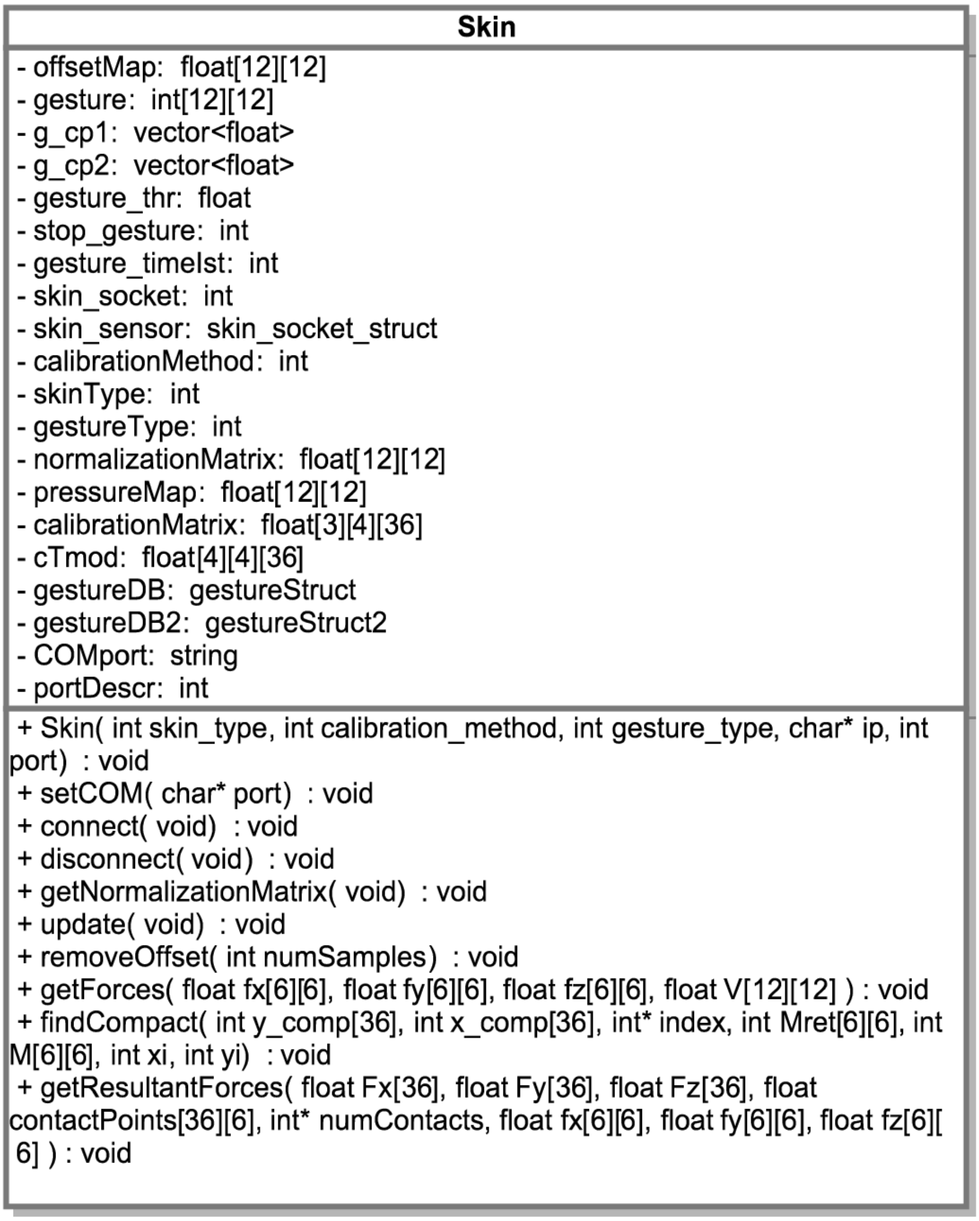


Figure 7.7: Skin Class diagrams: Linux O.S. version.

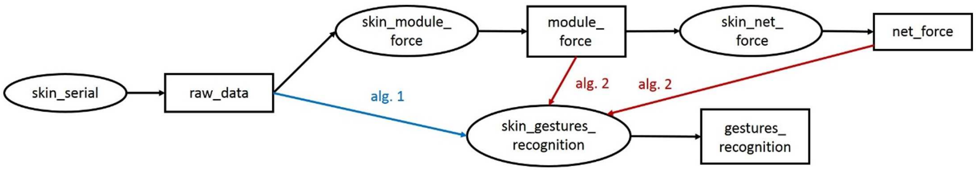


Figure 7.8: Scheme of the ROS nodes.

sensor surface.

- *skin_module_force* elaborates the data, which are provided to the *raw_data* topic from the previous node, the contact forces acting on each tactile sensor module. The 36 forces, expressed in Newton (N), are sent to the *module_force* topic. The node needs a set of calibration parameters that depend on the type of the tactile sensor (rigid or flexible) and on the type of the calibration procedure preliminarily carried out (fine or not fine). The information can be passed to the node with input parameters during the starting phase as described below.

- *skin_net_force* processes the data acquired from the *module_force* topic and it provides information about the net forces acting on one or more compact regions on the sensor surface. The data are saved in the *net_force* topic.

- *skin_gestures_recognition* node provides information about the touch gestures applied on the tactile sensor surface. It requires as input parameter the identifier of the algorithm to use for the recognition process. The two algorithms described in Section 4 have been implemented. The first one exploits the data provided by the *raw_data* topic and it uses a map-based recognition algorithm. The second one uses data acquired from both *module_force* and *net_force* topics and it used an algorithm based on the force contact point. The recognized gesture is communicated through the *gestures_recognition* topic.

It is possible to start the ROS nodes with the ROS command *rosrun* specifying the node input parameters in order to set specific user requests for the sensor data elaboration:

- *rosrun skin_driver skin_seral*

- *rosrun skin_driver skin_module_force skin_type calibration_type*
  with *skin_type* equal to *rigid*/*flex* and *calibration_type* equal to *not_fine*/*fine*

- *rosrun skin_driver skin_net_force skin_type calibration_type*
  with *skin_type* equal to *rigid*/*flex* and *calibration_type* equal to *not_fine*/*fine*

- *rosrun skin_driver skin_gestures_recognition algorithm*
  with *algorithm* equal to *img_based*/*cp_based*

For completeness, a schematization of the ROS topics involved in the driver network and of the message types exchanged in the ROS net is reported in Table 7.1: each row shows a topic and the corresponding message structure. Moreover, an example of a ROS node is reported in Appendix B. The code corresponds to the *skin_serial* node. A serial port is defined and the communication speed is set to the maximum baud rate achievable by the USB interface and compatible with the VirtualCom Port specifications, e.g., 921600 bps. The node starts to acquire a set of samples in order to compute the sensor signal offsets. Then, a message is filled with the data acquired from the sensor and it is published as the *raw_data* topic.

7.2 The sensor on the KUKA LWR4+ at PRISMALab

The KUKA LWR 4+ provided by the PRISMALab exploits the Fast Research Interface (FRI) Library, developed by the University of Stanford, to communicate with a control PC and/or third part devices, i.e., F/T sensor. The library intends to provide a simple user interface to the KUKA Light-Weight Robot IV and hides all communication and set-up issues behind interface. It is only an interface and it does not contain any control functionalities. It allows accessing to different controller interfaces of the KUKA system, e.g., joint position controller, cartesian impedance controller and gravity compensation controller. The FRI Library runs on a remote PC that is connected to the KRC (KUKA Robot Controller) via an Ethernet connection. In intervals of 1 to 100 ms, UDP packages are periodically sent from the KRC unit to the remote host. These packages contain a complete set of robot control and status data, e.g., joint positions, joint torques, drive FRIDriveTemperatures (see FRI documentation [112] for more details). The remote host (e.g., with QNX Neutrino RTOS) instantaneously send a reply message after the reception of each package. The reply message contains input data for the applied controllers, e.g., joint position set-points, joint stiffness set-points. In this way, the users become able to set-up own control architectures and/or application-specific controllers for the light-weight arm as it is often desired at research institutions.

Since a PC has to be used to run the FRI Library, the idea was to use the PC itself

Table 7.1: Schematization of ROS sensor driver topics and message types.

Topic name	Message structure	Description
raw_data	Header header	Message Header
	float32[144] data	A 144 elements vector that contains the tactile sensor signals
module_force	Header header	Message Header
	float32[36] fx float32[36] fy float32[36] fz	Three 36 elements vectors that contain the x-,y-,z-components of the module forces
net_force	Header header	Message Header
	float32[36] net_fx float32[36] net_fy float32[36] net_fz	Three 36 elements vectors that contain the x-,y-,z-components of the net forces
	float32[36] contact_points_x float32[36] contact_points_y float32[36] contact_points_z float32[36] contact_points_rx float32[36] contact_points_ry float32[36] contact_points_rz	Six 36 elements vectors that contain the position and orientation (in terms of XYZ Euler angle) of the frame in which the contact point of each estimated net force is expressed. They are expressed w.r.t. the sensor frame
	int32 num_contacts	The number of the estimated net forces
gestures_recognition	Header header	Message Header
	int32 recognized_gesture	An identifier of the recognized gesture

to host also the software library needed to communicate with the force/tactile sensor. Ideally, a minimal system architecture should consist of a host PC interfaced with the force/tactile sensor, through the C++ Sensor Library described in Section 7.1.1, and with the robot cabinet, through an Ethernet connection and the FRI Library. However, to properly work, the FRI Library requires to run on an Ubuntu 32-bit version that is incompatible with the VirtualCOM Port driver used to communicate with the force/tactile sensor. For the cited reasons, an alternative system architecture has been considered and implemented. It makes use of two PC. The first PC hosts an Ubuntu 64-bit version and it is used to communicate with the force/tactile sensor through the C++ Sensor Library. The second one is used to execute the robot control algorithm and to send the joint commands to the robot through the FRI Library. The to PCs are connected together with an Ethernet switch. Figure 7.9 shows a schematization of the minimal system architecture and of the architecture successfully tested in this specific case. In this configuration the force/tactile sensor is used to perform several experiments that are described in the following.

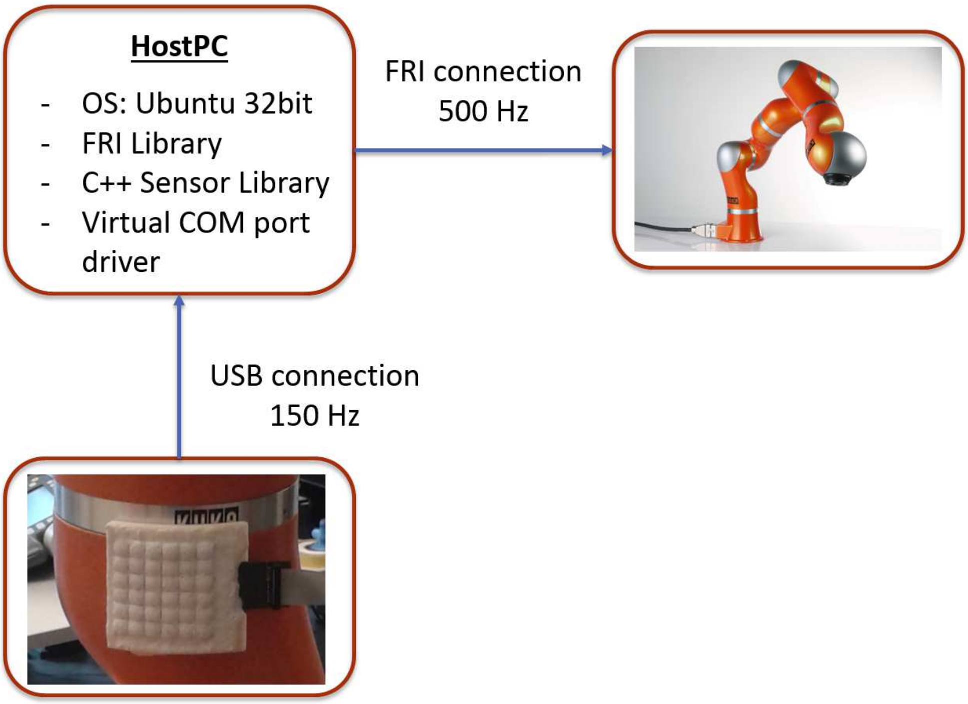
7.2.1 KUKA LWR4+ D-H table

In order to develop a positioning algorithm and to compute the direct and inverse kinematics of the robot a kinematic model of the KUKA LWR4+ in term of Denavit-Hartenberg (D-H) table, is required. With reference to Fig. 7.10, the D-H parameters are reported in Tab. 7.2. Moreover, if Σ_b and Σ_e are the base frame and the end-effector frame, respectively, the following homogeneous matrix have to be considered:

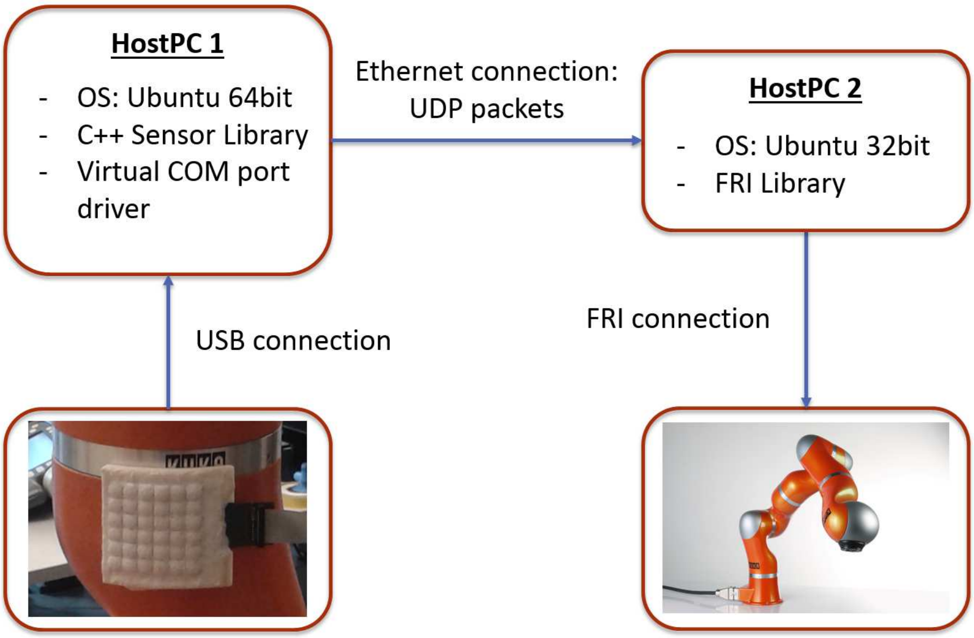
$$\boldsymbol{T}_1^b = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_1^b \\ 0 & 0 & 0 & 1 \end{bmatrix}, \boldsymbol{T}_e^7 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_e^7 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{7.1}$$

where $d_1^b = 0.3105\,\text{m}$ and $d_e^7 = 0.078\,\text{m}$. In the considered D-H convention the joint frame are all right-handed, while in the KUKA convention the frames 4 and 6 are left-handed. So, the relations between the KUKA convention and the D-H convention are:
$q_1^{D-H} = q_1^{KUKA}$
$q_2^{D-H} = q_2^{KUKA}$
$q_3^{D-H} = q_3^{KUKA}$
$q_4^{D-H} = -q_4^{KUKA}$
$q_5^{D-H} = q_5^{KUKA}$
$q_6^{D-H} = -q_6^{KUKA}$
$q_7^{D-H} = q_7^{KUKA}$
Similarly, the joint velocities and accelerations are:
$\dot{q}_1^{D-H} = \dot{q}_1^{KUKA}$
$\dot{q}_2^{D-H} = \dot{q}_2^{KUKA}$
$\dot{q}_3^{D-H} = \dot{q}_3^{KUKA}$
$\dot{q}_4^{D-H} = -\dot{q}_4^{KUKA}$
$\dot{q}_5^{D-H} = \dot{q}_5^{KUKA}$
$\dot{q}_6^{D-H} = -\dot{q}_6^{KUKA}$
$\dot{q}_7^{D-H} = \dot{q}_7^{KUKA}$

$\ddot{q}_1^{D-H} = \ddot{q}_1^{KUKA}$



(a) Minimal system architecture.



(b) Implemented system architecture.

Figure 7.9: System architecture adopted for KUKA LWR4+ at PRISMALab.

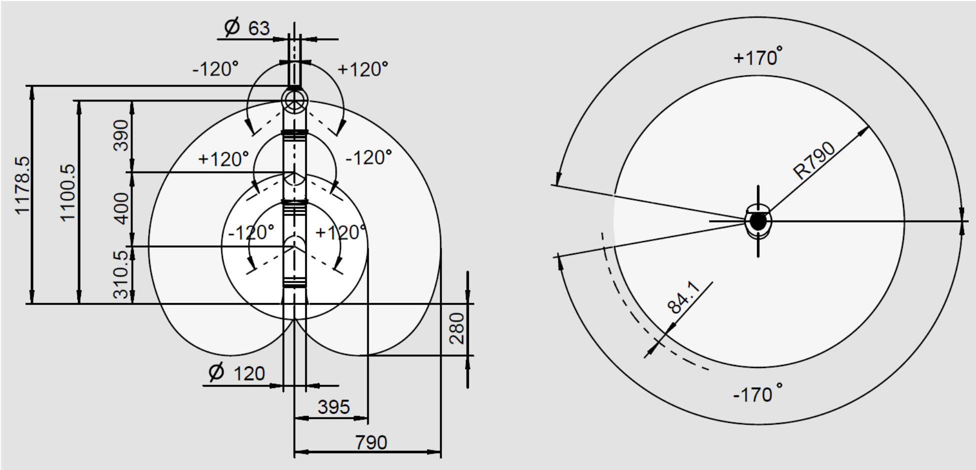


Figure 7.10: KUKA LWR4+ mechanical scheme.

N. Joint	α_i	a_i [mm]	d_i [mm]	θ_i
1	$\pi/2$	0	0	q_1
2	$-\pi/2$	0	0	q_2
3	$\pi/2$	0	0.4	q_3
4	$-\pi/2$	0	0	q_4
5	$\pi/2$	0	0.39	q_5
6	$-\pi/2$	0	0	q_6
7	0	0	0	q_7

Table 7.2: KUKA LWR4+ D-H table.

$$
\begin{aligned}
\ddot{q}_2^{D-H} &= \ddot{q}_2^{KUKA}\\
\ddot{q}_3^{D-H} &= \ddot{q}_3^{KUKA}\\
\ddot{q}_4^{D-H} &= -\ddot{q}_4^{KUKA}\\
\ddot{q}_5^{D-H} &= \ddot{q}_5^{KUKA}\\
\ddot{q}_6^{D-H} &= -\ddot{q}_6^{KUKA}\\
\ddot{q}_7^{D-H} &= \ddot{q}_7^{KUKA}
\end{aligned}
$$

7.2.2 Experiments

The aim of this section is to show how the sensor can be used to recognize simultaneously intentional and unintentional contacts and how the robot can be used in different way exploiting the information on the contact force on the basis of the control law described in Section 6.

A conformable force/tactile sensor patch is installed on the third link of the KUKA LWR4+ (see Fig. 7.1(a)) as described in Sec. 3.2. With this choice the number of joints from the base to the contact point is enough to move the contact point along any direction of the Cartesian space. The sensor is connected to an acquisition board with a flat cable. The acquisition board chosen to digitize the sensor voltages is the STM32F3 Discovery board based on a STM32F303 ARM Cortex-M4 microcontroller. Given the system architecture reported in 7.9(b), the acquisition board sends over an USB connection the acquired raw data to a HostPC, at a sampling frequency rate of 150 Hz, on which the C++ Sensor Library is running. The latter is able to provide, on the basis of the sensor voltages, the estimated force vectors applied to the 36 skin sensing modules and the contact point(s). The information, then, are sent via an UDP socket to a second PC used to compute the control algorithm. To work properly, the KUKA controller requires that the commanded joint positions values have to be updated with a rate of at least 500 Hz. So, a different thread is used to asynchronously acquire the sensor data. The second host is interfaced with the KUKA LWR4 robot with the KUKA FRI Library.

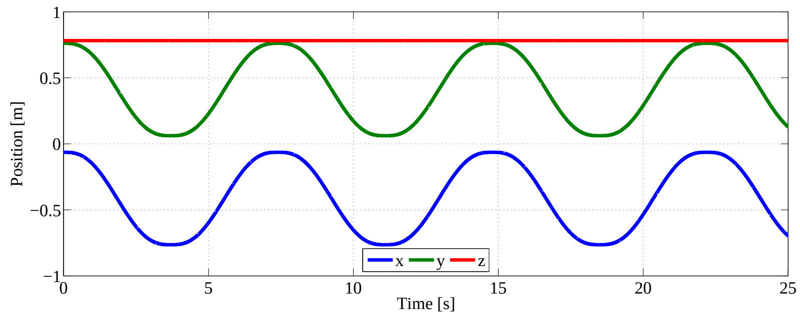
The force $\boldsymbol{f}_b$ measured with the sensor and used in the control (and in particular in (6.2) written with $c = b$) is defined as the net force acting on the whole contact surface of the sensor patch, computed as

$$\begin{aligned} \boldsymbol{f}_b &= \boldsymbol{R}_s \sum_{i=1}^{36} \boldsymbol{f}_i^s \\ \boldsymbol{R}_s &= \boldsymbol{R}_j \boldsymbol{R}_s^j \end{aligned} \tag{7.2}$$

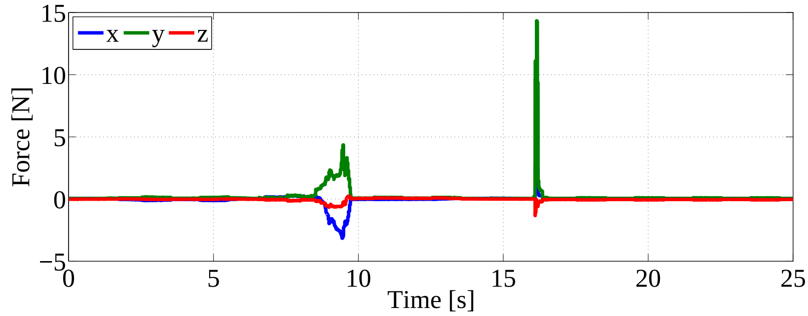
where $\boldsymbol{f}_i^s$ is the force measured by the $i-$th sensor sensing module as defined in Sec. 2.6.1, $\boldsymbol{R}_s^j$ is the rotation matrix that expresses the orientation of the sensor frame w.r.t. the robot j-th link and $\boldsymbol{R}_j$ is the rotation matrix of j-th link w.r.t. the robot base frame. The control parameters have been selected by choosing in equation 6.2 the scalar matrices: $\boldsymbol{M}_e = m_e\boldsymbol{I}_3$, $\boldsymbol{D}_e = d_e\boldsymbol{I}_3$, $\boldsymbol{M}_b = m_b\boldsymbol{I}_3$, $\boldsymbol{D}_b = d_b\boldsymbol{I}_3$. With this choice the response times for the end effector and the body control points are proportional to m_e/d_e and m_b/d_b, respectively, while the magnitudes of the repulsive accelerations are proportional to $1/d_e$ and $1/d_b$, respectively. In the experiments the following values have been selected: $m_e = 25\,\mathrm{kg}$, $d_e = 6\,\mathrm{Ns/m}$, $m_b = 0.5\,\mathrm{kg}$, $d_b = 2\,\mathrm{Ns/m}$, $k_p = 50\,\mathrm{s}^{-2}$ and $k_d = 10\,\mathrm{s}^{-1}$. This choice ensures a well damped response for both contact points with a quicker reaction time for the body control point.

7.2.2.1 First experiment

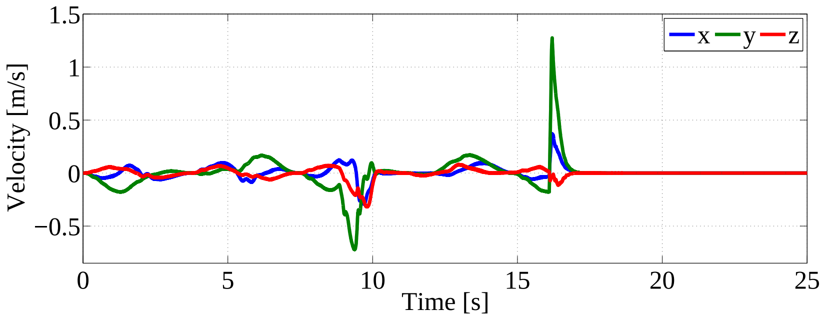
The objective of the first experiment is to show how the sensor can be used to intentionally interact with the robot and to safely react to unintentional contacts at the same time. With this aim, the control law reported in (6.1) has been implemented. In particular, the robot is programmed to follow, as primary task, a periodic trajectory at the end effector. Figure 7.11(a) reports the desired trajectory at the end effector that corresponds to a horizontal line in the space. A threshold equal to 7 N has been fixed for the force/tactile sensor. In the first 8 s, the primary task is correctly executed with a trajectory error near to zero (see Fig. 7.11(d)), while no contacts occur with the sensor (see Fig. 7.11(b)). During the task execution, an intentional contact is applied to the force/tactile sensor in order to reconfigure the robot in an elbow configuration which is more comfortable for the user that has to act in the robot workspace. The contact occurs between 8 s and 10 s and the detected force, reported in Fig. 7.11(b), is below the established threshold. Thus, the desired motion in Fig. 7.11(c), computed at the contact point, is projected in the null space of the first task, which is then preserved, as the low trajectory error demonstrates in Fig. 7.11(d). Figure 7.12(a) reports the angles of joints 2, 3 and 4 that move the elbow. Given the initial joints configuration ($q_2 > 0$ and $q_4 < 0$), the robot moves along the prescribed trajectory, in absence of contacts, in the elbow-up configuration (first 8 s). By applying a proper force and by exploiting the redundant DOFs, the user is able to safely move the robot from the elbow-up ($q_3 > -1.57$ rad) to the elbow-down ($q_3 < -1.57$ rad), while the robot still executes the main task (between 8 s and 10 s). Then, at 16 s, a second contact occurs and this time the force exceeds the threshold. Hence, the robot controller interprets the force as an unintended collision, and it imposes to the collision point a motion in the Cartesian direction of the collision force, to preserve the safety of the human. From this instant of time, the primary task is abandoned and when the contact force falls to zero, the robot stops (see last 9 s). Obviously, the robot could be programmed to resume the interrupted operation as soon as the contact is no longer detected. To assess the safety of the reflex motion, besides the direction of the motion that has been already shown coherent with the applied force, the reaction time of the whole system has been estimated. Figure 7.12(b) reports on the same plot a zoom view, around 16 s, of the measured net force magnitude and the velocity component along y axis, that is the main direction of motion of the collision point. The time that the velocity needs to change its sign, in which the robot reacts to the collision escaping from the contact area, is approximately 89 ms, meaning that only a limited amount of energy is transferred to the human during the unexpected contact. It is evident how



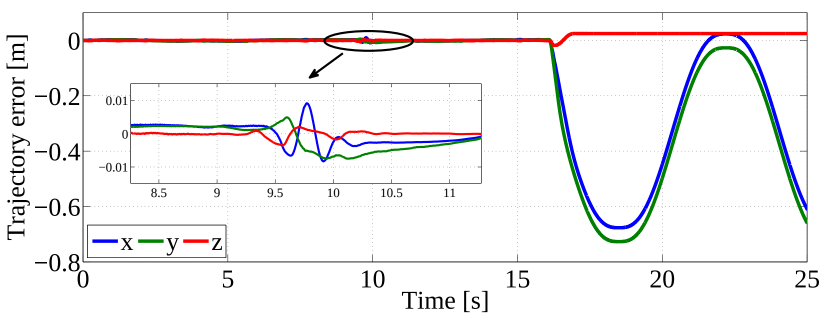
(a) Desired end-effector trajectory.



(b) Collision force components measured with the force/tactile sensor.

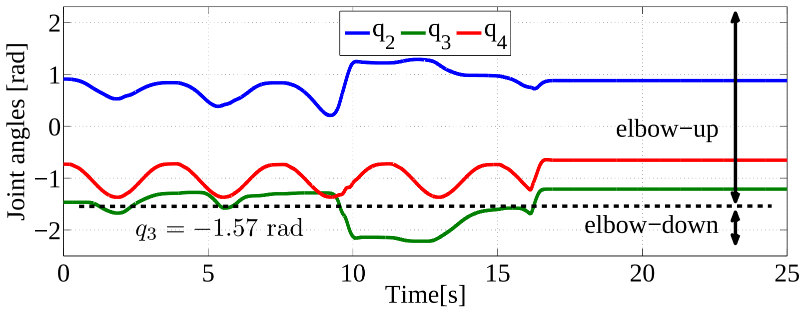


(c) Velocity of the collision point.

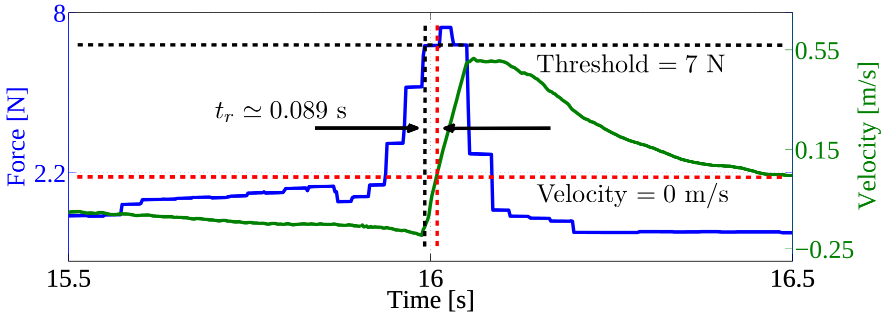


(d) Error on the end-effector trajectory.

Figure 7.11: KUKA LWR4+ - first experiment.



(a) Angles of joint 2, 3 and 4.



(b) Response time.

Figure 7.12: KUKA LWR4+ - first experiment: elbow reconfiguration and response time after a collision.

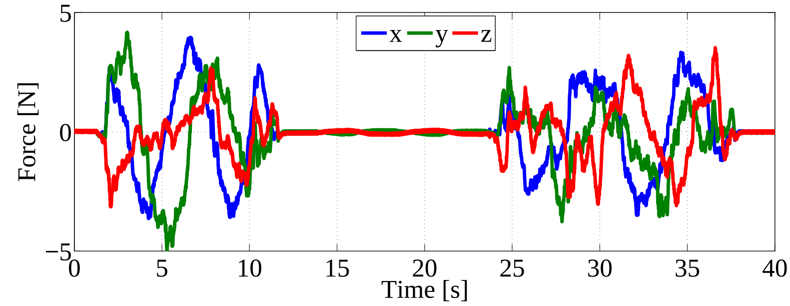
with the good sensibility of the sensor in the estimation of the three force components and with high mechanical robustness, the skin can be used at the same time to reconfigure the robot in a fine and precise way and to escape in case of dangerous situations due to unintentional collision.

7.2.2.2 Second experiment

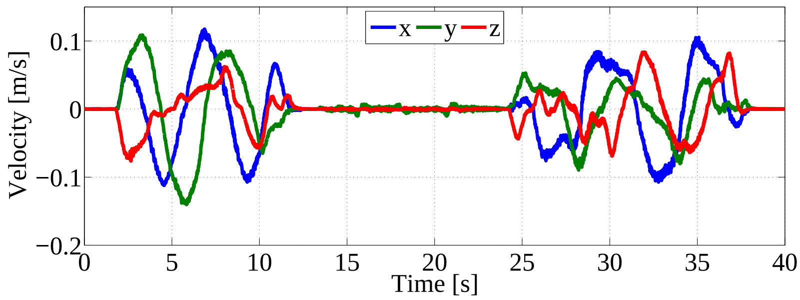
With the second experiment, the force/tactile sensor is used in a manual guidance task. The objective is to show how the high accuracy of the force estimation of the sensor allows to use the sensor in the same way a commercial F/T sensor is usually adopted when mounted on the robot wrist, with the advantage that the skin patch can be mounted in different parts of the robot structure. In this case, the task priority is not fixed and the contact force is below the force threshold during the whole experiment. In particular, as discussed in Sec. 6, the control law has been implemented in such a way that the priority of the two tasks is defined by the operator on the basis of the first point touched. The commercial sensor installed on the wrist to implement the proposed experiment is an ATI Mini45 F/T sensor. In both possible contact points, the desired position is computed by the admittance equation (6.2). In particular, the desired position of the end effector is computed by using the ATI sensor, while the position of the body point is computed on the basis of the data provided by the sensor skin. Two different case studies will be analyzed in order to show the robot behavior when the task priority changes, and the sensor used to manage the primary tasks switches, accordingly, from the commercial one to the skin sensor proposed in this paper.

Case study I The first case study illustrates the behavior of the robot when the operator first touches the end effector. In this case the desired position of the end effector constitutes the main task. The results are reported in Fig. 7.13. By observing Figs. 7.13(a) and 7.13(b) the end effector moves according to the forces exerted at the tip, measured by the commercial sensor. When the operator touches the point on the robot body (the force/tactile sensor), the exerted forces, represented in Fig. 7.13(c), produce a motion in the null space of the main task. The velocities $\dot{\boldsymbol{p}}_b$, reported in Fig. 7.13(d), are composed by the motion allowed in the null space and the motion produced by the main task. The forces applied to the point $\boldsymbol{p}_b$ (Fig. 7.13(c)) and the velocities $\dot{\boldsymbol{p}}_e$ (Fig. 7.13(b)) clearly show that the secondary task does not affect the task with higher priority and that the motion takes place in a direction coherent with the direction of the applied force. This makes the interaction with the robot very intuitive, in contrast to a simple gravity compensation mode, that, by the way, could be applied only with a steady end effector.

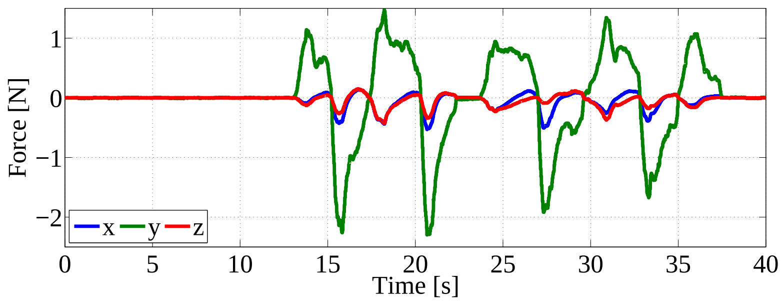
Case study II Figure 7.14 reports a similar analysis for the second case study. The desired position of the body point is selected as the main task by first touching the point $\boldsymbol{p}_b$ on the force/tactile sensor. Figures 7.14(b), 7.14(c) and 7.14(d) show that the velocities $\dot{\boldsymbol{p}}_e$ and $\dot{\boldsymbol{p}}_b$ are affected by the force $\boldsymbol{f}_b$ only. Instead, Figs. 7.14(a), 7.14(b) and 7.14(c) show that the end-effector motion does not generate a contribution to the velocity $\dot{\boldsymbol{p}}_b$, while it is coherent with the direction of the applied force. It is evident that in this second case the guidance of the robot, obtained by using the proposed sensor, is qualitatively similar to the previous case study. This experiment demonstrates that the sensibility and the accuracy of the proposed sensor, in the estimation of all contact force components, are high enough to use the sensor for intuitive guidance and programming of a robot also when the necessary measured forces are below 1 N, with the advantage that the force/tactile sensor can be conformed to be easily



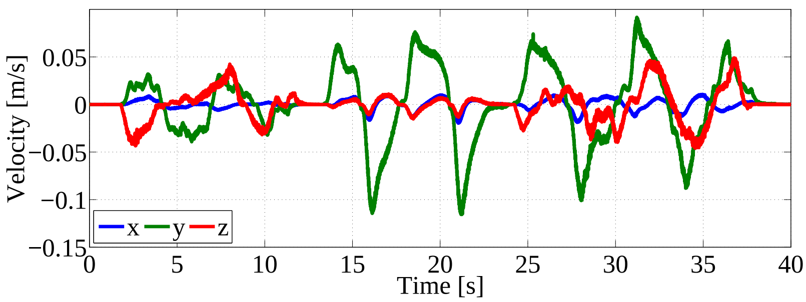
(a) Force components measured with the ATI F/T sensor.



(b) Velocity in Cartesian space of the end effector.



(c) Force components measured with the force/tactile sensor.



(d) Velocity in Cartesian space of the body contact point.

Figure 7.13: KUKA LWR4+ - second experiment (case study I): all components are expressed w.r.t. the robot base frame.

mounted on different parts of the robot.

7.3 The sensor on the KUKA iiwa

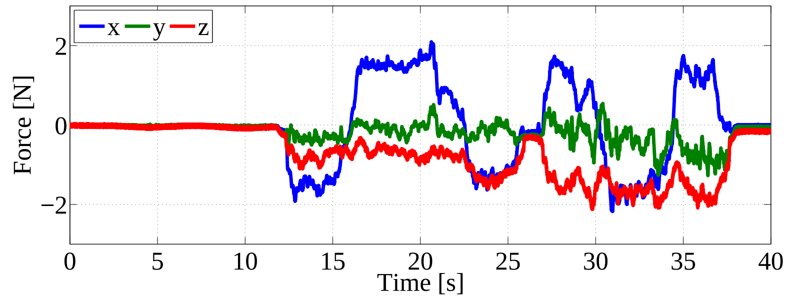
The system architecture experimentally tested with the KUKA iiwa is quite similar to the one described in the previous section. KUKA iiwa uses the new Sunrise.Connectivity FRI library to establish a communication channel between the PC that executes the control algorithm, developed in C++, and the Java-based controller. It provides a simple and light C++ Object Oriented library that allows commanding the robot through a joint position interface. So, just trivial modification in the software design has been enough to obtain a working system. In this specific case, both the capacities of the force/tactile sensor to provide the contact force vector and to work as HMI has been exploited to carry out a task of intuitive programming, in the sense that the touch gestures recognized by the sensor have been used to send different commands to the robot.

7.3.1 KUKA iiwa D-H table

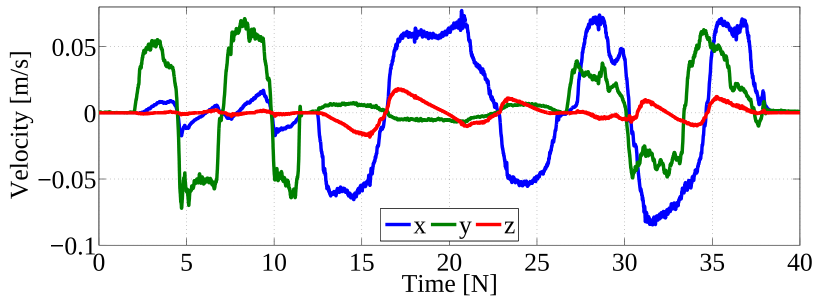
The D-H parameters of the KUKA iiwa differ from the ones of the KUKA LWR4+ just for few arguments. They are reported in Tab. 7.3. This time, if Σ_b and Σ_e are the base frame and the end-effector frame, respectively, the following homogeneous matrix have to be considered:

$$\boldsymbol{T}_1^b = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_1^b \\ 0 & 0 & 0 & 1 \end{bmatrix}, \boldsymbol{T}_e^7 = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & d_e^7 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{7.3}$$

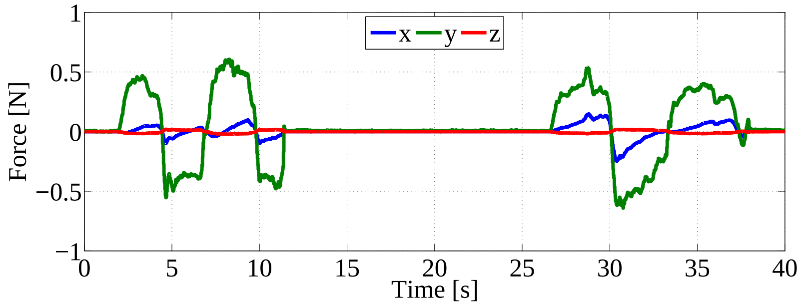
where $d_1^b = 0.34\,\mathrm{m}$ and $d_e^7 = 0.126\,\mathrm{m}$. In the considered D-H convention the joint frame are all right-handed, while in the KUKA convention the frames 4 and 6 are left-handed. So, the relations between the KUKA convention and the D-H convention are:
$q_1^{D-H} = q_1^{KUKA}$
$q_2^{D-H} = q_2^{KUKA}$
$q_3^{D-H} = q_3^{KUKA}$
$q_4^{D-H} = -q_4^{KUKA}$
$q_5^{D-H} = q_5^{KUKA}$



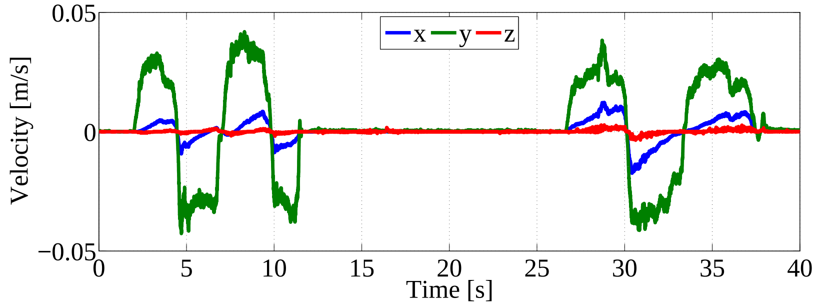
(a) Force components measured with the ATI F/T sensor.



(b) Velocity in Cartesian space of the end effector.



(c) Force components measured with the force/tactile sensor.



(d) Velocity in Cartesian space of the body contact point.

Figure 7.14: KUKA LWR4+ - second experiment (case study II): all components are expressed w.r.t. the robot base frame.

N. Joint	α_i	a_i [m]	d_i [m]	θ_i
1	$\pi/2$	0	0	q_1
2	$-\pi/2$	0	0	q_2
3	$\pi/2$	0	0.4	q_3
4	$-\pi/2$	0	0	q_4
5	$\pi/2$	0	0.4	q_5
6	$-\pi/2$	0	0	q_6
7	0	0	0	q_7

Table 7.3: KUKA iiwa D-H table.

$q_6^{D-H} = -q_6^{KUKA}$
$q_7^{D-H} = q_7^{KUKA}$
Similarly, the joint velocities and accelerations are:
$\dot{q}_1^{D-H} = \dot{q}_1^{KUKA}$
$\dot{q}_2^{D-H} = \dot{q}_2^{KUKA}$
$\dot{q}_3^{D-H} = \dot{q}_3^{KUKA}$
$\dot{q}_4^{D-H} = -\dot{q}_4^{KUKA}$
$\dot{q}_5^{D-H} = \dot{q}_5^{KUKA}$
$\dot{q}_6^{D-H} = -\dot{q}_6^{KUKA}$
$\dot{q}_7^{D-H} = \dot{q}_7^{KUKA}$

$\ddot{q}_1^{D-H} = \ddot{q}_1^{KUKA}$
$\ddot{q}_2^{D-H} = \ddot{q}_2^{KUKA}$
$\ddot{q}_3^{D-H} = \ddot{q}_3^{KUKA}$
$\ddot{q}_4^{D-H} = -\ddot{q}_4^{KUKA}$
$\ddot{q}_5^{D-H} = \ddot{q}_5^{KUKA}$
$\ddot{q}_6^{D-H} = -\ddot{q}_6^{KUKA}$
$\ddot{q}_7^{D-H} = \ddot{q}_7^{KUKA}$

7.3.2 Experiment

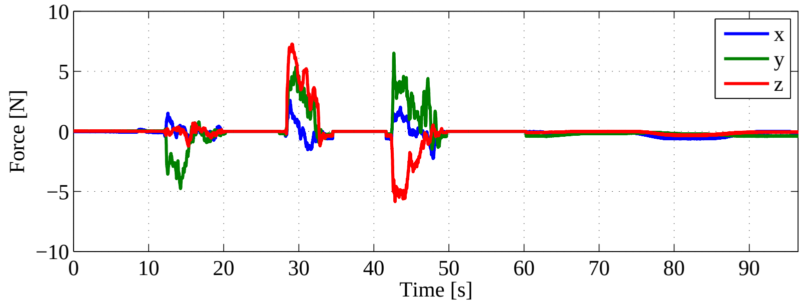
The aim of this section is to show how the force vector estimated by the force/tactile sensor can be used to move the robot in a desired position by exploiting the Admittance control described in Sec. 6 for a manual guidance task while, simultaneously, the sensor can be used as HMI sending to the robot some commands through touch gestures. By combining the two sensor information it is possible to execute more complex robotics task, e.g., a task of intuitive programming.

As in Section 7.2.2 a conformable sensor patch is installed on the third link of the KUKA iiwa (see Fig. 7.1(c)). A system architecture similar to that of the previous case is considered. So, the sensor acquisition board sends over an USB connection the raw data at a sampling frequency rate of 150 Hz to a HostPC running the C++ Sensor Library. The information, then, are sent via an UDP socket to a second PC used to compute the control algorithm. The computed joint commands are sent through the Sunrise.Connectivity FRI Library to the KUKA iiwa controller with a rate of 100 Hz. An ATI F/T Sensor Mini45 is installed on the robot end effector. For the experiment presented below the following values have been selected: $m_e = 15\,\mathrm{kg}$, $d_e = 8\,\mathrm{Ns/m}$, $m_b = 0.5\,\mathrm{kg}$, $d_b = 2\,\mathrm{Ns/m}$, $k_p = 50\,\mathrm{s}^{-2}$ and $k_d = 10\,\mathrm{s}^{-1}$.

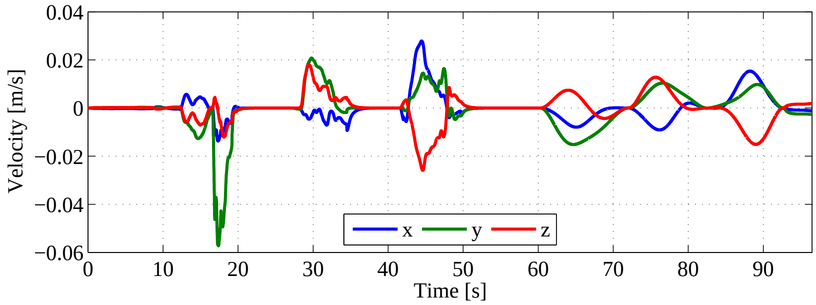
7.3.2.1 Experimental results

The objective of this experiment is to show how the force/tactile sensor can be simultaneously used to intentionally interact with the robot and to communicate with the robot itself through touch gestures (HMI) in a task of intuitive programming. As shown in the previous KUKA case, the high accuracy of the force estimation of the sensor allows to use it in the same way the ATI F/T Sensor Mini45 is used on the robot wrist. The idea is using the F/T Sensor installed on the robot wrist to move the robot end effector in order to reach some desired positions in the Cartesian space. The operator is able to communicate to the robotics system to save those positions by tracing a touch gesture on the force/tactile sensor in order to define a trajectory. The control law described in Sec. 6 has been again considered and it has been implemented in such a way that the position of the robot end effector represents the primary task, while the joint accelerations related to the intentional contacts exercised on the robot body point are projected in the null-space of the primary task. In particular, the desired position of the end effector is computed by using the ATI sensor, while the position of the body point is computed on the basis of the data provided by the force/tactile sensor.

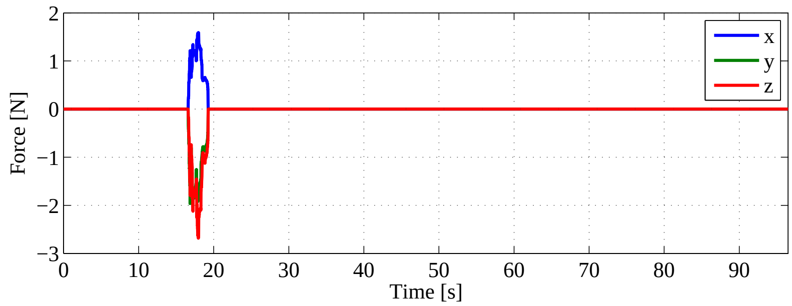
Figure 7.15 reports the results of the experiment. At the beginning of the task execution, an intentional contact is applied to the force/tactile sensor in order to reconfigure the robot in an elbow configuration which is more comfortable for the user that has to trace touch gestures in order to communicate with the robotics system. The contact occurs between 8 s and 10 s and the detected force is reported in Fig. 7.15(c). Thus, the desired motion in Fig. 7.15(d), computed at the contact point, is projected in the null space of the primary task. By applying a force to the sensing module placed in the bottom-right corner of the sensor it is possible to switch between the



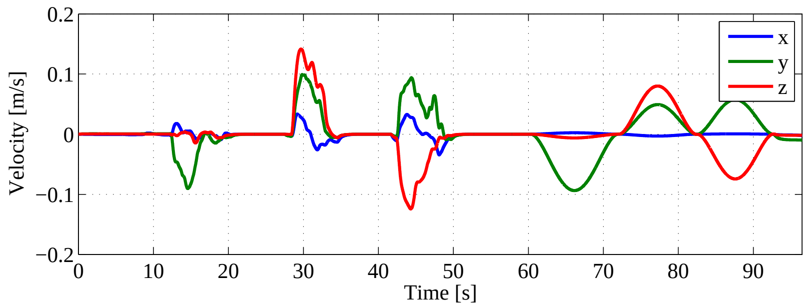
(a) Force components measured with the ATI F/T sensor.



(b) Velocity in Cartesian space of the end effector.



(c) Force components measured with the force/tactile sensor.



(d) Velocity in Cartesian space of the body contact point.

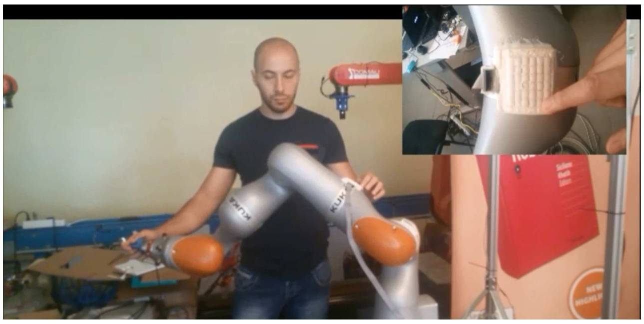
Figure 7.15: KUKA iiwa - experimental results: all components are expressed w.r.t. the robot base frame.

gesture recognition modality and the *manual guidance* modality (Fig. 7.16(a)). In the first 55 s the ATI F/T Sensor is used to move the robot end effector in three different positions of the Cartesian space. The operator communicated to the robot the beginning of the learning process with the *diagonal* touch gesture (Fig. 7.16(b)). Once a desired position is reached, the operator communicates to the robot to save its configuration by tracing an *horizontal line* as touch gesture on the force/tactile sensor (Fig. 7.16(c)). Finally, by tracing the *number 1* as touch gesture (Fig. 7.16(d)) the operator starts the automatic execution of the trajectory saved in the first part of the experiment as shown in last 40 s of Fig. 7.15.

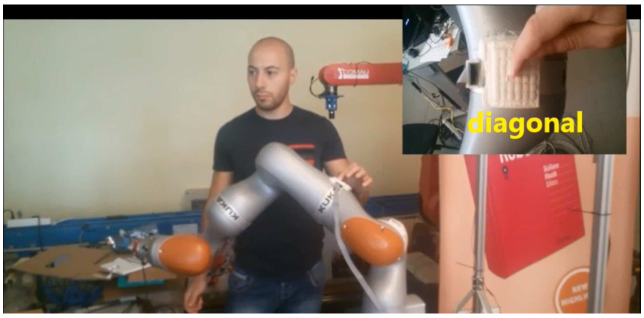
7.4 The sensor on ROS-based platforms: YASKAWA SIA5F and KUKA LWR4+

The choice of the specific integration systems considered in the previous sections are subject to the particular communication interface provided by the robotics systems. The adopted architectures are not flexibly enough to be used in other robotics systems, thus, more or less trivial modifications have to be made in order to adapt them to other platforms and communication interfaces. In general, this is a know issue related to all complex systems obtained by integrating works carried out by individual research groups that work autonomously on different topics. As described in Section 7.1.2 the ROS framework has been developed in the last years trying to make easer and faster the integration of heterogeneous robotics systems and parts. The developed ROS-based driver allow us using the force/tactile sensor with ROS-based robotics platforms in an intuitive and fast way introducing very poor modifications to the robot control software. In this section, the integration of the developed sensor in two ROS-based robotics platforms is described.

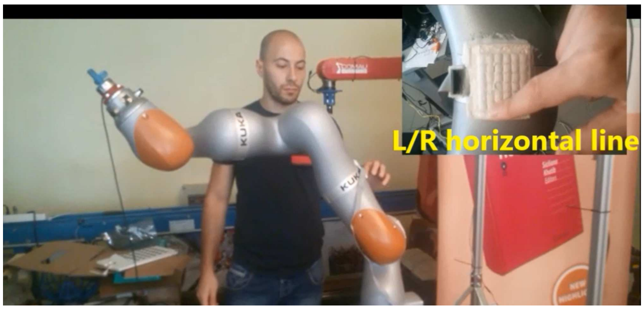
The first platform makes use of an industrial robot manipulator, e.g., the YASKAWA SIA5F. The interface provided with this specific robotics system includes a ROS communication interface that manages the command signals and guarantees a reliable access to the robot control unit through the ROS-Industrial SW stack. ROS-Industrial is an open-source project that extends the advanced capabilities of ROS software to manufacturing. Figure 7.17(a) shows the SW architecture used with the YASKAWA platform. The entire ROS net is executed on a single host PC connected to the force/-tactile sensor (USB interface) and to the robot controller (Ethernet interface). The *control_node* computes the joint positions on the basis of the forces acquired from the *module_force* and *net_force* topics following the admittance control law described



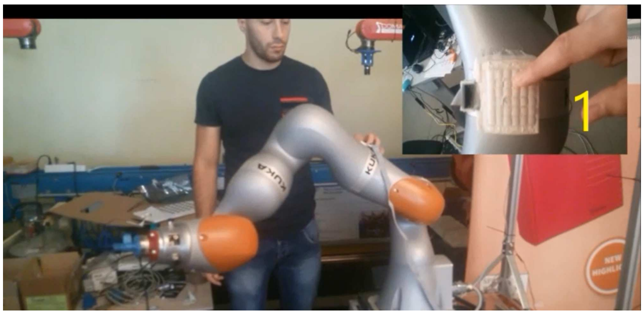
(a) Switching between the *gesture recognition* modality and the *manual guidance* modality.



(b) Starting the learning process.

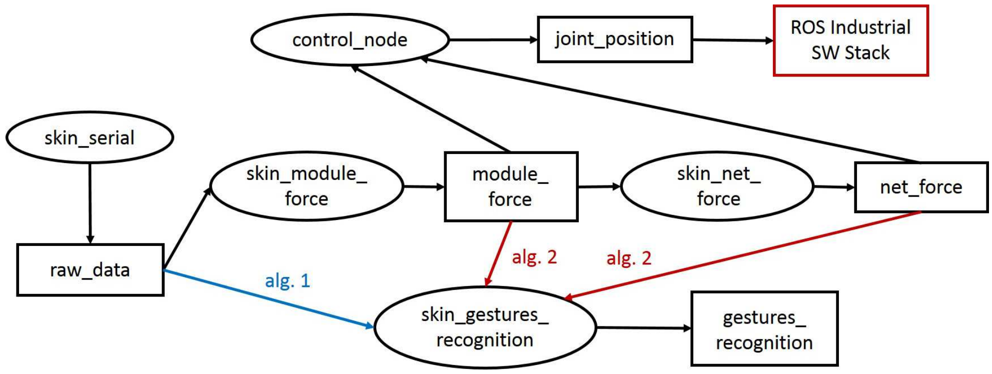


(c) Learning of an end effector positions.

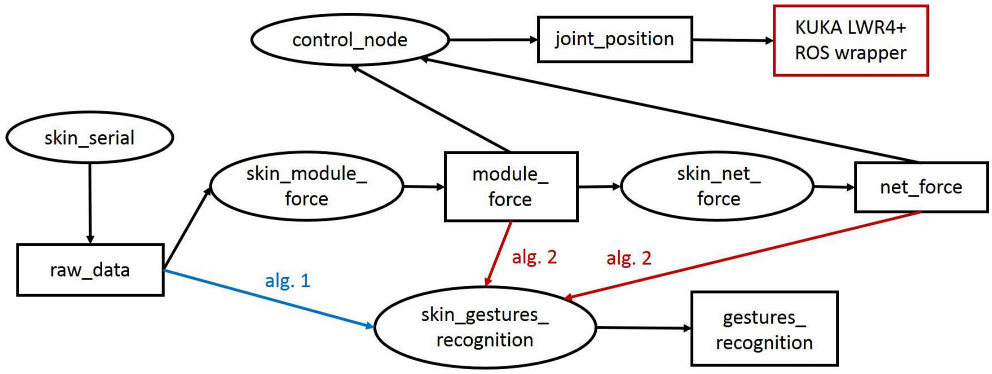


(d) Entering in the *automatic* modality.

Figure 7.16: KUKA iiwa - experimental results: learning process.



(a) SW architecture for YASKAWA SIA5F.



(b) SW architecture for KUKA LWR4+.

Figure 7.17: SW architecture adopted for ROS-based platforms.

in Sec. 6.1, and then it sends them to the ROS-Industrial SW stack.

The second platform uses the KUKA LWR4+ gently provided by TUM and Fig. 7.17(b) shows the SW architecture. Again, the ROS net is executed on a single host PC connected to the force/tactile sensor and to the robot controller. No modifications have been applied to the software layer used to communicate with the sensor. The only part changed in the system is the robot communication interface that, this time, consists of a KUKA FRI ROS wrapper developed by TUM.

As expected the use of a ROS-based sensor driver allows to make easier the integration with any robotics systems that use the ROS framework which requires only trivial adjustments to the interface used to communicate with the robot.

7.4.1 YASKAWA SIA5F D-H table

In this Section, the D-H table of the YASKAWA SIA5F is reported. With reference to Fig. 7.18, the D-H parameters are reported in Tab. 7.4. The relations between the

N. Joint	α_i	a_i [mm]	d_i [mm]	θ_i
1	$-\pi/2$	0	0	q_1
2	$\pi/2$	0	0	q_2
3	$\pi/2$	85	270	q_3
4	$\pi/2$	60	0	q_4
5	$-\pi/2$	0	270	q_5
6	$\pi/2$	0	0	q_6
7	0	0	148	q_7

Table 7.4: YASKAWA SIA5F D-H table.

YASKAWA convention and the D-H convention are: $q_1^{D-H} = q_1^{YASKAWA}$
$q_2^{D-H} = q_2^{YASKAWA}$
$q_3^{D-H} = q_3^{YASKAWA}$
$q_4^{D-H} = q_4^{YASKAWA} - \pi/2$
$q_5^{D-H} = -q_5^{YASKAWA}$
$q_6^{D-H} = q_6^{YASKAWA}$
$q_7^{D-H} = -q_7^{YASKAWA}$

Similarly, the joint velocities and accelerations are: $\dot{q}_1^{D-H} = \dot{q}_1^{YASKAWA}$
$\dot{q}_2^{D-H} = \dot{q}_2^{YASKAWA}$
$\dot{q}_3^{D-H} = \dot{q}_3^{YASKAWA}$
$\dot{q}_4^{D-H} = \dot{q}_4^{YASKAWA}$
$\dot{q}_5^{D-H} = -\dot{q}_5^{YASKAWA}$
$\dot{q}_6^{D-H} = \dot{q}_6^{YASKAWA}$
$\dot{q}_7^{D-H} = -\dot{q}_7^{YASKAWA}$

$\ddot{q}_1^{D-H} = \ddot{q}_1^{YASKAWA}$
$\ddot{q}_2^{D-H} = \ddot{q}_2^{YASKAWA}$
$\ddot{q}_3^{D-H} = \ddot{q}_3^{YASKAWA}$
$\ddot{q}_4^{D-H} = \ddot{q}_4^{YASKAWA}$
$\ddot{q}_5^{D-H} = -\ddot{q}_5^{YASKAWA}$
$\ddot{q}_6^{D-H} = \ddot{q}_6^{YASKAWA}$
$\ddot{q}_7^{D-H} = -\ddot{q}_7^{YASKAWA}$

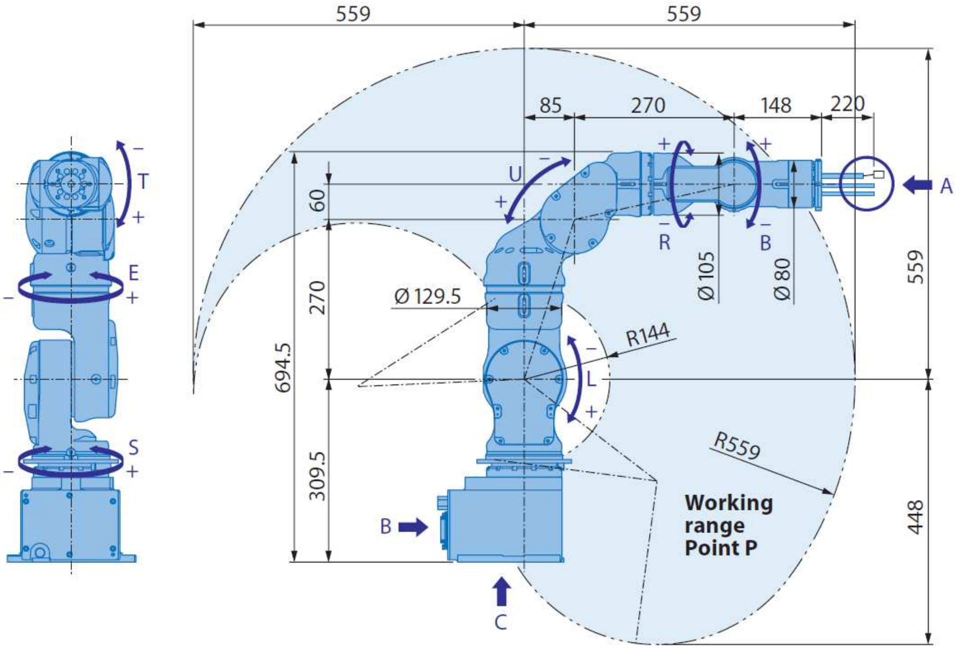


Figure 7.18: YASKAWA SIA5F mechanical scheme.

CHAPTER 8

Conclusions and future works

In this thesis the design and the development of a distributed force/tactile sensor has been described and its tests and usages in physical Human-Robot Interaction applications have been presented. The sensor is constituted by independent sensing modules, able to estimate both normal and shear contact force components. The idea behind the sensor, i.e., adopting four four optoelectronic couples (constituted by an infrared Light Emitting Diodes and a Photo-Detectors) covered by a silicone layer that transduces the external force in a mechanical deformation, represents an optimal choice to obtain a small sensing module able to reconstruct the contact force vector. The latter is estimated as a suitable combination of the four voltage signals measured by the four receivers. In particular, two estimation techniques have been presented, the first one based on a linear calibration model, and the second one based on an ANN. As shown the algorithms guarantee a high estimation accuracy for both the normal and shear components of the force vector. A complete prototype, with a 6×6 matrix of sensing modules, has been realized, characterized and tested. The algorithm used to discriminate multiple contact areas and to estimate the force resultants for each contact area has been described and analyzed.

The sensor, firstly developed in rigid PCB technology, has been re-designed and manufactured in flex PCB technology in order to guarantee mechanical compliant and conformability to curved surfaces, such as robot arms. Guidelines for the installation of the sensor on a generic robotics system are provided. The sensor has been successfully installed on few redundant manipulators of different brands, i.e., KUKA and YASKAWA, and, through the definition of proper system architectures and sensor drivers, it has been exploited in applications of safe physical Human-Robot Interaction, where contact forces over large distributed areas can occur. So, two software drivers has been developed in order to provide to a generic robotic system information such as pressure map, contact forces and contact points. Moreover, three algorithms that allow the use of the sensor as Human-Machine Interface and, then, the recognition of the touch gestures traced on the sensor surface have been designed and implemented. The three algorithms have been assessed in order to evaluate the recognition rate with several tests performed by 20 different performers. By adopting the classic formulation of the admittance control, the sensor has been used in manual guidance, intuitive programming, collision avoidance and reaction tasks. An analysis on the control algorithm stability has been proposed and discussed. The reliability and the robustness of the developed sensor as well as the effectiveness of the proposed method and technology have been showed and demonstrated thought several experiments carried out also in collaboration with PRISMALab of Università di Napoli Federico II and with the DHRI laboratory of TUM.

The described distributed force/tactile sensor results to be a good solution for those applications that require a high accuracy in the force vector estimation, e.g., manual guidance, and it has been shown that the sensor is robust enough to be adopted also in applications where a distributed contact, due to an unintentional collision, occurs.

Future works will be focused on finding new methods and technologies able to speed up the soldering process of the optoelectronic components. Smaller components that include in the same package both the emitter and the receiver could permit the use of automatic processes for the components placement and soldering, e.g., robotized pick and place, as well as they could allow an improvement of the force estimation accuracy and a reduction of time needed for the calibration procedure. Obviously, an interesting challenge will be the development of a new prototype with higher dimensions. In this terms, a new technique will be studied for interconnecting several prototypes together trying to adopt an event-based interrogation strategy in order to improve the sensor acquisition frequency. Moreover, the use of MEMS sensors, e.g., 3 axis accelerometers, for developing of a spatial self-calibration algorithm will be investigated.

APPENDIX A

C++ Sensor Library

Library methods

Skin

Input parameters:

- *skin_type*: integer value. It can be equal to RIGID_SKIN (0) or CONFORMABLE_-SKIN (1)
- *calibration_method*: integer value. It can be equal to NOT_FINE_CALIBRATION (0) or FINE_CALIBRATION (1)
- *gesture_type*: integer value. It can be equal to GESTURE_OFF (0), GESTURE_IMG (1) or GESTURE_CP (2)
- *ip*: char pointer. The IP address of the destination PC
- *port*: integer value. The Port of the destination application

Output parameters:

- void

Description: Class constructor. It properly initializes the support variables, it loads the calibration matrix, the pose of the reference frame of the sensing modules from files, it initializes the gesture recognition modality by setting the selected algorithm and it initializes the communication socket.

setCOM

Input parameters:

- *port*: char pointer. The name of the COM port

Output parameters:

- void

Description: it sets the COM port.

connect

Input parameters:

- void

Output parameters:

- void

Description: it connects the application to the COM port selected with the *setCOM* method.

disconnect

Input parameters:

- void

Output parameters:

- void

Description: it closes the serial connection.

getNormalizationMatrix

Input parameters:

- void

Output parameters:

- void

Description: it reads the normalization parameters from the file. The method is called by the Class constructor only if the NOT_FINE_CALIBRATION method is used.

update

Input parameters:

- void

Output parameters:

- void

Description: it reads the raw data from the sensor. By calling the *getForces* and *getResultantForces* it obtains the forces applied to the sensing modules and to the compact(s) and sends them through the UDP socket initialized by the Class constructor.

removeOffset

Input parameters:

- *numSamples*: integer value. The number of samples to use for the signal offsets computation

Output parameters:

- void

Description: it computes the voltages offset. To estimate properly the applied forces, it must be called before any *update* operation.

getForces

Input parameters:

- *fx*: 6×6 float matrix. The variable returned by the function that contains the x-component of the forces applied to the 36 sensing elements
- *fy*: 6×6 float matrix. The variable returned by the function that contains the y-component of the forces applied to the 36 sensing elements
- *fz*: 6×6 float matrix. The variable returned by the function that contains the z-component of the forces applied to the 36 sensing elements
- *V*: 12×12 float matrix. It contains the 144 voltage signals of the sensor organized in a 12×12 matrix

Output parameters:

- void

Description: it computes the forces applied to the 36 sensing elements on the basis of the voltages signals contained in the input parameter *V*. Returns the three components of the applied forces through the variables *fx*, *fy* and *fz*.

findCompact

Input parameters:

- *y_comp*: vector of 36 integer elements. The variable returned by the function that contains the row indexes of the sensing modules in the found compact
- *x_comp*: vector of 36 integer elements. The variable returned by the function that contains the column indexes of the sensing modules in the found compact
- *index*: integer pointer. The variable returned by the function that contains the number of the sensing elements included in the found compact
- *Mret*: 6×6 integer matrix. The variable returned by the function that contains the updated status of the analyzed sensor
- *M*: 6×6 integer matrix. It contains the status of the sensor to be analyzed. The i, j matrix element is set to 1 if the magnitude of the force applied to the i, j sensing module is greater than a threshold
- *xi*: integer value. The initial index of the column from which to start the analysis of the compact
- *yi*: integer value. The initial index of the row from which to start the analysis of the compact

Output parameters:

- void

Description: starting from the sensing module corresponding to the initial index *xi* and *yi*, it analyzes the matrix *M* in order to find a compact. It is recursively called by the *getResultantForces* method if a RIGID_SKIN is selected.

getResultantForces

Input parameters:

- *Fx*: vector of 36 float elements. The variable returned by the function that contains the x-component of the net force(s) applied to the found compact(s)
- *Fy*: vector of 36 float elements. The variable returned by the function that contains the y-component of the net force(s) applied to the found compact(s)
- *Fz*: vector of 36 float elements. The variable returned by the function that contains the z-component of the net force(s) applied to the found compact(s)
- *contactPoints*: a 36×6 matrix of float elements. The variable returned by the function that contains the x, y, z-coordinate of the contact point(s) in the first three columns and the ϕ, θ, ψ-angles (expressed in the XYZ Euler angles representation) of the reference frame related to the contact point(s)
- *numContacts*: integer pointer. An output variable containing the number of the detected contact regions
- *fx*: 6×6 float matrix. It contains the x-component of the forces applied to the 36 sensing elements
- *fy*: 6×6 float matrix. It contains the y-component of the forces applied to the 36 sensing elements
- *fz*: 6×6 float matrix. It contains the z-component of the forces applied to the 36 sensing elements

Output parameters:

- void

Description: it computes the net forces applied to the compact(s) found through the *findCompact* method on the basis of the forces applied to each sensing modules of the skin sensor. Returns the three components of the computed net forces through the variables *Fx*, *Fy* and *Fz*, the information about the reference frame of the contact regions and the number of the detected contact regions.

APPENDIX B

Example of a ROS sensor driver node

```
#include <stdlib.h>
#include <iostream>
#include <fstream>
#include <fcntl.h>
#include <stdio.h>
#include <string.h>
#include <sstream>
#include <fcntl.h>
#include <errno.h>
#include <termios.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/shm.h>
#include <math.h>

#include "ros/ros.h"
#include "std_msgs/String.h"
#include <skin_driver/raw_data.h>

#define TRANSD_CONST 3.3/4095.0

using namespace std;

int main(int argc, char **argv)
{
    /* Serial variables.*/
    int fd, wr, high = 0, low = 0;
    struct termios options;
```

```
float voltage, pressureMap[12][12], offsetMap[12][12];

/* Message variables. */
skin_driver::raw_data msg;

/* ROS node initialization. */
ros::init(argc, argv, "skin_serial");
ros::NodeHandle n;

/* Publisher. */
ros::Publisher raw_data_pub =
    n.advertise<skin_driver::raw_data>("raw_data", 100);

ros::Rate loop_rate(150); /* 150 Hz */

fd = open("/dev/ttyACM0", O_RDWR | O_NOCTTY | O_NDELAY);

if (fd == -1) {
    cout << "Error: connnection failed\r\n";
}
else {
    fcntl(fd, F_SETFL, 0);
    tcgetattr(fd, &options);
    cfsetispeed(&options, B921600);
    cfsetospeed(&options, B921600);
    options.c_cflag |= (CLOCAL | CREAD);
    tcsetattr(fd, TCSANOW, &options);
    sleep(2);
    tcflush(fd, TCIOFLUSH);
    cout << "Connection successful\r\n";
}

/* Computing voltages offset. */
for(int i = 0; i < 12; i++) {
    for(int j = 0; j < 12; j++) {
        offsetMap[i][j] = 0.0;
    }
}

/* USB data acquisition. */
for(int i = 0; i < 100; i++) {
    wr=write(fd, "a", 1);

    if(wr == 1) {
```

```
            for(int i = 0; i < 72; i++) {
                read(fd, &low, 1);
                read(fd, &high, 1);

                voltage = ((float)(low + (high << 8))) * TRANSD_CONST;
                offsetMap[(i/2)%2 + (4*(i/4))%12][i%2 + 2*(i/12)] += voltage;
            }
            for(int i = 0; i < 72; i++) {
                read(fd, &low, 1);
                read(fd, &high, 1);

                voltage = ((float)(low + (high << 8))) * TRANSD_CONST;
                offsetMap[2 + (i/2)%2 + (4*(i/4))%12][i%2 + 2*(i/12)] += voltage;
            }
        }
    }

    for(int i = 0; i < 12; i++) {
        for(int j = 0; j < 12; j++) {
            offsetMap[i][j] = offsetMap[i][j]/100;
        }
    }
    cout << "Offset removed." << endl << "Skin sensor raw data acquisition.";
    fflush(stdout);

    while (ros::ok()) {
        /* USB data acquisition.*/
        wr=write(fd, "a", 1);
        if(wr == 1) {
            for(int i = 0; i < 72; i++) {
                read(fd, &low, 1);
                read(fd, &high, 1);

                voltage = ((float)(low + (high << 8))) * TRANSD_CONST;
                pressureMap[(i/2)%2 + (4*(i/4))%12][i%2 + 2*(i/12)] = voltage;
            }
            for(int i = 0; i < 72; i++) {
```

```
                    read(fd, &low, 1);
                    read(fd, &high, 1);

                    voltage = ((float)(low + (high << 8))) *
                        TRANSD_CONST;
                    pressureMap[2 + (i/2)%2 + (4*(i/4))%12][i%2 +
                        2*(i/12)] = voltage;
                }

                /* Fill the message to send. */
                for(int i = 0; i < 12; i++) {
                    for(int j = 0; j < 12; j++) {
                        msg.data[i*12+j] = pressureMap[i][j] -
                            offsetMap[i][j];
                    }
                }

                msg.header.stamp = ros::Time::now();
                raw_data_pub.publish(msg);
            }

            cout << ".";
            fflush(stdout);

            loop_rate.sleep();
        }
        cout << endl;
        return 0;
}
```

Bibliography

[1] Brenna D. Argall and Aude G. Billard. A survey of tactile human-robot interactions. *Robotics and Autonomous Systems*, 58:1159–1176, 2010.

[2] EN953. *Safety of Machinery - Guards - General Requirements for the Design and Construction of fixed and movable Guards*, 1997.

[3] Sami Haddadin, Alin Albu-schÃd'ffer, and Gerd Hirzinger. Safety Evaluation of Physical Human-Robot Interaction via Crash-Testing. In *Robotics: Science and Systems*, 2007.

[4] Mallory L. Hammock, Alex Chortos, Benjamin C.-K. Tee, Jeffrey B.-H. Tok, and Zhenan Bao. 25th anniversary article: The evolution of electronic skin (e-skin): A brief history, design considerations, and recent progress. *Advanced Materials*, 25(42):5997–6038, 2013.

[5] Vladimir J. Lumelsky, M.S. Shur, and S. Wagner. Sensitive skin. *Sensors Journal, IEEE*, 1(1):41–51, June 2001.

[6] Fukang Jiang, Yu-Chong Tai, K. Walsh, T. Tsao, Gwo-Bin Lee, and Chih-Ming Ho. A flexible mems technology and its first application to shear stress sensor skin. In *Micro Electro Mechanical Systems, 1997. MEMS '97, Proceedings, IEEE., Tenth Annual International Workshop on*, pages 465–470, Jan 1997.

[7] B. Crone, A. Dodabalapur, Y. Y. Lin, R. W. Filas, Z. Bao, A. LaDuca, R. Sarpeshkar, H. E. Katz, and W. Li. Large-scale complementary integrated circuits based on organic transistors. *Nature*, 403(6769):521–523, February 2000.

[8] I-Chun Cheng and Sigurd Wagner. Overview of flexible electronics technology. In WilliamS. Wong and Alberto Salleo, editors, *Flexible Electronics*, volume 11 of *Electronic Materials: Science & Technology*, pages 1–28. Springer US, 2009.

[9] StÃl'phanie PÃl'richon Lacour, Sigurd Wagner, Zhenyu Huang, and Z. Suo. Stretchable gold conductors on elastomeric substrates. *Applied Physics Letters*, 82(15):2404–2406, 2003.

[10] Sigurd Wagner, StÃl'phanie P. Lacour, Joyelle Jones, Pai hui I. Hsu, James C. Sturm, Teng Li, and Zhigang Suo. Electronic skin: architecture and components. *Physica E: Low-dimensional Systems and Nanostructures*, 25(2âĂŞ3):326 – 334, 2004. Proceedings of the 13th International Winterschool on New Developments in Solid State Physics - Low-Dimensional Systems.

[11] Stephanie P. Lacour, Joyelle Jones, S. Wagner, Teng Li, and Zhigang Suo. Stretchable interconnects for elastic electronic surfaces. *Proceedings of the IEEE*, 93(8):1459–1467, Aug 2005.

[12] Dae-Hyeong Kim, Jizhou Song, Won Mook Choi, Hoon-Sik Kim, Rak-Hwan Kim, Zhuangjian Liu, Yonggang Y. Huang, Keh-Chih Hwang, Yong-wei Zhang, and John A. Rogers. Materials and noncoplanar mesh designs for integrated circuits with linear elastic responses to extreme mechanical deformations. *Proceedings of the National Academy of Sciences*, 105(48):18675–18680, 2008.

[13] Dae-Hyeong Kim, Jianliang Xiao, Jizhou Song, Yonggang Huang, and John A. Rogers. Stretchable, curvilinear electronics based on inorganic materials. *Advanced Materials*, 22(19):2108–2124, 2010.

[14] Takao Someya, Yusaku Kato, Tsuyoshi Sekitani, Shingo Iba, Yoshiaki Noguchi, Yousuke Murase, Hiroshi Kawaguchi, and Takayasu Sakurai. Conformable, flexible, large-area networks of pressure and thermal sensors with organic transistor active matrixes. *Proceedings of the National Academy of Sciences of the United States of America*, 102(35):12321–12325, 2005.

[15] Tsuyoshi Sekitani, Yoshiaki Noguchi, Kenji Hata, Takanori Fukushima, Takuzo Aida, and Takao Someya. A Rubberlike Stretchable Active Matrix Using Elastic Conductors. *Science*, 321(5895):1468–1472, 2008.

[16] Tsuyoshi Sekitani, Tomoyuki Yokota, Ute Zschieschang, Hagen Klauk, Siegfried Bauer, Ken Takeuchi, Makoto Takamiya, Takayasu Sakurai, and Takao Someya. Organic Nonvolatile Memory Transistors for Flexible Sensor Arrays. *Science*, 326(5959):1516–1519, 2009.

[17] Christian Metzger, Elgar Fleisch, Jan Meyer, Mario DansachmÃijller, Ingrid Graz, Martin Kaltenbrunner, Christoph Keplinger, Reinhard SchwÃűdiauer, and Siegfried Bauer. Flexible-foam-based capacitive sensor arrays for object detection at low cost. *Applied Physics Letters*, 92(1):–, 2008.

[18] Ingrid Graz, Martin Kaltenbrunner, Christoph Keplinger, Reinhard SchwÃűdiauer, Siegfried Bauer, StÃl'phanie P. Lacour, and Sigurd Wagner. Flexible ferroelectret field-effect transistor for large-area sensor skins and microphones. *Applied Physics Letters*, 89(7):–, 2006.

[19] Zhibin Yu, Xiaofan Niu, Zhitian Liu, and Qibing Pei. Intrinsically stretchable polymer light-emitting devices using carbon nanotube-polymer composite electrodes. *Advanced Materials*, 23(34):3989–3994, 2011.

[20] Tsuyoshi Sekitani, Hiroyoshi Nakajima, Hiroki Maeda, Takanori Fukushima, Takuzo Aida, Kenji Hata, and Takao Someya. Stretchable active-matrix organic light-emitting diode display using printable elastic conductors. *Nature Materials*, 8(6):494 – 499, 2009.

[21] A.M. Petrina. Advances in robotics (review). *Automatic Documentation and Mathematical Linguistics*, 45(2):43–57, 2011.

[22] Mark H. Lee. Tactile sensing: New directions, new challenges. *The International Journal of Robotics Research*, 19(7):636–643, 2000.

[23] E. Guizzo and E. Ackerman. The rise of the robot worker. *Spectrum, IEEE*, 49(10):34–41, October 2012.

[24] Ming Ying, Andrew P Bonifas, Nanshu Lu, Yewang Su, Rui Li, Huanyu Cheng, Abid Ameen, Yonggang Huang, and John A Rogers. Silicon nanomembranes for fingertip electronics. *Nanotechnology*, 23(34):344004, 2012.

[25] Gregor Schwartz, Benjamin C-K Tee, Jianguo Mei, Anthony L Appleton, Do Hwan Kim, Huiliang Wang, and Zhenan Bao. Flexible polymer transistors with high pressure sensitivity for application in electronic skin and health monitoring. *Nature communications*, 4:1859, 2013.

[26] G. Cannata, M. Maggiali, G. Metta, and G. Sandini. An embedded artificial skin for humanoid robots. In *Proc. of IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 434–438, 2008.

[27] M. Fumagalli, M. Randazzo, F. Nori, L. Natale, G. Metta, and G. Sandini. Exploiting proximal f/t measurements for the icub active compliance. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 1870–1876, Oct 2010.

[28] A. Del Prete, S. Denei, L. Natale, F. Mastrogiovanni, F. Nori, Giorgio Cannata, and G. Metta. Skin spatial calibration using force/torque measurements. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 3694–3700, Sept 2011.

[29] Matteo Fumagalli, Serena Ivaldi, Marco Randazzo, Lorenzo Natale, Giorgio Metta, Giulio Sandini, and Francesco Nori. Force feedback exploiting tactile and proximal force/torque sensing. *Auton Robot*, 33:381–398, 2012.

[30] F. Flacco, T. Kroger, A. De Luca, and O. Khatib. A depth space approach to human-robot collision avoidance. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 338–345, May 2012.

[31] A. De Luca, A. Albu-Schaffer, S. Haddadin, and G. Hirzinger. Collision detection and safe reaction with the dlr-iii lightweight manipulator arm. In *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems 2006 (IROS2006)*, pages 1623–1630, Beijing, China, 2006.

[32] Sami Haddadin, Alin Albu-Schaffer, Alessandro De Luca, and Gerd Hirzinger. Collision detection and reaction: A contribution to safe physical human-robot interaction. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3356–3363, 2008.

[33] F. Flacco and A. De Luca. Integrated control for pHRI: Collision avoidance, detection, reaction and collaboration. In *Proc. of The Fourth IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechatronics*, pages 288–295, 2012.

[34] E. Magrini, F. Flacco, and A. De Luca. Estimation of contact forces using a virtual force sensor. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 2126–2133, Sept 2014.

[35] H. Sadeghian, L. Villani, M. Keshmiri, and B. Siciliano. Task-space control of robot manipulators with null-space compliance. *Robotics, IEEE Transactions on*, 30(2):493–506, April 2014.

[36] F. Ficuciello, A. Romano, L. Villani, and B. Siciliano. Cartesian impedance control of redundant manipulators for human-robot co-manipulation. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 2120–2125, Sept 2014.

[37] F. Ficuciello, L. Villani, and B. Siciliano. Variable impedance control of redundant manipulators for intuitive human-robot physical interaction. *Robotics, IEEE Transactions on*, 31(4):850–863, Aug 2015.

[38] H. Alirezaei, A. Nagakubo, and Y. Kuniyoshi. A tactile distribution sensor which enables stable measurement under high and dynamic stretch. In *3D User Interfaces, 2009. 3DUI 2009. IEEE Symposium on*, pages 87–93, March 2009.

[39] T. Mukai, M. Onishi, T. Odashima, S. Hirano, and Zhiwei Luo. Development of the tactile sensor system of a human-interactive robot (ri-man). *Robotics, IEEE Transactions on*, 24(2):505–512, April 2008.

[40] D. Goĺger, N. Gorges, and Heinz Worn. Tactile sensing for an anthropomorphic robotic hand: Hardware and signal processing. In *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pages 895–901, May 2009.

[41] Z. Chu, P.M. Sarro, and S. Middelhoek. Silicon three-axial tactile sensor. In *Solid-State Sensors and Actuators, 1995 and Eurosensors IX.. Transducers '95. The 8th International Conference on*, volume 1, pages 656–659, Jun 1995.

[42] A. Schmitz, M. Maggiali, L. Natale, and G. Metta. Touch sensors for humanoid hands. In *RO-MAN, 2010 IEEE*, pages 691–697, Sept 2010.

[43] A. Schmitz, M. Maggiali, L. Natale, B. Bonino, and G. Metta. A tactile sensor for the fingertips of the humanoid robot icub. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 2212–2217, Oct 2010.

[44] F. Cuellar, T. Yamamoto, and H. Ishiguro. Design and development of a low power tactile multi-sensor network for robotic systems. In *Mechatronics and Automation (ICMA), 2014 IEEE International Conference on*, pages 331–336, Aug 2014.

[45] A.P. Gerratt, N. Sommer, S.P. Lacour, and A. Billard. Stretchable capacitive tactile skin on humanoid robot fingers - first experiments and results. In *Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on*, pages 238–245, Nov 2014.

[46] C. Melchiorri, L. Moriello, G. Palli, and U. Scarcia. A new force/torque sensor for robotic applications based on optoelectronic components. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 6408–6413, May 2014.

[47] R.S. Dahiya, P. Mittendorfer, M. Valle, G. Cheng, and V.J. Lumelsky. Directions toward effective utilization of tactile skin: A review. *Sensors Journal, IEEE*, 13(11):4121–4138, Nov 2013.

[48] N. Elkmann, M. Fritzsche, and E. Schulenbur. Tactile sensing for safe physical human-robot interaction. In *Proc. of International Conference on Advances in Computer-Human Interactions*, 2011.

[49] V. Duchaine, N. Lauzier, M. Baril, M. A. Lacasse, and C. Gosselin. A flexible robot skin for safe physical human robot interaction. In *Proc. of IEEE International Conference on Robotics and Automation*, pages 3676–3681, 2009.

[50] X. Lamy, F. Colledani, F. Geffard, Y. Measson, and G. Morel. Robot skin structure and performances for industrial robot comanipulation. In *Proc. of IEEE International Conference on Advanced Intelligent Mechatronics*, pages 427–432, 2009.

[51] M. W. Strohmayr, H. Worn, and G. Hirzinger. The DLR artificial skin step I: Uniting sensitivity and collision tolerance. In *Proc. of IEEE International Conference on Robotics and Automation*, pages 1004–1010, 2013.

[52] J. Ulmen and M. Cutkosky. A robust, low-cost and low-noise artificial skin for human-frindly robots. In *Proc. of IEEE International Conference on Robotics and Automation*, 2010.

[53] Y. Ohmura, Y. Kuniyoshi, and A. Nagakubo. Conformable and scalable tactile sensor skin for curved surfaces. In *Proc. of IEEE International Conference on Robotics and Automation*, pages 1348–1353, 2006.

[54] T. Hoshi and H. Shinoda. Robot skin based on touch-area-sensitive tactile element. In *Proc. of IEEE International Conference on Robotics and Automation*, pages 3463–3468, 2006.

[55] T. Takeshita, Y. Peng, N. Morita, H. Ando, and R. Higurashi, E. amd Sawada. Characteristics of a monolithically integrated micro-displacement sensor. In *Joint International IMEKO TC1+ TC7+ TC13 Symposium*, 2011.

[56] J. Missinne, E. Bosman, B. Van Hoe, G. Van Steenberge, S. Kalathimekkad, P. Van Daele, and Vanfleteren J. Flexible shear sensor based on embedded optoelectronic components. In *Proc. of IEEE Photonics Technology Letters*, volume 23, 2011.

[57] Takuma Iwasaki, Toshihiro Takeshita, Yuji Arinaga, Koji Uemura, Hideyuki Ando, Satoshi Takeuchi, Masutaka Furue, Eiji Higurashi, and Renshi Sawada. Shearing force measurement device with a built-in integrated micro displacement sensor. *Sensors and Actuators A: Physical*, 221:1 – 8, 2015.

[58] L. Ascari, P. Corradi, L. Beccai, and C. Laschi. A miniaturized and flexible optoelectronic sensing system for tactile skin. *Journal of Micromechanics and Microengineering*, 17, 2007.

[59] Ting Zhang, Hong Liu, Li Jiang, Shaowei Fan, and Jing Yang. Development of a flexible 3-d tactile sensor system for anthropomorphic artificial hand. *IEEE Sensors Journal*, 13(2):510–518, 2013.

[60] Ting Zhang, Li Jiang, Xinyu Wu, Wei Feng, Dingjiang Zhou, and Hong Liu. Fingertip three-axis tactile sensor for multifingered grasping. *Mechatronics, IEEE/ASME Transactions on*, 20(4):1875–1885, Aug 2015.

[61] YWR Amarasinghe, AL Kulasekera, and TGP Priyadarshana. Quantum tunneling composite (QTC) based tactile sensor array for dynamic pressure distribution measurement. In *2013 Seventh International Conference on Sensing Technology (ICST)*, pages 1–4, 2013.

[62] Tao Liu, Yoshio Inoue, and Kyoko Shibata. A small and low-cost 3-d tactile sensor for a wearable force plate. *IEEE Sensors Journal*, 9(9):1103–1110, 2009.

[63] R. Bekhti, V. Duchaine, and P. Cardou. Miniature capacitive three-axis force sensor. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 3939–3946, Sept 2014.

[64] A. Narendiran and B. George. Capacitive tactile sensor with slip detection capabilities for robotic applications. In *Instrumentation and Measurement Technology Conference (I2MTC), 2015 IEEE International*, pages 464–469, May 2015.

[65] K. O. Johanson and J. R. Phillips. Tactile spatial resolution. i. two-points discrimination, gap detection, grating resolution, and letter recognition. *Journal of Neurophsiology*, 46:1177–1192, 1981.

[66] J. C. Craig. Grating orientation as a measure of tactile spatial acuity. *Somatosensory and Motor Res.*, 16:197–206, 1999.

[67] J. C. Craig. Factors affecting tactile spatial acuity. *Somatosensory and Motor Res.*, 15:29–45, 1998.

[68] J. C. Craig and K. B. Lyl. A comparison of tactile spatial sensitivity on the palm and fingerpad. *Perception and Phychophysics*, 63:337–347, 2001.

[69] J. B. F. Van Erp. Vibrotactile spatial acuity on the torso: effect of location and timing parameters. In *1st Joint Eurohaptics Conf. on Haptic Interfaces for Virtual Environment and Teleoperator Sys.*, pages 80–85, 2005.

[70] A. Cirillo, P. Cirillo, G. De Maria, C. Natale, and S. Pirozzi. An artificial skin based on optoelectronic technology. *Sensors & Actuators: A. Physical*, 212:110–122, 2014.

[71] G. De Maria, C. Natale, and S. Pirozzi. Force/tactile sensor for robotic applications. *Sensors and Actuators A: Physical*, 175:60–72, 2012.

[72] A. Cirillo, P. Cirillo, and S. Pirozzi. A modular and low-cost artificial skin for robotic applications. In *Proc. of 4th IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechatronics*, pages 961–966, Rome, Italy, 2012.

[73] A. Cirillo, P. Cirillo, G. De Maria, C. Natale, and S. Pirozzi. A proximity/contact-force sensor for human safety in industrial robot environment. In *Proc. of IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pages 1272–1277, Wollongong, Australia, 2013.

[74] A. D'Amore, G. De Maria, L. Grassia, C. Natale, and S. Pirozzi. Silicone-rubber-based tactile sensors for the measurement of normal and tangential components of the contact force. *Journal of Applied Polymer Science*, 122:3758–3770, 2011.

[75] A. Cirillo, P. Cirillo, G. De Maria, C. Natale, and S. Pirozzi. A FE analysis of a silicone deformable interface for distributed force sensors. In *7th International Conference on Times of Polimers & Composites*, 2014.

[76] R. S. Dahiya, G. Metta, M. Valle, and G. Sandini. Tactile sensing - from humans to humanoids. *IEEE Transactions on Robotics*, 26:1–20, 2009.

[77] T. Horii, Y. Nagai, L. Natale, F. Giovannini, G. Metta, and M. Asada. Compensation for tactile hysteresis using gaussian process with sensory markov property. In *14th IEEE/RAS International Conference on Humanoid Robots*, pages 993–998, Madrid, 2014.

[78] V. K. Thakur. Robust hand gesture recognition for human machine interaction system. *Journal of Global Research in Computer Science*, 5:14–19, 2014.

[79] Reza Azad, Babak Azad, Nabil Belhaj Khalifa, and Shahram Jamali. Real-time human-computer interaction based on face and hand gesture recognition. *International Journal in Foundations of Computer Science & Technology*, 4:37–48, 2014.

[80] Dan Xu, Xinyu Wu, Yen-Lun Chen, and Yangsheng Xu. Online dynamic gesture recognition for human robot interaction. *Journal of Intelligent & Robotic Systems*, 77:583–596, 2015.

[81] L. Brethes, P. Menezes, F. Lerasle, and J. Hayet. Face tracking and hand gesture recognition for human-robot interaction. In *IEEE International Conference on Robotics and Automation*, pages 1901–1906, 2004.

[82] E. Sánchez-Nielsen, L. Antón-Canalís, and M. Hernández-Tejera. Hand gesture recognition for human-machine interaction. *Journal of WSCG*, 12(1-3):395–402, 2004.

[83] F.A Bertsch and V.V. Hafner. Real-time dynamic visual gesture recognition in human-robot interaction. In *9th IEEE-RAS International Conference on Humanoid Robots*, pages 447–453, 2009.

[84] T. Fujii, J.H. Lee, and S. Okamoto. Gesture recognition system for human-robot interaction and its application to robotic service task. In *Proceedings of the International MultiConference of Engineers and Computer Scientists*, 2014.

[85] R. Stiefelhagen, H.K. Ekenel, C. Fugen, P. Gieselmann, H. Holzapfel, F. Kraft, K. Nickel, M. Voit, and A. Waibel. Enabling multimodal humanŰrobot interaction for the karlsruhe humanoid robot. *IEEE Transactions on Robotics*, 23(5):840–851, 2007.

[86] Vassilis Pitsikalis, Athanasios Katsamanis, Stavros Theodorakis, and Petros Maragos. Multimodal gesture recognition via multiple hypotheses rescoring. *Journal of Machine Learning Research*, 16(5):255–284, 2015.

[87] L. Lucignano, F. Cutugno, S. Rossi, and A. Finzi. A dialogue system for multimodal human-robot interaction. In *Proceedings of the 15th ACM on International Conference on Multimodal Interaction*, pages 197–204, 2013.

[88] R. Stiefelhagen, C. Fugen, R. Gieselmann, H. Holzapfel, K. Nickel, and A. Waibel. Natural human-robot interaction using speech, head pose and gestures. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2422–2427, 2004.

[89] L. Muscari, L. Seminara, F. Mastrogiovanni, M. Valle, M. Capurro, and G. Cannata. Real-time reconstruction of contact shapes for large area robot skin. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 2360–2366, May 2013.

[90] E. Baglini, S. Youssefi, F. Mastrogiovanni, and G. Cannata. A real-time distributed architecture for large-scale tactile sensing. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 1663–1669, Sept 2014.

[91] H. Iwata and S. Sugano. Human-robot-contact-state identification based on tactile recognition. *Industrial Electronics, IEEE Transactions on*, 52(6):1468–1477, Dec 2005.

[92] Ze Ji, F. Amirabdollahian, D. Polani, and K. Dautenhahn. Histogram based classification of tactile patterns on periodically distributed skin sensors for a humanoid robot. In *RO-MAN, 2011 IEEE*, pages 433–440, July 2011.

[93] Sven Parusel, Hannes Widmoser, Saskia Golz, Tobias Ende, Nico Blodow, Matteo Saveriano, Kai Krieger, Alexis Maldonado, Ingo Kresse, Roman Weitschat, Dongheui Lee, Michael Beetz, and Sami Haddadin. Human-robot interaction planning. AAAI 2014 Video Competition, 2014.

[94] C. Zhu and Weihua Sheng. Online hand gesture recognition using neural network based segmentation. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 2415–2420, Oct 2009.

[95] P. Neto, D. Pereira, J. Norberto Pires, and A.P. Moreira. Real-time and continuous hand gesture spotting: An approach based on artificial neural networks. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 178–183, May 2013.

[96] Nhan Nguyen-Duc-Thanh, Sungyoung Lee, and Donghan Kim. Two-stage hidden markov model in gesture recognition for human robot interaction. *Int J Adv Robot Syst*, 39(9):1468–1477, 2012.

[97] R. Amstutz, O. Amft, B. French, A. Smailagic, Dan Siewiorek, and G. Troster. Performance analysis of an hmm-based gesture recognition using a wristwatch device. In *Computational Science and Engineering, 2009. CSE '09. International Conference on*, volume 2, pages 303–309, Aug 2009.

[98] M. Meenaakumari and M. Muthulakshm. Mems accelerometer based hand gesture recognition. *International Journal of Advanced Research in Computer Engineering & Technology*, 2(5):1886–1892, May 2013.

[99] S. Mitra and T. Acharya. Gesture recognition: A survey. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 37(3):311–324, May 2007.

[100] S.T. Bow. *Pattern Recognition and Image Preprocessing*. CRC press, 2002.

[101] George Wolberg. *Digital Image Warping*. IEEE Computer Society Press, 1994.

[102] Philippe Thévenaz, Thierry Blu, and Michael Unser. Handbook of medical imaging. chapter Image Interpolation and Resampling, pages 393–420. Academic Press, Inc., Orlando, FL, USA, 2000.

[103] F. Flacco and A. De Luca. Integrated control for phri: Collision avoidance, detection, reaction and collaboration. In *Proc. of The Fourth IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechatronics*, pages 288–Ű295, Roma, 2012.

[104] E. Magrini, F. Flacco, and A. De Luca. Estimation of contact forces using a virtual force sensor. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2126Ű–2133, Chicago, 2014.

[105] B. Siciliano and J.-J.E. Slotine. A general framework for managing multiple tasks in highly redundant robotic systems. In *Advanced Robotics, 1991. 'Robots in Unstructured Environments', 91 ICAR., Fifth International Conference on*, pages 1211–1216 vol.2, June 1991.

[106] F. Flacco, A. De Luca, and O. Khatib. Prioritized multi-task motion control of redundant robots under hard joint constraints. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 3970–3977, Oct 2012.

[107] H. Sadeghian, L. Villani, M. Keshmiri, and B. Siciliano. Dynamic multi-priority control in redundant robotic systems. *Robotica*, 31(7):1155–1167, 2013.

[108] Pietro Falco and Ciro Natale. Low-level flexible planning for mobile manipulators: a distributed perception approach. *Advanced Robotics*, 28(21):1431–1444, 2014.

[109] Anthony A. Maciejewski and Charles A. Klein. Numerical filtering for the operation of robotic manipulators through kinematically singular configurations. *Journal of Robotic Systems*, 5:527–552, 1988.

[110] F. Caccavale, S. Chiaverini, and B. Siciliano. Second-order kinematic control of robot manipulators with jacobian damped least-squares inverse: theory and experiments. *Mechatronics, IEEE/ASME Transactions on*, 2(3):188–194, Sep 1997.

[111] C. Natale, B. Siciliano, and L. Villani. Spatial impedance control of redundant manipulators. In *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, volume 3, pages 1788–1793 vol.3, 1999.

[112] Fast Research Interface Library documentation. http://cs.stanford.edu/people/tkr/fri/html/.