Development of Hardware and Algorithms for a Multi-Functional Capacitive Proximity Sensing System

by

Fan Xia

M.A.Sc., Tianjin University, 2014

B.Sc., Tianjin University, 2011

Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of
Doctor of Philosophy

in the
School of Mechatronic Systems Engineering
Faculty of Applied Science

© Fan Xia 2018

SIMON FRASER UNIVERSITY

Fall 2018

Copyright in this work rests with the author. Please ensure that any reproduction or re-use is done in accordance with the relevant national copyright legislation.

Approval

Name:	**Fan Xia**
Degree:	**Doctor of Philosophy (Mechatronic Systems Engineering)**
Title:	**The Title: Development of Hardware and Algorithms for a Multi-Functional Capacitive Proximity Sensing System**
Examining Committee:	**Chair: Amr Marzouk** Lecturer

Behraad Bahreyni
Senior Supervisor
Associate Professor

Marek Syrzycki
Supervisor
Professor

Michael Adachi
Supervisor
Assistant Professor

Carlo Menon
Internal Examiner
Professor

Pedram Mousavi
External Examiner
Professor
Mechanical Engineering
University of Alberta

Date Defended/Approved: 05 November 2018

Abstract

This thesis focuses on the development of a multi-functional capacitive proximity sensor to improve the worker safety during the industrial human-robot interactions. The sensor is to be mounted on the worker and used to maintain a safe distance between the worker and robot or the parts moved by the robot. The response of a capacitive proximity sensor is a function of the actual distance as well as the geometry of the approaching object. This uncertainty can lead to a wrong estimation of distance or possibly a missed detection. The proposed sensing system in this work aims to solve this issue.

Three sensing capabilities, namely distance measurement, surface profile recognition, and parallel motion tracking are implemented in a single platform. These capabilities are achieved by a capacitive sensing element coupled to reprogrammable interface electronics. The sensing element features a 4×4 matrix of electrodes that can be reconfigured to different arrangements at run-time to obtain information on the desired parameters of interest (i.e., distance, shape, and trajectory). The control modules are mapped on a field programmable gate array while the capacitance generated by each configuration of electrodes is measured and quantized by a capacitance-to-digital chipset. Digital filters are used to pre-process the raw capacitive data in order to compensate for random walk and environmental interferences such as temperature and humidity variations. Statistical learning methodologies are applied to classify objects and calculate distance values. Quantitative regression models are built to seek out distance values while classification tools including K nearest neighbors, neural network, and support vector machine are employed to recognize the surface profiles.

The performance of the sensing modalities is experimentally assessed with lab equipment as well as on an industrial robot. The system can detect objects and classify their geometries at distances up to about 20 cm with high accuracy. Three different surface profiles can be recognized by all the classifiers. Recognizing the shape of the object improved the regression models and reduced the close-distance measurement error by a factor of five compared to methods that did not take the geometry into account. The capability of tracking the parallel motion is demonstrated by combining the capacitive responses from different electrode connection configurations. The breakthroughs made through this work will make capacitive sensing a viable low-cost alternative to existing technologies for proximity detection in robotics and other fields.

Keywords: Capacitive sensing; distance measurement; surface profile recognition; parallel motion tracking; regression; classification

Acknowledgements

I would not have been able to make it to this point without the support of my seiner supervisor Dr. Behraad Bahreyni. His wise suggestions, generous help, and patience helped me a lot during my Ph.D. studies. In addition to his thoughtful guidance throughout my research work, he taught me a lot about technical writing and technical presentation skills. I also want to express my thanks to Dr. Marek Syrzycki and Dr. Michael Adachi for being my committee members. Their comments and technical suggestions through my research progress are helpful to me.

I would also like to thank all the members of the Intelligent Sensing Laboratory. Without their support and generous help, I would not be able to reach this point. Further, I want to thank faculties and staffs in both School of Engineering Science and Mechatronic Systems Engineering at SFU, especially Chao Cheng and Aladdin Tayeh for their help in technical problems solving. Also, I thank all the graduate students who had worked with me. I cannot list every name as too many of them helped me in my studies as well as in my researches.

I appreciate the support and the generous help from the Menrva Research Group, especially the technical support from Umme Zakia on programming the industrial robot. Without her great work, I would not be able to verify my sensing system in a practical scenario with a real robot.

I would like to thank all my friends who gave me the energy and the emotional supports all the time, I am so grateful to have them. Finally, I want to thank my beloved parents and sister for their endless love and support. My life will never be as happy without them.

Lastly, I thank my love, Shengkai, for his emotional support, sincere suggestions, and technical advice. Thank him for loving me even when I lose the faith.

Table of Contents

List of Tables

List of Figures

List of Acronyms

HRI	Human-Robot Interaction
FEM	Finite Element Method
PCB	Print Circuit Board
CDC	Capacitance to Digital Converter
FPGA	Field Programmable Gate Array
PCA	Principal Component Analysis
SSE	Sum of Squares of Errors
R^2	R square
ML	Machine Learning
KNN	K Nearest Neighbors
NN	Neural Network
SVM	Support Vector Machine
CV	Cross Validation
VLSI	Very Large Scale Integrated
VHDL	VHSIC Hardware Description Language
CE	Cross Entropy
OAO	One Against One

Chapter 1.

Introduction

1.1. Background

Robots are used in increasing numbers in the workspace and in society in general. From the industrial perspective, specialized robots are critical components in a variety of fully automated manufacturing processes due to their high strength, high positional precision, repeatability, and durability. According to the statistics of industrial robot installation from International Federation of Robotics (IFR): the demand for industrial robots has accelerated considerably due to the ongoing trend toward automation and continued innovative technical improvements in industrial robots since 2010 [1]. In 2016, robot sales increased by 16% to 294,312 units, a new peak for the fourth year in a row. However, the author in [2] provides a comprehensive overview of emerging technologies in automotive assembly. Their results show that mass customization with the smaller lot sizes of the products requires high technological flexibility. So that the demands for increased adaptability, flexibility, and reusability of a manufacturing system are rising continuously [3]. In spite of significant advances in the field of industrial automation, human intelligence is far superior in terms of reasoning, comprehension, vision, and ingenuity. Robots and humans present complementary features for the development of manufacturing process. Recently, researchers propose various designs to cope with this by integrating both automated and human-based manufacturing processes together. Therefore, a robot-assisted but human guided manufacturing shows various advantages compared to full automation. Thus the close cooperation of human and machine is highly demanded [4].

Combining the positive points of both manual and automated work-cells enables new flexible systems and opens up new application scopes. One way to realize such a collaboration is Human-Robot Interaction (HRI). HRI is currently a very extensive and diverse research and design activity in both academia and industry. It is a field dedicated to understanding, designing, and evaluating robotic systems that are used by or with humans. In general, the interaction can be separated into two categories depending on whether the human and the robot are in close proximity to each other: remote interaction

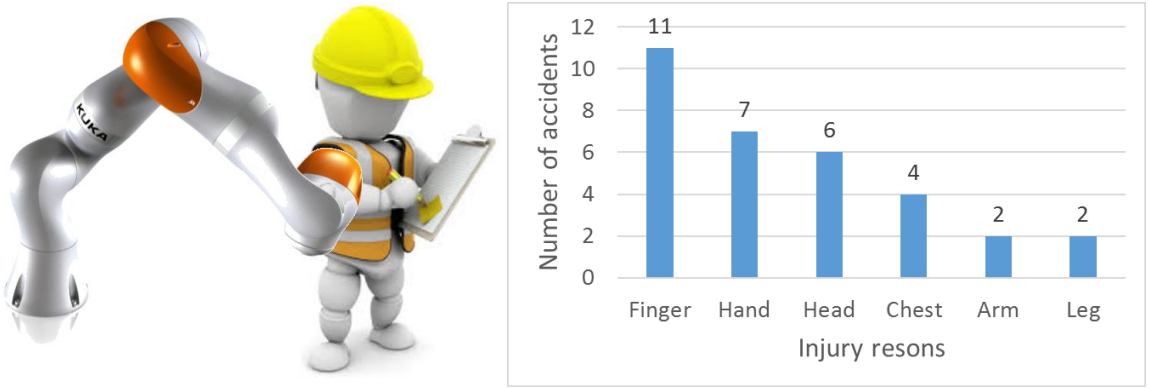
and proximate interaction [5]. The human supervisory control of robots in the performance of routine tasks falls to the second category, the proximate interaction that may involve physical interactions. There is a huge variety of robots doing industrial tasks, and the performed tasks include handling of parts on manufacturing assembly lines, accessing and delivery of packages and components in working spaces [6]. The concept of robots collaborating with human workers in manufacturing processes dates back to 1999 [7], the collaboration can reduce ergonomic concerns while improving the productivity. The hybrid human-machine collaboration is a hot topic in multiple industrial applications, at the same time, more advanced collaborations are highly required.

1.2. Motivation

Popular notions of robotics have long foreseen humans and robots existing side by side, sharing the same work, and integrating to a greater whole. Since the appearance of the industrial robotics, a great deal of attention has been paid to the human workers' safety because these robots usually have fairly large dimensions, are heavy and operate at high velocities. Therefore, collisions between humans and industrial robotic manipulators might be extremely dangerous for humans and they must be completely avoided. However, data on industrial robot-related fatalities indicate that safety is not a solved problem, especially because the human operators are by necessity physically close to the mechanical arm or vehicle [8]. Until today, there exists a notable lack in the "partnership" between humans and robotic manipulators: the machine "intelligence" is quite limited or even unavailable to enable a proper "co-operation" [9].

Robots are generally set up for an operation by the teach-and-repeat technique. In this mode, the trained programmer typically uses a control device to teach a robot its tasks manually. The study carried out in [10] indicates that many robot accidents do not occur under normal operating conditions, but during programming, maintenance, repair, testing, refinement, setup, or adjustment period. During these scenarios, human workers may temporarily be within the robot's working envelope as shown in Figure 1.1 (a) where unintended operations might result in injuries. In the same study, the typical robotic accidents can be categorized into four groups: Impact or collision accidents that represent unpredicted movements or component malfunctions that result in physical contact accidents. Crushing and tapping accidents that an individual can be trapped or be

physically driven into and crushed by the equipment. Mechanical accidents meaning the breakdown of the robot or its components. And finally other potential accidents.



(a) (b)

Figure 1.1. (a) Hazard happens when a human worker is within a robot's working envelop. (b). Breakdown of 32 robotic accidents by types of injury [12].

The hazards of the robot to human can be expected with several variations include human errors, control errors, unauthorized accesses, mechanical failures, environmental sources, improper installations and issues in power systems. In fact, workers who are operating and maintaining automated machinery are at high risk of serious injuries. US statistics suggest that every year, there are more than 18,000 amputations are attributable to such close human-robot interactions [11]. A comprehensive report presented in [12] analyzed 32 accidents and provided an understanding of the most common reasons for the injuries. Pinch injury that accounts for 56% of the total injuries occurs when a robot traps a worker between itself and an object. Whereas impact injuries happen when a robot and a worker collide account for the rest 44% of the injuries. Moreover, this report revealed the most frequently injured human body parts as shown in Figure 1.1 (b). It is observed that fingers are subject to injuries the most in 36.3% of the reported accidents, followed by the hands that account for 21.2%. This directly corresponds to the way a person works in the proximity to a robot.

Most of the early techniques to the safety problems are based on isolating the robots and the humans with physical barriers [13] which stop the robot immediately when crosses by a human. However, the segregation paradigm limits the flexibility of the tasks. In fact, this kind of solutions fails in cases where the human and the robot have to share the physical environment and the successful completion of a task depends on the

collaboration. These drawbacks have triggered the development of new safety standards that permit the coexistence of humans and robots in the same workspace under certain circumstances. The first solid step toward ensuring the safety of human workers in human-robot interaction systems is proposed in the standard ISO 10218-1 and ISO 10218-2 [14] which were published in 2011. These documents provide supplemental and supporting information to the industrial robot safety standards. A new ISO 15066 [15] technical specification for collaborative robot system safety pushes the boundary of safe distance limits and allows people and robots working more closely together. These standardization efforts demonstrate the increasing interest of industries in human-robot collaboration.

Plenty of approaches have been developed in previous research in order to implement safety in human-robot interaction systems. Up to now, collision detection and motion re-planning represent the best choice in collaborative workspaces to guarantee the safety. Collision avoidance is basically a set of procedures for rearranging the robot motion in the presence of dynamic obstacles, especially humans. Such procedures, therefore, involve both environmental monitoring and the control of the robot motion. More specifically, pre-collision planners employ sensing systems to monitor the adjacent area, and trigger the power brake of the robotic manipulators or modify their trajectories depending on the presence and features of the approaching object.

In such an articulated scenario, our research narrows down to focus on how to safely monitor and track the position and movement of humans within a collaboration workspace. The aim of our proposed project is to develop a multi-functional capacitive proximity sensory system, to improve the safety of human-robot interactions in the automotive manufacturing environment. More specifically, characteristics ranging from reconstructing the surface profile of an approaching object to tracking the movement of a human worker are intended to be achieved in the single sensing system. In this way, the ambiguities of the approaching object can be reduced that a wrong distance estimation or possibly a missed detection can be eliminated. Ultimately, it is intended to integrate the sensor into worker garment along with the sensor that monitors the loads on worker bodies.

1.3. Organization of the thesis

This thesis is divided into six chapters. Chapter 2 reviews the related work in the literature on approaches to human-robot interaction safety. An overview of principle properties of proximity sensors that are the critical components of the safety approaches is provided, followed by a detailed discussion of the capacitive-based proximity sensors. Recent works for pattern recognition are also presented.

In Chapter 3, the theoretical basis on the electric field sensing is described. The proposed sensor structure is modeled with Finite Element methods (FEM). The design concerns and simulation results are provided, as well as a detailed explanation of the sensor's working mode.

Chapter 4 presents the involved data processing procedure to extract desired information from the sensing responses. The sensory system is described. Followed by the discussion of suitable statistical methodologies for each data processing stage including raw data pre-processing, distance estimation, and surface profile recognition.

Chapter 5 details the final hardware implementation and the performance estimation. The experimental setups are described and the multiple sensing capabilities are experimentally assessed. The testing results, as well as the performance of the statistical learning methods, are provided.

The final chapter overviews the main contributions of the thesis and provides concluding remarks about the proposed sensing system. Directions for future work are also outlined.

Chapter 2.

Literature review

This chapter overviews the existing works in solving the human-robot interaction safety issues. To begin, Section 2.1 reviews broadly the current strategies for improving the safety. Then the topics of the researches are narrowed down to the specific emphasis of this work: the proximity sensors and the capacitive-based sensors. In addition, the methodologies for the proposed sensing modalities are investigated.

2.1. Human-robot interaction safety

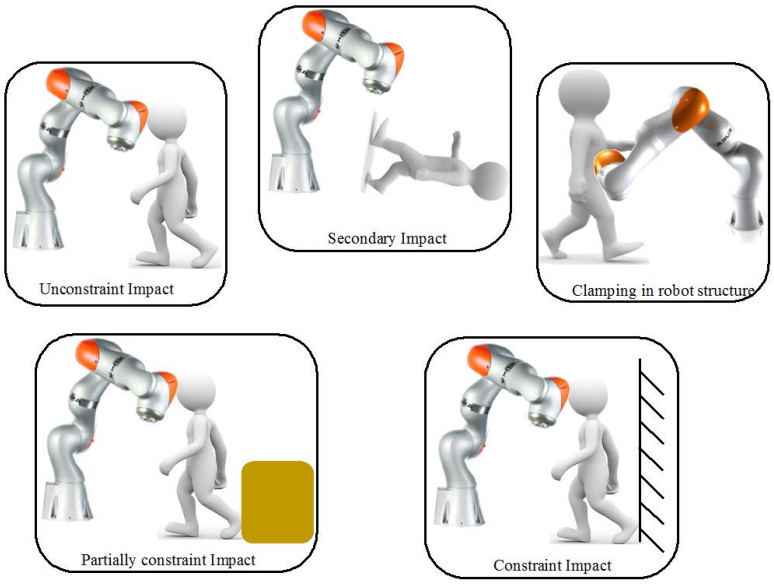


Figure 2.1. Classification of undesired contact scenarios during human-robot interaction [17].

Physical human-robot interaction and cooperation has become a topic of increasing importance and of major focus in the field of robotics. The primary requirement of the robot is it must in no case pose a threat to the human. However, over the years there have been many robot-related accidents, including fatal [16]. Management of risks for a human-robot shared working space involves in general very broad considerations. The five relevant physical contact scenarios that could potentially lead to human injury are summarized in [17]. They are differentiated to unconstrained impacts, clamping in the robot structure, constrained impacts, partially constrained impacts, and resulting

secondary impacts as depicted in Figure 2.1. During each of the collisions, various injury sources may be present, including fast blunt impacts, dynamic and quasi-static clamping, or cuts by sharp tools.

In this line of research, a vast amount of solutions for realizing safe collaborative tasks have been explored in recent decades. As mentioned above, industrial safety standards focus on ensuring safety by isolating the humans from the robots so that are not directly applicable to human-robot interaction applications. The coexistence of robotic systems and humans in the same physical domain poses the fundamental problem of ensuring the safety for the users and the robots. One common method for achieving safety during human-robot interaction is through low-level control of robot motion [18]. Control methods for improving the safety are divisible into two main classes: pre-collision and post-collision. In this context, “collisions” are not limited to blunt impacts, but can also include any harmful forms of contacts. Pre-collision approaches are implemented before a physical collision occurs, either by preventing the contact from happening in the first place or by bounding the robot’s velocity or energy. On the contrary, if unpredicted contacts occur, post-collision control methods should be able to quickly detect the collision and minimize the harm to the human operator as well as the robot.

Pre-collision control methods can also be referred as “prevention” methods, are techniques intended to ensure safety by monitoring the human, or the robot, or both so that the trajectory of the robot can be modified prior to incidence of collision. One common subset of the pre-collision approach is to set quantitative limitations to a robot so that it cannot pose any threat to a human even in the case of a physical collision. This can be achieved by effectively limiting the velocity of the robot: The relationship among impact mass, velocity, geometry, and medical observable soft-tissue injury is investigated by Haddadin et al. in [19]. Safety curves are developed for representing “safe” robot speed for a given configuration, mass, impact geometry, and impacted human body part. This insight is applied to design and control the velocity of the robot such that injury due to human-robot impacts is explicitly into account. Another approach is about planning the motion trajectory so that the safety constraints are satisfied [20]. The proposed planner limits jerk, acceleration and velocity in Cartesian space, and can be used online to establish visual and force control loop and to compute new trajectories in real time. Another energy-based control strategy is presented in [21]: Instead of planning the trajectories that require complex adjustments due to the changes in the position changes

in the environment, this approach bounds the dangerous behavior of the robot by tracking and limiting the total amount of energy stored within the system. Their proposed energy regulation control is implemented in a series elastic actuator prototype joint and is tested in both accidental collision and free motion.

Collision avoidance is an attractive broad area of research to solve the human-robot interaction safety issue, and several promising approaches have been proposed in the previous work. A projection-based safety system is proposed in [22] for ensuring hard safety in human-robot collaboration. The current joint positions and velocities of a robot are used to dynamically establish a safe space around the robot and display the safety zone on the surface via a projector. At any time, the robot is enclosed by a minimal safety space and enables the worker to utilize as much workspace as possible. A commonly used method in the field of collision avoidance is dynamic window approach which is originally proposed in [23], and then be modified as the field dynamic window in [24]. This strategy considers the obstacles near the trajectory by using the histogram grid representation of obstacles to eliminate the crashing probability. An effective mean of safeguarding against injury is to shut the robot system off depending on the safety distance between the robot and the human operator. In order to enhance the safety in human-robot interaction in industrial environments, the active control of robot based on sensors' measurements represents a viable method. Najmaei et al. introduced a dynamic risk assessment method [25] which considers all effective factors in a collision. A risk index includes the physical factors as well as the direction of eye gaze and human body orientation is proposed in this work. Two sensory systems are designed for measuring these values. The safety is finally guaranteed by using the obtained values to evaluate the danger and plan the robotic systems. A similar solution involves using an onboard distributed distance sensor is presented in [26]. The sensor's output is used as part of a newly conceived control strategy, aimed at improving the safety by means of assessing the level of danger induced by the robot.

Another popular approach to collision prevention via multi-level robot control structure is the calculation of danger criteria and fields. An early attempt is developed by Khatib [27], it defines a field of repulsive vectors that guide the robot's motion trajectory in response to dynamically changing environmental factors. Another real-time safety system capable of allowing safe human-robot interaction at low distances is proposed in [28]. This implementation turns a standard industrial robot into a human-safe platform without

requiring any robot hardware modifications such as special actuators or internal force sensors. By leveraging known robot joint angles and utilizing accurate human localization, a virtual representation of the workspace can be reconstructed to calculate the separation distances. The obtained distance information can be used to precisely control robot speed, allowing for safe HRI.

The potential field approach can also be deployed as a component of integrated human safety frameworks. One such framework proposed in [29] is based on a hierarchy of consistent behaviors. The safety coexistence is achieved with a layered approach for coping with undesired contacts. Based on the explicit measures of danger during the human-robot interaction, Kulie [30] developed another planning and control strategy. The level of danger they estimated were factors influencing the impact force such as the effective robot inertia, the relative velocity, and the distance. It is a methodology that integrates sensor-based information into the medium and short-term safety strategies.

Using a variety of strategies to prevent collisions can be an effective way to protect the human operators during the HRI. However, depending upon the complex factors including the type of the robot and the assigned task, strict collision avoidance is not always possible. As a result, some post-collision strategies are developed to detect a collision as it occurs and attempt to minimize the resulting damage.

Detecting and interpreting contacts is a crucial task in physical human-robot interaction. Golz et al. work towards a way to discriminate between intended and unintended contact types [31]. The way to achieve this goal is by deriving a set of linear and non-linear features based on physical contact model insights and from observing real impact data that relies on proprioceptive sensation only. A non-linear support vector machine (SVM) is used as the classifier that is capable to perform the discrimination online. Improvements to reduce injury in collisions can also be realized through proper mechanical design [32]. In their work, a boundary layer argument sliding mode controller is used for position control with mitigated chattering while securing reasonable positioning performance. The safety of both humans and robots during the collisions is improved by the increased joint compliance.

To address the safety and performance issues, a collision detection and control system based on the variable stiffness actuation device is developed in [33]. By using a

feedback linearization approach, the actuation capabilities of the device can be fully exploited under a mild condition on stiffness pre-loading. An end-user signal-based approach to collision detection and reaction is presented in [34] for an industrial manipulator having a closed control architecture without needing for additional sensors. The involved signals are those available to a generic end-user through the data interface provided by the manufacturer: joint velocity reference, joint position, and a signal related to the internal motor currents. Their method is implemented and tested on a KUKA robot using the robot sensor interface.

In addition to the studies on detecting collisions and designing reaction plans, some other researchers come up with quantitative assessment of the effectiveness of the post-collision strategies. One example is described in [35]. Two different collision detection methods and several reaction strategies are presented and validated experimentally. A mechanical verification platform has been built to assess and compare these methodologies. It has been proved that these strategies are able to reduce the contact lower than the level that might cause danger to humans. Another efficient safety improvement method that has been assessed with the DLR-III lightweight manipulator arm is reported in [36]. A complete treatment of the post-collision phase from collision detection and identification to robot reaction strategies is proposed in this study. More specifically, they take the directional information on contact forces provided by identification scheme and employ it to drive the robot away from the human operator.

Instead of simply moving the robot away or switching control methods to minimize the damage, some other works focus on making the robot be able to reason about the human's collaborative intent and support the operator better during the interaction. One framework in [37] developed a physically interactive control scheme for a manipulator robot arm. The indicator used to control the robot's behaviors is the different physical touch forces of the human. The intent of the human is estimated by observing the changes in control effort. After receiving the evaluated human intent, the robot is able to update its position reference accordingly. They also developed a switching scheme to make the robot go between the modes of pure impedance control with a fixed-position reference and the interactive control under human intent. When the human touches disappear, the robot goes back to the initial fixed position.

In summary, a variety of control-based attempts including pre-collision and post-collision methods for facilitating safety human-robot interaction have been carried out by researchers. Rach of the above-mentioned methods requires different sensors and has unique benefits and drawbacks. Trade-offs generally exist in these approaches: the more intelligent safety behaviors a robot can perform, the more complex the implementation is. Overall, the control-based methods have been proven experimentally to be effective for safe human-robot interaction.

2.2. Proximity sensors

As discussed in the previous section, the highlighted approaches that modify a robot's trajectories based on safety zones or separation distances have shown their superiority in practical applications. In these techniques, nonintrusive sensors for human localization and distance measurement are critical for real-world deployment.

Undoubtedly, one of the most necessary capabilities for the robots with the aim of having interactions with humans in their environment is docking. Video cameras are broadly used to provide details to the available information for decision making. Such a robot control strategy using the information provided by a video camera is reported in [38]. A pan video camera is designed together with a visual gaze algorithm that mimics the ability to live insects to control their direction of gaze and enable the fixation on a specific part of an environment. Computer vision processing techniques have to be involved in the proposed system in order to recognize an object of interest. Another collision detection approach proposed in [39] uses multiple cameras. They test collisions based on several stationary, calibrated video cameras, each supervising the entire three-dimensional space shared by unknown and known objects. They improved the basic image method in terms of classifying foreground pixels, exploiting epipolar line information, considering pixel sets in the collision detection, and automatically updating the reference image.

Though being widely applied, video cameras often suffer from a limited field of view. In a dynamic working environment, the obstacles including human operators can obstruct the cameras' view. If the cameras are placed at long distances, their performance will be limited due to the inevitable blind spots especially considering various approach angles between workers and the robots. Furthermore, real-time processing of video signals tends to be complicated and requires additional date from other sensor sources.

Another approach to achieve collision avoidance is using proximity sensors to measure the distances between robots and humans, and the measurement results are applied as the guidance for robot control.

Table 2.1. Commercially available proximity sensors

Type	Company	Model	Range(cm)	Price
Laser triangulation	Banner	Q4X [41]	2.5 – 30	$329
	Idec	MX1A/B-B [42]	5 – 13	$875
	SunX	HG-C1030 [43]	2.5 – 3.5	$345
Optical reflection intensity	SunX	FX-301 [44]	0 – 110	$135
	SunX	FX-411 [45]	0 – 150	$146
	Banner	DF-G3 [46]	0 – 300	$227
Infrared radiation	Sharp	GP2Y0A02YK0F [47]	15 – 150	$14.95
	Sharp	GP2Y0A21YK [48]	10 – 80	$13.95
	Grove	SEB39046P [49]	10 – 80	$13.9
Ultrosonic time of flight	Parallax	28015 [50]	2 – 3	$29.99
	Sick	UM30 [51]	0 – 800	$275
	Omron	E4E2 [52]	0 – 50	$802
Inductive	Sick	IME12 [53]	0 – 0.4	$18.9
	Panasonic	GX-8MU [54]	0 – 0.2	$93
	Schneider	IP69K [55]	0 – 1.5	$60.8
Capacitive	Eaton	E53 [56]	0 – 3.5	$232
	Autonics	CR18-8AC [57]	0 – 0.8	$83.9
	Sick	CQ35 [58]	0 – 2.5	$112

If accurate distance measurements can be made from each point on the surface of the human to the robot, unexpected contacts can be avoided depending on the obtained information. However, there are no perfect sensors and no way to provide a complete coverage, so the key factor becomes the balance among the detection range, the cost,

and the quality of the sensor data. The available proximity sensing technologies generally utilize one of the five physical principles of operation: intensity of reflection, triangulation, time of flight, capacitance, and inductance [40]. A list of commercially available proximity sensors [41]–[58] is provided and compared in Table 2.1. The chart is divided horizontally into the sensing technologies, and vertically by company name, model name, detection range in terms of centimeters, and the price. Some general ideas about different sensor categories can be observed from this chart: Sensors based on optical reflection intensity, infrared radiation, and ultrasonic time of flight tend to be able to achieve longer proximity detection ranges. In terms of measuring short distances with high accuracy, the sensors prices become relatively high.

Among those different categories of proximity sensors, the optical intensity of reflection sensors are one of the most widely available in the number of manufactures, the number of models, and the ranges of operation. The fundamental principle behind them is to measure the intensity of light reflected off an approaching object. The emitter and receiver optical fiber pairs are usually used to make optical sensors in a 2-D workspace. Design and construction of a 32-point optical fiber proximity sensor (OFPS) array mounted within a 36mm diameter robot finger has been presented in [59]. It provides a reliable detection for applications such as pre-contact velocity reduction and non-contact contour following. However, the small objects and dark or highly reflective objects will cause only small activation of the sensor and confuse it. In addition to the 2-D approach, a 3D optical proximity sensing panel for sensing a single object has been successfully built by Huang et al. in [60] and realized satisfactory accuracy by employing LED emitter arrays and photo detector (PD) arrays. This kind of sensors has several notable advantages including high resolution, high reliability, high sensor density, large detection range, and inherently robust design. Relative high cost and large size is another issue which makes it only acceptable for specialized applications such as an end-effector ranging device in the space shuttle manipulator system.

Different from optical sensors, laser diode proximity sensors employ the dependence of the laser output on the distance between the laser emitter and the target, so that the distance can be detected by monitoring the power modulation of the laser without any external optical components [61]. Therefore, the advantages of compact package and flexibility of integration with other systems make laser sensors a promising solution for distance and displacement measurement with millimeter-scale precision. Fang

et al. demonstrated a laser diode position sensor in [62] for a near-field height control system. It can approach the surface of a spinning disk within several wavelengths' range with a high accuracy up to nanometers. Their research offers an alternative method to achieve high-precision distance detection and can be applied in any field where nanoscale accuracy is required. Another low-cost multi-point laser distance sensor, based on a smartphone, that primarily working outdoors has been presented in [63]. The prototype of their proposed sensor combines a phone with an off-the-shelf line laser module so that it can achieve a 6 cm accuracy at 5 meters at outdoors environment. The major limitation of the utilization of laser is the safety to human's eyes. Many laser types, visible or invisible, are not safe to eyes and require not to stare into the beam. Moreover, dirt or other foreign debris on sensors will affect accuracy, so frequent cleaning is required. These eye safety and stain sensitivity issues are the reasons prevent laser proximity sensors from being wide-spread in industrial robotic systems.

Infrared radiation (IR) itself was unknown until early 1800's when Herschel's experiment with thermometer and prism was first reported. Detectors based on infrared radiation have been developed rapidly in the past few decades. The majority of IR sensors can be classified into two broad categories: photon sensors and thermal sensors. In photo sensors, radiation is absorbed within the material by interaction with electrons either bound to lattice atoms or to impurity atoms with free electrons [64]. A near-infrared proximity sensor array proposed in [65] can detect the position of a moving object at a distance of 10cm in real time. It is achieved by integrating a polymer light-emitting diode, a polymer photodetector, and an inorganic phosphor together. The second group of IR detector is thermal detector. They operate on the simple principle that, when heated by incoming IR radiation, their temperature increases and this temperature change is measured by temperature-dependent mechanisms such as resistance and pyroelectric voltage. By taking the advantage of modern MEMS technology, it becomes possible to achieve lower noise level through thermal isolation of sensors from on-chip heat sinks, obviating the need for cooling of sensor chips [66]. However, both the two types have shortcomings: Photon sensors are limited by generation-recombination noise arose from photon exchanges with radiation background. Thermal detectors are fundamentally limited by temperature fluctuation noise arising from radiant power exchange with a radiating background [67].

An alternative technique for proximity sensing is using ultrasonic transmitters and receivers. Ultrasonic sensors radiate a short ultrasonic pulse generally in the 20 kHz to 500 kHz, and this pulse bounces off a local object and the echo is detected, often by the same transducer. The time elapsed between the transmission and detection of the waves is used to measure the path covered by ultrasonic waves, and consequently, the reflecting surface distance. The maximum detectable distance depends on the power of the ultrasonic waves, the acoustic impedance of the propagation medium, the sensitivity of the receiver, and the reflectivity of the target. Moreover, the characteristics, which are to be tailored according to the specific application, of the measuring system have an effect on both maximum and minimum detectable target distances [68]. A 40 kHz nominal frequency ultrasonic sensor was developed for the monitoring system and respiration signals are measured using ultrasonic attenuation characteristics in [69]. The proposed sensor avoided the need to apply electrodes and to wire the subject to the monitor, and the measurement range can be extended to around 100cm. However, this system could not detect enough body motion information for monitoring of a subject with thick cloth or covers. The complementary features of reflecting behavior for ultrasonic waves and infrared radiations in the air makes it possible to fuse together the information they supply. Determining the properties of the surface of an object with an ultrasonic sensor and subsequently calculating the distance by using an infrared sensor is described in [70]. They experimentally verified that reliable distance measurements can be achieved by combining the two types of sensors.

Another family of proximity sensors is inductive proximity sensors that are useful for precision measurement and inspection applications due to their ability to detect at close range. Inductive proximity sensors produce an oscillating radio frequency (RF) field at the sensor face. When metallic objects are brought into this field, the oscillating field is interrupted and the current in the sensing coil is altered. This change is detected by the detector circuit. Different types and sizes of detectors have their specific sensing range switch point so that metal target displacement and position detection is accurate and repeatable. A highly sensitive inductive proximity sensor microsystem based on a flat coil and a new electronic interface circuit is proposed in [71]. The coil is flat and realized with a low-cost technology. The technology, involves a negative laminated dry-film resist mold, is used for electroplated parts. Fabrication processes are totally CMOS compatible. The obvious advantages of this design are its simplicity, versatility, high sensitivity, a sub-

micrometric threshold even with non-ideal coils, and no need for exciting alternating current supply. In [72], the micro sensor was designed as multi layers of micro planar spiral coils which were connected through a single point contact. To get a higher signal to noise ratio, the absolute value of the inductance needs to be larger, because a larger inductance variation induced by eddy current can be expected. Spiral coil configuration has been selected in this work due to its larger inductance density compared to planar coil structures. The strengths of inductive sensors are immune to adverse environmental conditions, high switching rate, and long operational life with virtually unlimited operating cycles. However, the weaknesses are also obvious: limited sensing range, detect only metallic objects, and may be affected by metal chips accumulating on sensor face.

Quite similar to inductive detectors, magnetic sensors use a DC magnetic field generated by a permanent magnet instead of the AC field. Hall Effect, discovered by E. F. Hall in 1879, happens in a conductive sheet. With a linear current flowing in one axis, a linear field-dependent voltage in the other axis can be measured as the magnetic field is induced through the sheet. A drawback of commercially available Hall Effect device is the large and poorly controlled offset voltage which may be compensated by the use of AC coupling or signal calibration strategies.

Among the different types of available proximity sensors, the small size, low cost, high sensitivity, and self-test capabilities make the capacitive-based detectors appealing for both academic research and industrial applications. Capacitive proximity sensors detect the target objects by measuring either the change in dielectric constant of the sensing volume or electric field shielding due to the presence of a human, metallic, or non-metallic objects. The aim of this work is developing a capacitive-based sensing system for the human-robot interaction in the manufacturing environment, so a review on the previous works on capacitive proximity sensors is provided with a separate sector as follows.

2.3. Capacitive proximity sensing

The capacitive-based proximity sensor is one of the most commercial and viable solutions for improving the safety in human-robot interaction applications. It is capable to detect the presence of most obstacles with large coverage and accurately measure small gaps with fast response [73]. Capacitive proximity sensors can be divided into two categories based on whether or not the obstacle forms one plate of the sensor. One uses

the principle of a parallel-plate capacitor; this configuration consists of a single plate on the sensor itself and uses the obstacle to be measured as the second capacitor plate as shown in Figure 2.2 (a). Despite its simplicity, this parallel-plate type exhibits three major drawbacks: the object being measured has to be a conductor; the inverse distance-capacitance relationship is highly nonlinear as depicted in Figure 2.2 (b); it is impossible to use this kind of sensors in applications when one part is inaccessible [74]. The other type of structure that based on fringe effects as shown in Figure 2.2 (c) has also been constantly studied. The first capacitive sensor of the second type was proposed by Noltingk in [75] at 1969. Two sensing structures were constructed, with one having two rectangular electrodes and one grounded metal screen in between, and the other having a circularly symmetrical system which made the sensor more robust. The achieved approximate linear ranges were provided: up to 24mm with the rectangular structure and 6mm with the circular structure. However, there was no analytical formula for the optimization process at that time.

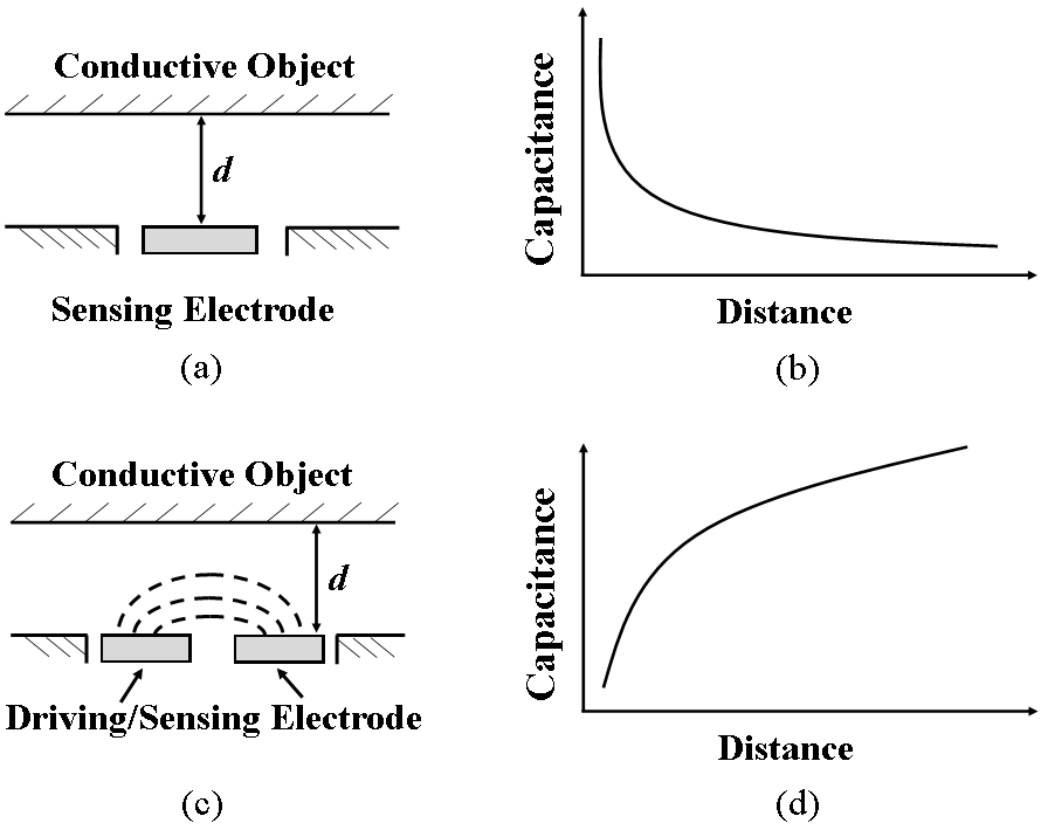


Figure 2.2. Two types of the capacitive-based proximity sensors. (a) Parallel-plate capacitor. (b) Typical response of the parallel-plate type of sensor. (c) Fringing effect capacitor. (d) Typical response with fringing effect sensor.

The capacitive proximity sensor used for safety applications should have long detection range and fast response in order to secure enough response time before colliding with ambient objects. The electrode field is used in [76] to provide a sense of “pre-touch”, with the range intermediate between vision and contact-based touch sensing. Then, a series of increasingly complex manipulators incorporating the capacitive sensor is presented in this work: The initial experiment is about simply avoiding a dynamic object. Next, a 1-D alignment task for grasping is described. Finally, the sensor is built into the fingers and the wrist of a manipulator so that these parts can be rotated. Their work is mainly about an effective pre-touch-based grasping. A similar work is presented in [77], a grasping system that is guided at short range by a capacitive proximity sensor. Multiple sensors are involved so that two different tasks can be achieved: The capability of performing human-to-robot and robot-to-human handoffs, including the use of the capacitive sensor to detect whether or not the human is also touching the object that the robot is holding. The ability to pick up standalone objects.

Capacitive sensing approaches are widely applied for indoor localization, in which embedding large loading mode electrodes underneath floor tiles to determine where the user is standing is commonly required. A flexible and integrated capacitive sensing system that allows the detection of the presence of a human over distance is developed in [78] for indoor localization as well as fall detection. Their proposed system is based on sensing mats that can be placed under various types of floor covering. The mats can wirelessly transmit data to a central platform. However, this method is not viable for large empty spaces due to a limited maximum mat size and the fact that the sensor has to be placed at the border. One further step is taken by Sousa et al. [79], in which a method for user tracking and localization based on textile capacitive sensor arrays placed under the floor is introduced. A grid of capacitive sensor plates are placed under the floor for detecting and a small user-bound device equipped with an accelerometer is used for user identification.

Alternatively, some passive capacitive sensing systems are designed for the application of estimating the human’s location. Platypus, proposed in [80], is the first system to localize and identify people remotely and passively. The sensing principle based on the body electric potential is changing naturally during walking and when interacting with the environment. The working procedure of the proposed system includes observing the ambient electric field to localize a person, reconstructing the body electric potential

change, and extracting a signature for identification. Dynamic time warping (DTW) is applied for classification, with the localization of events as time series. One limitation located in this work is that personal-specific model training is required and identification of user depends on wearing the same footwear. Another passive measurement of static electric field is presented earlier in [81] to infer the amount and type of body motion, gesture, and activity. The name of their approach is “Mirage”, and it uses non-body contact technique leveraging human-generated body charge. Human body introduces electric field distortions when performing motions and the electric field is measured through an electrode placed at a distance from the subject for extracting desired information.

Capacitive sensors can also be used to infer a human’s pose often operate on the similar principles as the above-mentioned indoor localization systems. The active transmitting mode sensing is used in [82] for an unobtrusive recognition of human’s height and posture. The measurement method is based on the capacitive coupling of low-frequency signals and a setup that is used to measure the capacitance between the human body and a hidden floor electrode. The system measures the distance from the floor to the tallest point of the body which generally is at the crown of the head to obtain the height information in pervasive and context-aware applications. On the contrary, the loading mode sensing technique is applied in [83] to achieve similar goals. The application of their proposed capacitive proximity sensor is using the whole expressivity of human body motion for interacting with intelligent environments.

Wearable systems usually offer a more specialized way of recognizing postures or motions. Conductive textile patches can be used to measure the capacitance of the human body and can be used to reveal information about a broad range of activities. A flexible, textile capacitive sensor proposed in [84] can provide information that is complementary to motion sensors. Instead of measuring the electric field generated by human beings, active capacitive sensing is used in this work to generate the electric field and to measure the capacitance of the human body underneath the electrodes. Their designed sensor can be integrated into soft wristband, neckband, and socket to provide detailed information on complex user activities. A flexible capacitance type sensor that can detect an approaching human without contact is designed in [85] for the applications of maintaining and promoting health. Thin film can support impact-resistant and high from factor devices, it can be attached to a wide range of objects including wall, floor, carpet, bed, etc., such that they are unnoticeable. In their proposed work, the sensing electrodes are arranged at the

top and bottom surfaces of a polyethylene terephthalate film with simple duplex printing technique.

Capacitive proximity sensing can be extended beyond surface interactions. It has been explored for 3D gesture interaction since the mid-1990s [86]. Most gesture recognition systems involve the usage of the shunt sensing mode which is able to provide more geometrical information per electrode than using other sensing modes. Early attempts include the compact Field Mice [87] and Lazy Fish [88] platform. The former approach measures the value of a component in an effective circuit diagram that summarizes all possible current pathways involving the body to be sensed and the sensing electrodes. An analytical model of the sensor response and a probabilistic framework are presented for inferring the geometrical information. The latter attempt of inferring 3D geometrical information is achieved by multiplexing the transmitters and receivers so that multiple “projections” can be measured. This platform was then evolved into the modular School-of-Fish [89] platform.

More recent work investigated generic proximity sensing systems that sense 3D information to control other devices including computers, vehicles, automation systems, etc. One example is a multi-purpose capacitive proximity sensor input device [90] that allows modeling of devices based on advanced sensor units that involves data processing. A hand-tracking prototype based on an array of capacitive proximity sensors is modeled, and a method to integrate the complex pre-processing procedures is developed in this work. Towards interactive car interiors, the active armrest is developed in [91] in which a regular car armrest is equipped with a capacitive proximity sensor that combines limb detection and gesture recognition together. The sensor is designed for invisible integration into existing environments and can be used to create interactive surfaces in a car. In terms of interaction systems that are not directly apparent, a capacitive gesture recognition system “Rainbowfish” is developed in [92]. It combines a semi-transparent capacitive sensing surface with an LED array which is used to visually indicate possible gestural movements and provide feedback on the current interaction status.

After a comprehensive review of the previous and ongoing research on capacitive proximity (electric field) sensing technology, it is notable that capacitive-based sensors have been widely used in a broad range of applications. Capacitive sensing has some unique features that provide benefits in particular applications: The generated electric field

propagates through any non-conductive material; the electrode plates can be integrated to various object unobtrusively; the energy consumption is small; they have a high versatility that is achieved by modifying the electrode's geometry and electrical parameters.

However, the design challenges for this type of sensors are also obvious. One of the largest technical obstacles to its development is the response of a capacitive sensor is a nonlinear function of the input such that extracting useful information from measurements is a difficult computational problem. Moreover, noise that adds random unwanted parts to the capacitive responses is a property of any capacitive sensing system. Environmental factors and drifts that subject to changes over time have a strong influence on the signals. Lastly, the inherent ambiguity of sensor readings is a prevalent problem to infer high-level information from raw measured data. More specifically, in terms of distance estimation, a small object that is close to the sensor most likely results in the same reading as an object with larger size but at an increased distance. In our design, we will take advantages of the above-mentioned merits of capacitive sensors, consider these issues thoroughly, and finally provide our solutions.

2.4. Continuous tracking and classification methodologies

Capacitive sensing inherently involves inferring information from measured data, either continuous (e.g. for estimating the human's position), or discrete (e.g. for gesture recognition).

Models that determine continuous properties are often physically motivated. Many models are based on pseudo-probability distributions that estimate the most probable system state. One such technique for interpretation of an array of capacitive sensors is proposed in [93] so that the information can be used for direct foreground interaction control and background control or context sensing. Bayesian approach is adopted in their work to explicitly model uncertainty. In addition, as the distributions evolve over time, Sequential Monte-Carlo techniques (particle filters, PFs) are involved. The usage of these techniques allows to cope with dynamic and noisy sensor inputs and to create systems whose level of autonomy increases as input ambiguity increases. This work has been extended by the same group of researchers [94] to be applied in real-time capacitive touch tracking. Again, a probabilistic model is built to estimate the pose effectively and give

appropriate uncertainty evaluations. Another approach is based on the Swiss-Cheese-Algorithm which detects objects using elimination methods. An object recognition method extended from Swiss-Cheese is proposed in [95] for ubiquitous interfaces. The most likely configuration of body parts is determined based on the readings of many distributed capacitive proximity sensors. Moreover, a list of probabilities for object presence in each point in the space is obtained by the developed algorithm. The method is able to track existing and newly appearing objects in real time.

An alternative approach to deal with the continuously 3D localization problems is explained in [96]. Their contribution is a thin and transparent design for capacitive sensing, allowing for 3D finger and hand tracking as well as in-air gestures on a mobile device. The regression method involved in their work is a random decision forest (RDF), which works as a black box. It is a machine learning method to approximate the complicated mapping problems in this work and yield fast and reliable results.

Generally, the applications of capacitive proximity sensors involve more than one detected state, so classifiers are frequently used. One commonly used classifier is decision tree that can overcome the problem of conditional dependence of features. In practice, binary decision trees with threshold decision functions for a single feature are broadly used. One application is presented in [97], a mobile service with a smartwatch-style system to promote high-fives in everyday workspace interactions. Decision tree is used to extract features and identify unique peaks from high-five. This methodology is also involved in home automation and environmental control applications through flexible textile-based capacitive sensor arrays [98]. Their proposed system is named as “Inviz”, a gesture recognition system to detect movements of paralysis patients. The signal processing procedure for gesture recognition combines feature extraction algorithms and machine learning techniques in a hierarchy of processing. Machine learning techniques such as nearest-neighbor, Bayesian inferences, and decision trees are in the highest tier of the signal processing hierarchy. The average accuracies of gesture recognition for these three algorithms are compared: the accuracies are comparable while the nearest neighbor classifier performs slightly better than the other two. Multiple learning algorithms are tried in determining contextual information from furniture using capacitive proximity sensors presented in [99]. In their work, Naïve Bayes classifier, decision tree, a Radial basis function (RBF) networks are used and their performance in terms of training and

testing precision is compared. Their conclusion is RBF network is a robust classifier that performs the best on the test data set for user posture classification tasks.

Support Vector Machine (SVM) is an alternative category of classification methodologies that frequently applied in extracting information from sensing data. Laput et al. use SVM for touch recognition of un-instrumented, electrical and electromechanical objects [100]. Most everyday electrical and electromechanical objects emit a small amount of electromagnetic (EM) noise during their regular operation. This unique body-coupled EM signatures can be used to identify different objects and infer object states with SVM classifier. Across the 24 classes, SVM achieves an overall accuracy of 97.9% with the majority of the objects (18 out of 24) reach the accuracy of 100%. SVM is also used in capacitive fingerprinting [101], to explore user differentiation. The electrical properties of humans and their attire can be used to support user differentiation, the ability to tell users apart but not uniquely identify them, on touchscreens. For single finger touches, the SVM yields an all-pair average accuracy of 97.3%; while in distinguishing between users performing a variety of gestures, the achieved average accuracy is 97.8%.

Multiple classification tools have been tried in [102] for a graspable user interface which is an intuitive way to manipulate 3D objects in a virtual environment. A spherical shaped device “Grip-Ball” that based upon capacitive multi-touch sensing is proposed in this paper. It is able to recognize five hand grip patterns so that intuitive accessing and manipulation of the virtual object can be performed. A minimum distance classifier (MDC), a naïve Bayes classifier (NBC), and a support vector machine (SVM) are involved for the grip-pattern recognition. Their achieved test accuracy are 98.2%, 98.8%, and 99.4% respectively. A similar hand grip pattern recognition interface is designed for handheld mobile devices [103]. It is based on the sensed signal from an array of capacitive touch sensors. The pattern recognition algorithms being used include naïve Bayes (NB) support vector machine (SVM). The authors verified that the SVM achieves higher cross-validation accuracies under different parameters’ values (higher than 90%) compared to NB classifier (76%). Moreover, typical confusion matrices of NB and SVM are also provided: SVM still performs much better than NB classifier in terms of the number of misclassified samples.

In response to the demand of promoting the human-robot interaction safety in the workspace, an innovative technology is proposed to reduce the risks of injury for next-

generation manufacturing. More specifically, the challenge is addressed by subdividing the problem in three distinct but complementary scenarios as described in Figure 2.3.

In the first scenario, as shown in Figure 2.3 (a) when the distance between the robot and the worker is long (0.25m - 2m), the directional infrared sensor is involved to detect the presence of the human. Once the operator is detected to be within this range, the system starts to constantly monitor his/her movements, and detect the distance and direction of the worker from robotic components. This is particularly important when a machine is moving at high speed since sufficient deceleration time will be required, especially if the payload is supported with cables.

When the worker and the robot are within a short distance from each other (shorter than 30cm), the scenario 2 (Figure 2.3 (b)), a capacitive sensor needs to be worn by the worker to detect the objects in his/her proximity. The specific requirement for this sensor is reliably estimates the distance between the object and the human. However, there is a technological challenge: estimate the distance accurately and eliminate the ambiguities brought by different objects' shapes. This problem, to the best of our knowledge, has no reliable and un-expensive commercially available solution.

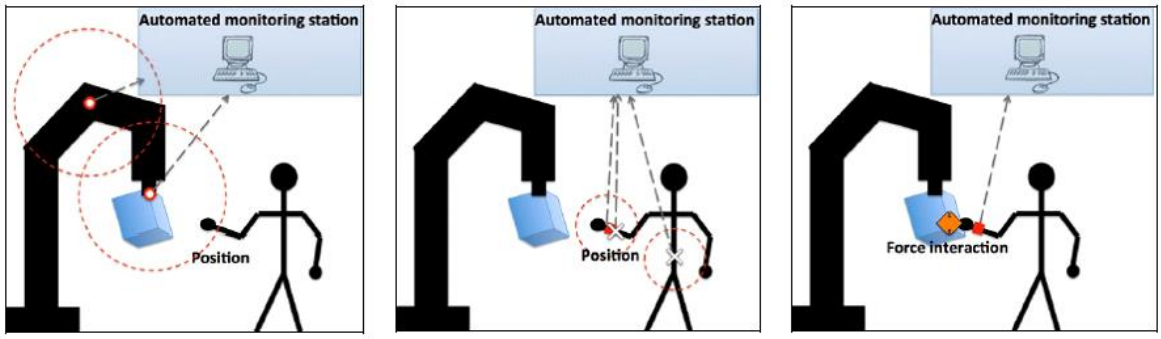


Figure 2.3. Three sub-scenarios of the human-robot interaction and corresponding safety guard methods. (a) Directional infrared sensor to monitor the movements of the human when robot and human are relatively far from each other. (b) Capacitive proximity sensor accurately estimate the distance when they come closer. (c) The force is monitored by Smart Skin when human is in contact with the payload. This figure was created by Dr. Carlo Menon.

In the last scenario that is described in Figure 2.3 (c), the human operator is in contact with the payload held by the robotic manipulator. In this case, the information used

to guarantee the safety is the force he/she exerting on the payload. The Smart Skin is used to monitor this force, and a force value approaching the threshold limit indicates a potentially hazardous situation.

The focus of this work is the development of a high-performance capacitive proximity sensor that is used in the second scenario. It enables the detection of distance as well as classify the objects into distinct categories.

Chapter 3.

Structure modeling

The design of a multiple-electrode fringing electric field sensor relies on a good understanding of the fundamental principles and trade-offs. The purpose of this chapter is to highlight the critical aspects of the capacitive sensing theory and to illustrate the theory with numerical simulations. For different functionalities, the major goals of sensor design are different. For basic proximity detection, the longest linear detection range is with the highest design priority. Whereas for more complex applications, the optimum balance of sensitivity, resolution, and signal strength should be achieved.

3.1. Capacitive sensing theory

The general term Electric Field Sensing is used to refer to a family of non-contact measurements that be made with slowly varying electric fields [104]. Several of these measurements had been lumped together under the rubric of “capacitive sensing”, in which low-frequency voltage signal is applied to the transmit electrode, and displacement current flows from the transmitter to the other conductors (receivers) through the effective capacitors.

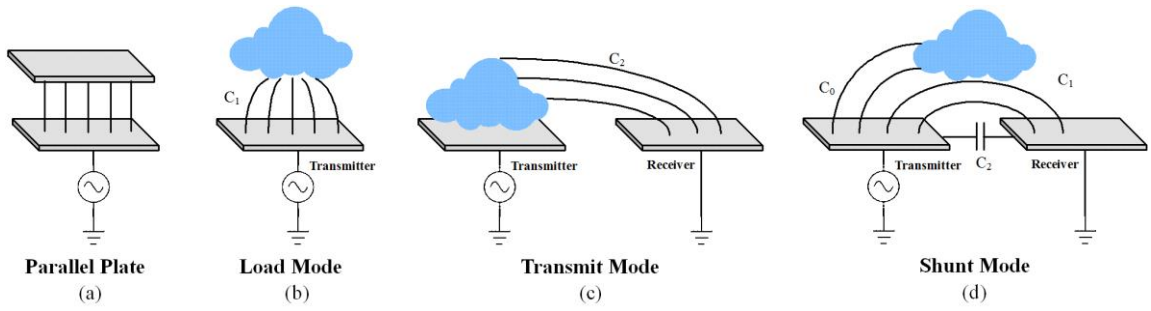


Figure 3.1. Four different capacitive proximity detection modes.

Conventionally, a capacitor is in the form of a parallel-plate, where the transmitter and receiver are placed opposite to each other in close proximity so that the electric field is distributed between them uniformly as shown in Figure 3.1 (a). For proximity sensors, the capacitance is dependent on the common plate area *A* over average plate spacing *d* assuming no fringing effects:

$$C = \varepsilon_0 \varepsilon_r \frac{A}{d} \qquad (3.1)$$

where ε_0 is the permittivity of the vacuum, and ε_r is the relative permittivity of the material in between the two electrodes. However, its gap-capacitance relationship is highly nonlinear and the sensitivity drops significantly in the cases of larger spacing. These are the two major constraints prevent this type of capacitive sensors from being used in broader applications [105].

When the electrodes gradually open up, the electric field is no longer confined within the small region between the electrodes, but expands into a wider space and forms a fringe field. This structure provides a possibility to interrogate an object from only one side, which is particularly useful in the scenario of proximity sensing. Corresponding to the different current pathways, the measurements can be further divided into three modes: transmitter loading mode, transmit mode, and shunt mode [106] as depicted in Figure 3.1 (b), (c), and (d).

Transmitter loading mode is the original electric field sensing pathway, in which the current flowing from the transmitter is measured. The value of C_1 in Figure 3.1(b), and thus the load on the transmitter, changes with the position of the object: when the object approaches the single transmitting electrode, the loading current increases. Despite its wide applications, the highly nonlinear capacitive responds and the strict restrictions on the material of the detectable object prevent it from becoming a promising candidate for this work.

In transmit mode, the transmitter is coupled strongly to the conductive object, either by direct electrical connection or capacitive coupling through insulator [104], so the object is essentially at the potential of the transmitter. The electric field properties would, therefore, depend on the spacing from object to receiver: as the object approaches the receiver, the value of C_2 in Figure 3.1 (c) increases, and the displacement current at receive electrode increases. This mode is most ubiquitously used in touch or floating touch sensing, not suitable for proximity sensing.

Shunt mode is a three-terminal measurement: neither the transmitter nor the receiver is in contact with the object. In this regime, C_0, C_1, and C_2 in Figure 3.1 (d) are of the same order of magnitude. When an object enters the electric field, C_1 increases and C_0 decreases, leading to a drop in received current: displacement current that had been

flowing to the receiver is shunted by the object to ground (hence the term shunt mode) [107]. There is a strong sense in which the shunt mode can offer more informative measurements: it is possible to create numerous virtual sensors while having a manageable number of electrodes, based on different transmitter-receiver configurations. With *N* ordinary capacitive sensors (loading mode), *N* capacitive values can be collected. These *N* numbers turn out to be the diagonal of the capacitance matrix for the system of electrodes. Whereas in shunt mode, one can get *N(N-1)* off-diagonal measurements that result in *1/2N(N-1)* independent capacitive values due to the symmetrical capacitance matrix. Moreover, shunt mode measurements can also be used in combination with multiplexing methods allowing parallel accesses to multiple transmitters at the same time. With all these advantages, shunt mode within an electrode matrix is adopted as the basic architecture of this proposed capacitive proximity sensor.

Real-world electric field sensing involves moving charges and AC voltage sources. For an accurate physical analysis, Maxwell's equations relating electric and magnetic fields, charge density, and current density should be used. However, a simplified approximation that ignores magnetic fields always is possible without significant loss of accuracy [105]. The capacitive sensor applications are almost all small and work at a low frequency so that the time constant is much shorter than the circuit response time, and thus, the simplified versions of Maxwell's equations can be applied. Systems in which this approximation is reasonable are defined as electroquasistatic. A given distribution of charge density ρ produces the electric field intensity E, and the magnetic field intensity H is approximated by zero as followings:

$$\nabla \times E = -\frac{\partial}{\partial t}\mu_0 H \approx 0 \qquad (3.\ 2)$$

$$\nabla \times H = \frac{\partial}{\partial t}\varepsilon_0 E + J \approx 0 \qquad (3.\ 3)$$

$$\nabla \cdot \varepsilon_0 E = \rho \qquad (3.\ 4)$$

$$\nabla \cdot \mu_0 H = 0 \qquad (3.\ 5)$$

In which the magnetic permeability of vacuum, μ_0, is a fundamental physical constant, defined as $4\pi \times 10^{-7}\ N/A^2$.

There are three types of solutions to the zeroth and first order Maxwell equations, which correspond to the three basic types of circuit components: capacitive, inductive, and resistive. For electric field proximity sensing, only the capacitive solutions are relevant.

The circuit definition of the capacitance is defined in Equation (3.6), in which a capacitance couples a current to the time derivative of a voltage. Associated with the zeroth-order electric field is a zeroth-order charge, and since the zeroth-order electric field can be represented by a scalar potential, this first-order current is coupled to the time derivative of the charge as shown also in (3.6). The static (zeroth-order) electric fields satisfy Laplace's equation, and the charge on the conductor *i* due to the electric field is written as (3.7), where S_i is the surface completely enclosing the conductor, *E* is the electric field vector normal to S_i.

The capacitance of conductor *i* due to another conductor *j* is the ratio between the induced charge on *i* and the voltage between *j* and a reference. Because of the linearity of all the equations involved, the total charge *Qi* is the sum of the separately induced charges as in (3.8). The off-diagonal term of the capacitance matrix *Cij* represents the mutual capacitance, and the diagonal self-capacitance term *Cii* represents the "loading" of the transmit electrode by the approaching object. The capacitance matrix is symmetrical that means *Cij* = *Cji*.

$$I = C\frac{dV}{dt} = \frac{dQ}{dt} \qquad (3.\ 6)$$

$$Q_i = -\oint \varepsilon E \cdot dS \qquad (3.\ 7)$$

$$Q_i = \sum_j C_{ij} V_j \qquad (3.\ 8)$$

At this point, it can be observed that by measuring the current received at the electrodes, the mutual capacitance can be derived by specifying V_j. However, there is a lack of insight into the operation of the sensor without knowing the theoretical solution for the mutual capacitance as a function of the device geometry and the distance between the sensor and the approaching object.

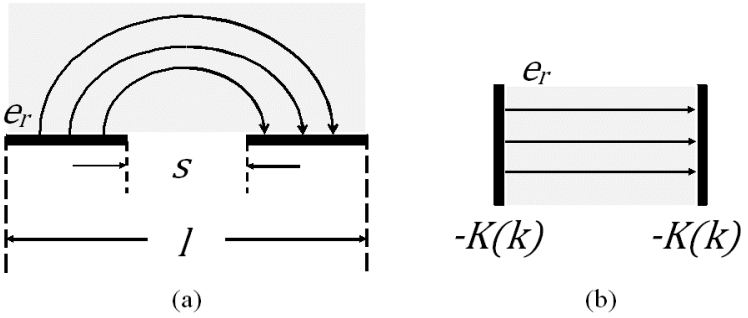


Figure 3.2. Conformal mapping technique. (a) Coplanar structure. (b) Equivalent parallel structure.

To calculate the mutual capacitance, a conformal mapping technique can provide a good approximation [108]. It is based on Christoffel-Schwarz transformation, which makes it possible to transform the coplanar structure into the rectangular parallel structure as shown in Figure 3.2(a). In the analysis presented in [109], the electrode plates are assumed to have zero thickness and infinite conductivity; the plate spacing *l* is larger than the separation *s* so that the end effect can be avoided. The calculated mutual capacitance is given as:

$$C = \frac{\varepsilon_0 \varepsilon_r}{2} \cdot F(k);\ \ F(k) = \frac{K(k')}{K(k)};\ \ k = \frac{s}{l};\ \ k' = \sqrt{1-k^2} \qquad (3.\ 9)$$

where *K(k)* is the complete elliptic integral of the first kind, *k* is the modulus of the elliptic integral. *F(k)* can be expressed by the approximate formula (3.10), and the effects of different spacing and separation to *F(k)* are plotted in Figure 3.2(b). It can be found that the mutual capacitance is effectively determined by the sensor geometrical parameters *s* and *l*: reducing s or increasing l can result in larger capacitance value which means higher sensitivity, but a shorter linear range of measurement. It is also proven in [110] that separation *s* is the most dominant and influential parameter.

$$F(k) = \frac{K(k')}{K(k)} = \begin{cases} \pi^{-1} \operatorname{Ln}\left[2\frac{1+(1-k^2)^{0.25}}{1-(1-k^2)^{0.25}}\right] & for\ k^2 \le 0.5 \\ \pi\left[\operatorname{Ln}(2\frac{1+k^{0.5}}{1-k^{0.5}})\right]^{-1} & for\ k^2 \ge 0.5 \end{cases} \qquad (3.\ 10)$$

In practical applications, electrodes are fabricated on the supporting substrate and the nearby object interferences the generated electric field. The cross-section view of a two-dimensional model for one electrode pair as shown in Figure 3.3(a): the two infinitely long flat conductors (A, B) are separated by a gap *s* and each has a width of *w*. The substrate thickness is *p* and its relative permittivity is ε_r, and *d* represents the distance between electrodes and the object. There exist two modes in which the two electrodes may be electrically excited: even mode and odd mode [111]. Even mode operation is achieved by driving the two conductors with the same time varying potential while the odd mode results from using a 180° phase shift between the excitation signals. For conductors with thickness much less than the width and relatively small substrate thickness (*w>0.35p*), the mutual capacitance per unit length can be expressed as the difference between the total even and odd mode capacitances from conductors to the object [112] as described in equation (3.11). The total even and odd mode capacitances are evaluated by Cohn in [113]:

$$C_{mutual}(s,d) = \frac{1}{2}(C_{odd} - C_{even}) \qquad (3.\ 11)$$

$$C_{odd}(s,d) = \left[\frac{s}{d} - \frac{2}{\pi}\ln\left[sinh\left(\frac{\pi s}{2d}\right)\right]\right]\varepsilon_0\varepsilon_r \qquad (3.\ 12)$$

$$C_{even}(s,d) = \left[\frac{s}{d} - \frac{2}{\pi}\ln\left[cosh\left(\frac{\pi s}{2d}\right)\right]\right]\varepsilon_0\varepsilon_r \qquad (3.\ 13)$$

By utilizing superposition of electric fields, the total mutual capacitance caused by the substrate and the nearby object can be expressed as the sum of both components:

$$C_{mutual}(s,p,d) = C_{low}(s,p) + C_{up}(s,d) \qquad (3.\ 14)$$

$$C_{low}(s,p) = \frac{1}{2}C_{mutual}(s,2p) \qquad (3.\ 15)$$

$$C_{up}(s,d) = \frac{1}{2}C_{mutual}(s,2d) \qquad (3.\ 16)$$

This two-dimensional model can be used to investigate the relationships between the device geometry and its performance. In order to maximize the change in mutual capacitance due to the change in the distance to the object *d*, it is desired to minimize the substrate thickness *p* so that as much of the electric field on the back side can be shielded out as possible. It is predicted that the mutual capacitance responds monotonically to the changes in distance as shown in Figure 3.3(b) where the mutual capacitance as a function of *d/s* is plotted with a solid line. The device sensitivity, also known as the rate of capacitance change, vs. d/s is plotted in Figure 3.3(b) with the blue line, from which it can be observed that the greatest sensitivity happens when the distance is equal to the separation. Increasing the sensitivity by increasing separation will result in a reduction in the magnitude of the mutual capacitance as well as the dynamic range. So the device geometrical parameters should be determined to achieve best tradeoffs among various performance indexes.

All the theoretical analyses are based on a large number of assumptions and strict boundary conditions, so they only point out the direction for improving the design of the sensing patterns. The responses on capacitance due to the approaching object may be affected a variety of factors that cannot be precisely expressed by any formula.

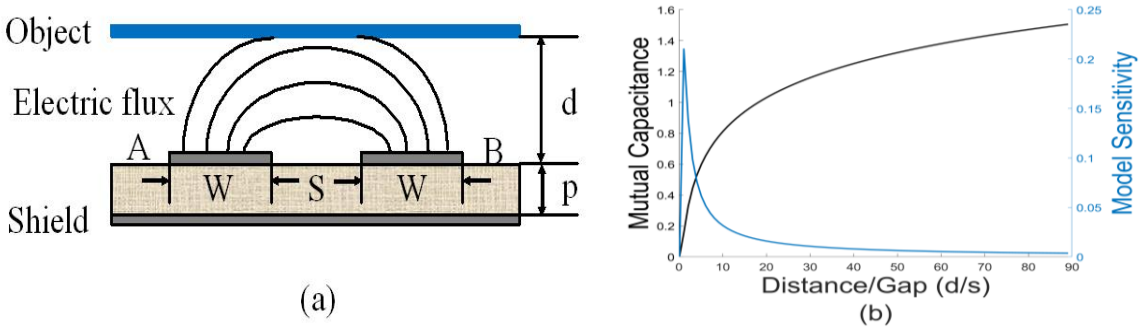


Figure 3.3. Two dimensional model for one electrode pair. (a) Cross section view of the model. (b) Mutual capacitance and model sensitivity vs. distance/gap (d/s).

3.2. Structure modeling

As mentioned in the last part, the analytical models are usually built based on some idealized assumptions [114]–[117] due to the inherently nonlinear characteristics of fringing electric field sensors. That limits their accuracy for real-world applications. The capacitive sensor design therefore relies highly on the numerical simulations. Among all the simulation methods, finite element methods (FEM) are extensively used for capacitive sensor modeling, optimization, and evaluation [118]–[120], especially when the precise theoretical model is hard to achieve. COMSOL Multiphysics® is an interactive environment that uses FEM to solve scientific and engineering problems [121]. Several modules are available for COMSOL, categorized according to the application areas, namely Electrical, Mechanical, Fluid, Chemical Multipurpose and Interfacing. In this study, electromagnetic fields in low-frequency applications are mainly involved, so the MEMS module belonging to the Electrical module is employed. The accuracy of the FEM simulation results depends on the model definition as well as mesh generation. When the model that properly reconstruct the real applications, COMSOL can generate results with high quality.

Before diving into designing and modeling the sensor structure, it behooves us some time to clarify some figures of merit that is used to evaluate the performance. These figures of merit include penetration depth, measurement sensitivity, signal strength, and noise tolerance.

Penetration depth is one of the most important parameters used to estimate the performance of a proximity sensor: it indicates how quickly the electrical field decays as

the distance increases. Though there is no explicit definition, one way to evaluate effective penetration depth is to measure the distance at which the capacitance difference to asymptotic (sample infinitely far from the sensor) value equals to 3% of the difference between the highest and the lowest values [122]. Penetration depth is roughly proportional to the spatial wavelength λ, which is the distance between the centerlines of neighboring electrode groups of the same type (e.g. transmitter or receiver).

Measurement sensitivity is defined as the ratio between the change in sensor output and the change in the measured physical parameter (in this case is the distance) of the sample [123]. Because the electric field of the sensor is non-uniform, the measurement sensitivity is position dependent: sensitivity decreases exponentially with the increase of the distance. Sensitivity also depends on the area of the electrodes: for a fixed spatial wavelength, a greater electrode area generally means higher sensitivity.

The principle of the capacitive sensor is based on fringing electric field, so the mutual capacitance between two adjacent electrodes is relatively small, that leads to low signal strength. The signal strength decays exponentially with its distance to the object. By integrating more electrodes into one single transmitter/receiver, the signal strength can be improved.

The generated electric field is easily being interfered by approaching objects from undesired directions, and that will cause noise issues. Adding guard electrodes is an effective way to shield sensing electrodes from noises. The guard electrodes need to be carefully connected to avoid stray capacitances and to be positioned properly for optimal sensor performance. The guard electrodes can take the form of a guard ring surrounding the sensing electrodes, the guard plane underneath the sensing electrodes, or a 3D shield around the sensing area. The driven-guard technique presented in [124] can be used to reduce stray capacitances, where the guard electrodes are connected to the same excitation signals as the sensing electrode.

3.2.1. Electrode structure based on rigid substrate

Initially, the multi-functional coplanar capacitive sensor is intended to be fabricated on a rigid substrate. The scales of all the parameters for modeling are considered under the condition of using a rigid printed circuit board (PCB). The design issues include the

geometrical parameters of the electrodes, shielding strategies, the number of electrodes, and electrode connection patterns. To model and quantify the capacitive performance, the electrostatics model inside COMSOL 3D software package is employed to mesh the geometries and calculate the mutual capacitances. The basic simulation conditions are identical for all the studies: the thickness of the electrode is 0.1mm; the excitation voltage is 1V; the zero charge boundary condition is applied to the exterior air environment; the electrode material is copper. The simulation is electrostatic, which is adequate for the low operating frequency of this sensing system.

Among all the design variables, electrode geometry is the dominant factor for the sensor performance. Therefore, modeling and selecting the most proper electrode size, shape, spacing, and separation is critical in order to meet the requirements.

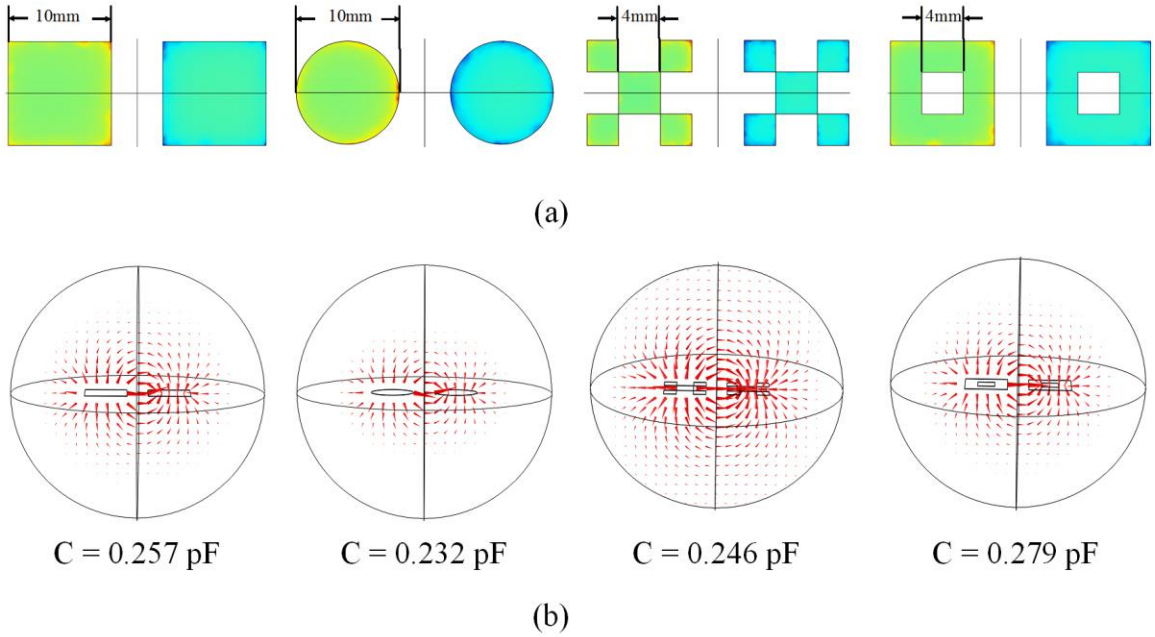


Figure 3.4. Investigation on different electrode shapes. (a) Electrode shape and dimension. (b) Simulated electric field and mutual capacitance

In addition to the traditional design featuring square or round electrodes, two more electrode shapes namely fractal and hollow square are taken into consideration. Electrodes with the four different shapes have identical principle parameters (i.e. diameter or the outer side length) of 10 mm as shown in Figure 3.4 (a). The excitation voltage is applied to the left electrode, and the electric charge distributes throughout the electrode to minimize the Coulombic energy with the amount of charge set by the capacitance [125]. The geometric parameters contribution to capacitance includes effective area and

perimeter. The increase in the effective area causes a net effect of maximizing the mutual capacitance, an electrode with larger bounding perimeter holds more charges due to the fact that much of the charges reside on the bounding [126]. The charge distribution simulations (also shown in Figure 3.4 (a)) provide a physical explanation for why the hollow square electrode holds even more charge though with smaller effective area compared to the square electrode. Further analysis is conducted to evaluate the distribution of the electric field. Figure 3.4 (b) shows the simulation results of the spatial electric field distribution of each pair of electrodes. For all the electrode shapes, the magnitude of electric field strength becomes weak as the field goes up. The electric field strength needs to be sufficient in order to interact with approaching objects [127], that means the slower the electric field attenuates the longer detection distance the sensor could achieve. From the simulation results, the electric field generated by fractal electrode pair penetrates the highest into space, followed by the one generated by square electrodes. To achieve the best trade-offs among the charge storage, electric field strength, and manufacture complexity, the traditional square-shaped plate was chosen as the basic construction unit in this design.

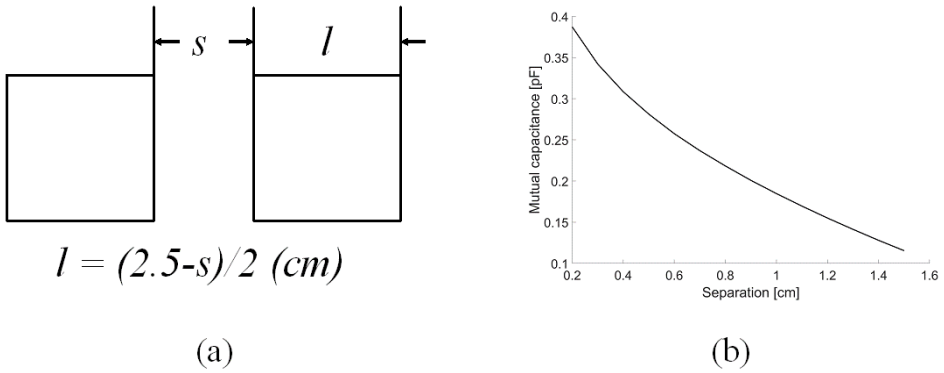


Figure 3.5. Mutual capacitance influenced by electrode separation and side length. (a) Relationship between separation s and side length l within one electrode pair. (b) Simulated mutual capacitance vs. separation.

The number of electrodes depends on the complexity of the sensor itself as well as the number and complexity of the system variables to be solved for the sake of the target applications. Provided that the sensitivity is adequate, the desired information about distance and object geometrical properties can be extracted from the mutual capacitance between two electrodes within an electrode-array. The total number of independent two-electrode measurements (M) can be expressed as M = N(N-1)/2, where N is the number of electrodes in the array [128]. There could be more measurements when taking multiple

electrodes alignments into consideration. In this work, a 4×4 electrode matrix yielding 16 independent electrodes is selected as the basic configuration as it meets the functional requirements with an acceptable complexity at the same time. In the context of the applications where the designed sensor should be fitted on a piece of working garment, the electrode matrix area is restricted to 6.5×6.5 cm^2. Consequently, the length of each sensing unit that composed of two electrodes should be limited to 2.5cm. A series of combinations contain different separation and electrode length values are studied. As demonstrated in Figure 3.5 (a), the separation *s* and electrode side length *l* are inversely correlating to each other. The mutual capacitance as a function of different separation values from 0.2cm to 1.5cm is plotted in Figure 3.5 (b), and it can be observed that the capacitance decreases as separation increases. However, the depth that the electric field penetrates to space is positively correlated to the separation, that causes the trade-off between penetration depth and electric field strength. In this work, the separation of 0.5cm that results in the side length of 1cm is selected as the basic scale. With these parameters, the obtained absolute capacitance value is on the large side and the penetration depth of the electric field generated by two neighboring electrodes does not decrease much. Moreover, the penetration depth can be adjusted by using electrodes from different locations in the matrix.

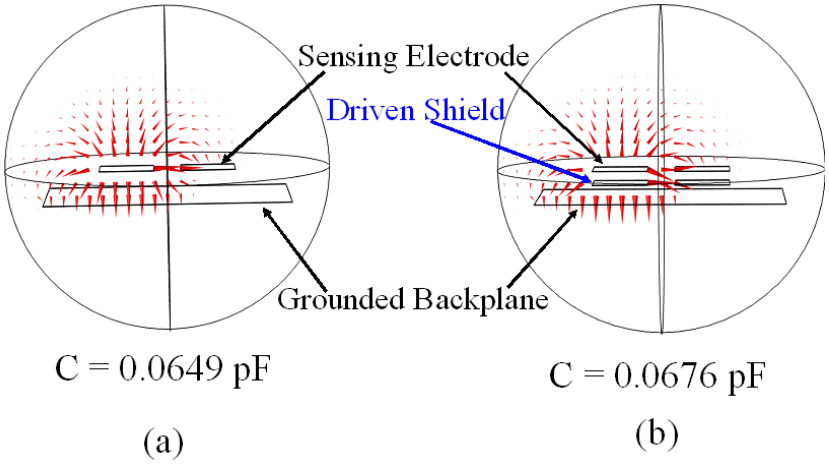


Figure 3.6. Shielding effects. (a) With a grounded backplane. (b) Adding driven shield electrodes.

As studied and shown in Figure 3.4 (b), the generated electric field penetrates to the space from both sides of the sensor, that may cause undesired detection from the back-side of the sensor. In addition, the capacitive measurements can be as low as femtofarads. At such low levels, the reduction of electrical noise through electromagnetic

shielding and output signal filtering become particularly important [129]. A grounded backplane deposited on the bottom side of the substrate is able to provide shielding from external perturbations, define ground potential, and predominantly confines the electric field to the upper half-space. However, with the adding of the backplane, a large portion of the electric field is concentrated in between the electrode and the backplane as shown in Figure 3.6 (a). Consequently, the mutual capacitance drops dramatically from 0.257pf to 0.0649pf given by the simulation. Another shielding element that is inserted between the electrode layer and backplane layer is used to neutralize the side effect of the ground shielding and optimize the sensitivity. The driven-shield technique, where the shield electrode is kept at the same voltage potential as its corresponding sensing electrode is proven to be effective in removing or reducing the stray capacitances [124]. The cross-section view of the updated sensor unit structure and the simulated electric field distribution are shown in Figure 3.6 (b). In this simulation setup, the shield electrode has the same size as the sensing electrode. The mutual capacitance is slightly improved from 0.0649pf (only with grounded backplane) to 0.0676pf (with both active shielding layer and the backplane).

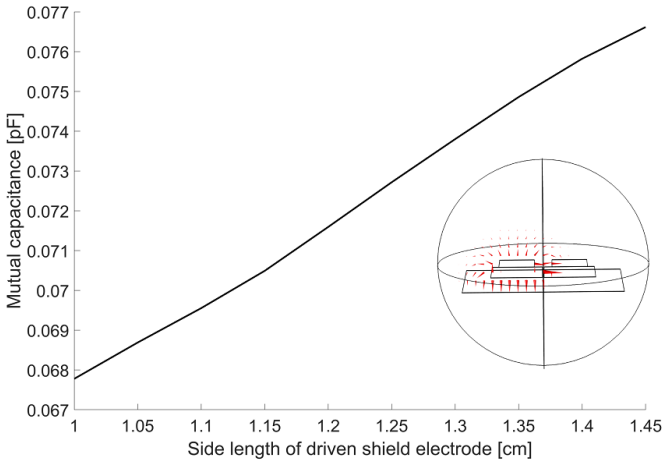


Figure 3.7. Mutual capacitance vs. different sizes of driven shield electrode.

A further investigation about the active shielding electrode is about its size. A series of simulations have been performed with the side length of the active shielding electrode increasing from 1 cm to 1.45 cm. The mutual capacitance between two sensing electrodes as a function of the size of the active shielding electrode is plotted in Figure 3.7. With the increase of the active shielding electrode's size, the mutual capacitance

increases. Therefore, 1.45cm is used as the side length of the active shielding electrodes in this design.

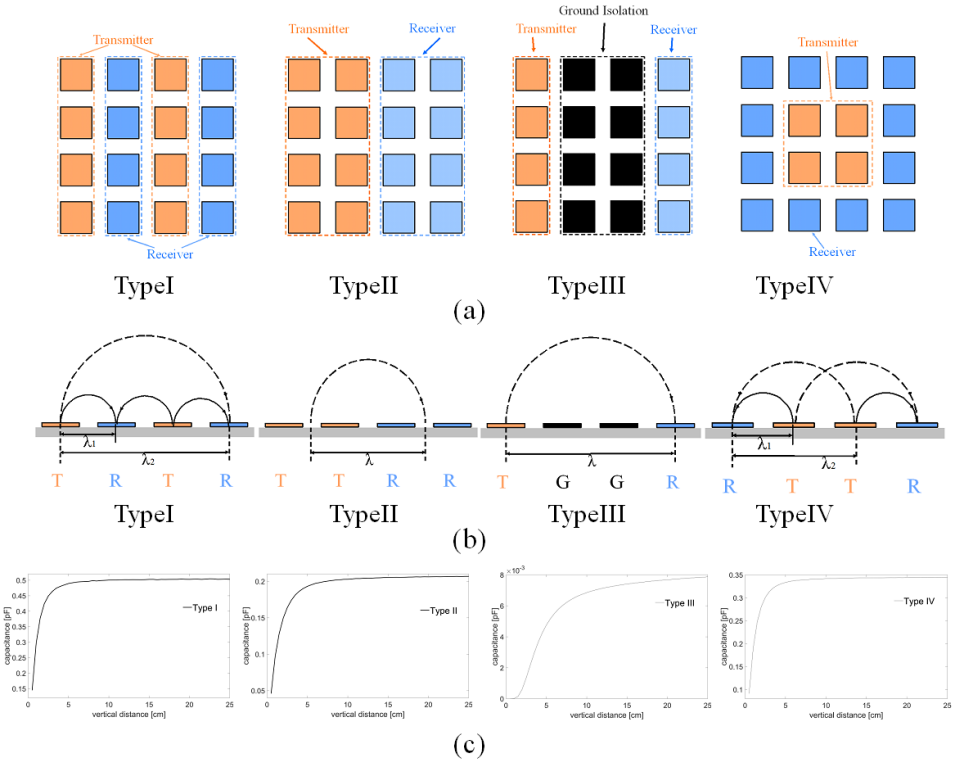


Figure 3.8. Electrode connection types. (a) Four different electrode group types. (b) Different spatial wavelengths generated by the four types. (c) Capacitive responses from the four connection types.

The sensing performance of the array-structured proximity sensor typically depends on the geometrical parameters including the number and the arrangement of electrodes that forming the transmitter and the receiver. The optimal sensor parameters are achieved by proper selection among different electrode connection patterns. The excitation voltage source is applied to two group of selected electrodes that perform as the transmitter and the receiver. Figure 3.8 (a) illustrates the four connecting possibilities of the designed electrode matrix. In Figure 3.8 (b), the side view of the sensor showing how the electric field is formed between positive and negative terminals is given. It is seen that the penetration depths of electric field lines is different for different spatial wavelength λ that is defined as the distance between two consecutive electrodes of the same polarity [130]. The penetration depth can be increased by increasing this pitch length; however, the electric field strength generated at the neighboring electrodes will be weak. Type I represents the interdigital mode that is one of the most widely applied proximity sensing

structure. Type II only generates one electric field but with stronger strength [131]. Type III is selected as a promising candidate as it generates the largest spatial wavelength. Type IV is a symmetric structure that works similarly as a ring-shaped sensing pattern. These sensing possibilities would provide various responses that enable us to dig out the most suitable working patterns for different application scenarios.

For FEM simulation setup: the sensor is composed of a sensing electrode array, an active shielding electrode array, and a grounded backplane. The same as the sensing electrode matrix, there are four different connection schemes within the active shielding array as each shielding electrode is driven by a buffer operational amplifier to stay in the same voltage as the sensing electrode immediately above. A copper plate with the size of 7cm × 7cm × 0.01cm is used to mimic the approaching object. All these components are placed in an air box whose dimension is much larger than the sensor and the object. The box is employed to complete the simulation model: it acts as an electrically insulated space, where the charge at the infinity balances the internal charges. The vertical distance between the object and the sensor *d* varies from 0.5cm to 25cm for the study of proximity sensing performance. The capacitance between the formed transmitter and receiver pair as a function of the vertical distance for all the four connection types is plotted in Figure 3.8 (c). In order to investigate the linear detection range and the effective penetration depth, the normalized capacitance as a function of vertical distance for all the four types is summarized to one plot as shown in Figure 3.9 (a). More specifically, given the maximum capacitive value C_{max} and the lower bound C_{min} for one set of measurement results, a normalized value is calculated by $C_N = (C-C_{min})/(C_{max}-C_{min})$ to keep the values in the [0,1] range [132]. Penetration depth $\gamma_{3\%}$ corresponds to the vertical distance d where the difference between the capacitance at this position $C(d = \gamma_{3\%})$ and the asymptotic capacitance $C(d = \infty)$ equals to 3% of the difference between the highest and the lowest values of the capacitance [124]. The evaluation method is illustrated in equation (3.17) and Figure 3.9(b), where C(d = 0) represents the capacitance when the distance between the sensor and the object is minimum. The calculated penetration depths are listed in Figure 3.9 (a).

$$\frac{C(d=\infty)-C(d=\gamma_{3\%})}{C(d=\infty)-C(d=0)} \times 100\% = 3\% \qquad (3.17)$$

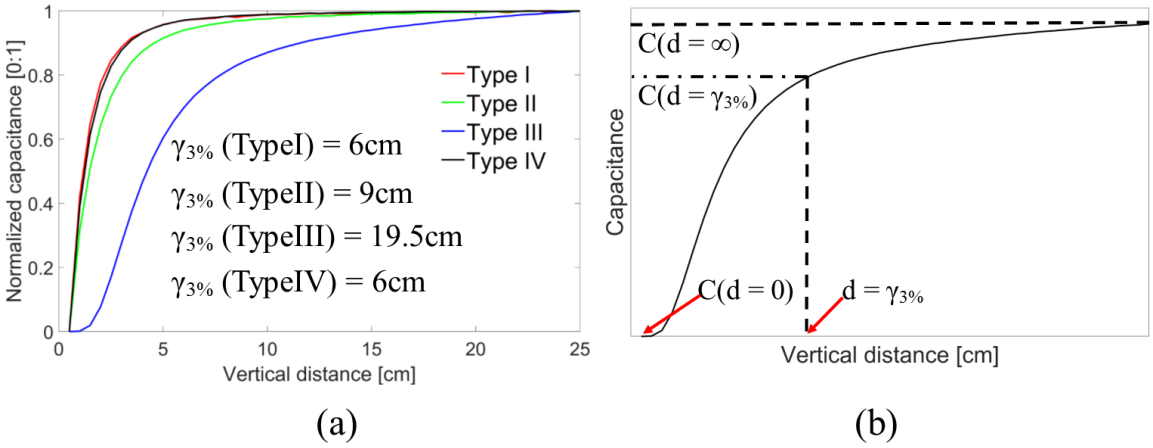


Figure 3.9. Capacitive comparison among the four electrode connection types. (a) Normalized capacitance vs. vertical distance. (b) Explanation on penetration depth calculation.

From the results shown in Figure 3.8 and Figure 3.9, the absolute capacitance obtained from type III is significantly less than the other three types. However, Type III offers a larger penetration depth that is essential in detect range. In the meantime, interdigital mode and ring-shaped structure can provide larger electric field strength that improves the immunity to environmental noises.

One of the challenges of using capacitive sensors is they are susceptible to shape and size of the object it perceives, that makes it difficult to infer high-level information from the measured capacitances. A small object that is close to the sensor might result in the same response as a larger object at a further distance [133]. Moreover, the relative horizontal location between the object and the electrode-matrix also makes a difference in measurement results. Thus, a model is required that evaluates the behavior and influence of different shapes and locations of the object within the generated electric field.

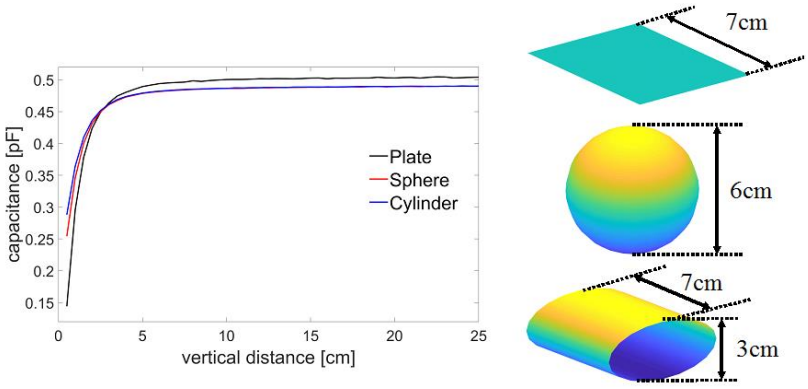
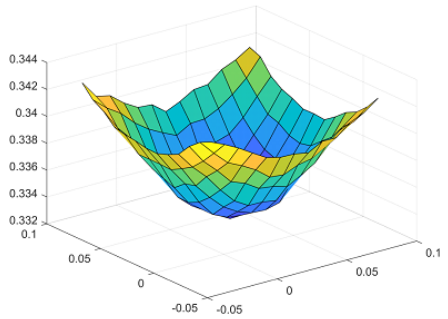


Figure 3.10. Capacitive responses from three shapes: plate, sphere, and cylinder.

Typically, the actual shape of the desired object is approximated by simple geometric shapes that are easier to process. Plate, sphere, and cylinder are selected for modeling the most frequently encountered shapes in the industry working space. The area of the plate is 7cm × 7cm; the diameter of the sphere is 6cm; the radius and the height of the cylinder are 1.5cm and 7cm respectively as shown in Figure 3.10. Based on the previous simulation results, Type I that represents the interdigital connection mode is chosen as an example to illustrate the effects brought by different shapes. The simulation process is similar to the aforementioned one: the object moves vertically so that the distance d from the sensor to the lowest point of each object varies from 0.5cm to 25cm. The three sets of responding capacitance as a function of vertical distance are plotted in the same coordinate axis for comparison (also shown in Figure 3.10). It can be observed that the shape of an object affects the capacitive responses in terms of absolute values as well as the slope.

One more set of simulations is performed to investigate how different horizontal positions affect the vertical distance evaluation and to figure out which connection type could neutralize this effect the most. In this simulation, the mutual capacitances are measured when the object is performing a grid scan at the plane with a vertical distance of 5cm. The same process is repeated for each object with each electrode connection pattern. Figure 3.11 (a) depicts the capacitances obtained from ring-shaped sensing type (Type IV) with respect to the relative horizontal positions of the plate. For all the three objects with all the connection types, the measured capacitance shares the same trend: the capacitance value increases when the object moves from the center to the edge of the electrode matrix. In order to understand the total span and variation of the data, the variation percentage that can be expressed as $(C_{max}-C_{min})/C_{max}$ is calculated and also be listed in Figure 3.11 (b). A smaller variation percentage indicates the horizontal position has a smaller effect on distance evaluation under the certain connection type. In general, detection for a sphere is affected the most from its horizontal positions; on the contrary, horizontal positions of a cylinder make the least difference. For a plate, the ring-shaped pattern (Type IV) provides the minimum variation, whereas the interdigital mode (Type I) works the best for both the sphere and the cylinder. A remarkable point can be achieved at this point is that for distance evaluation, the uncertainty and variation brought by horizontal positions is much less than that by different shapes.



(a)

Variation	Type I Interdigital	Type II Two-group	Type III Two-column	Type IV Ring-shaped
Plate	14.8%	3.14%	34.8%	1.57%
Sphere	12.9%	22.9%	316.2%	13.2%
Cylinder	4.5%	9.89%	87.5%	5.03%

(b)

Figure 3.11. Investigate on the effects from horizontal positions. (a) Simulated capacitance vs. grid locations on the same plane. (b) Calculated horizontal variation percentages for all the object and connection types.

After all these comprehensive simulations, quantifying the distance between the object and the sensor accurately is not an easy task, and requires multiple moves. Based on the performed studies, sensing electrode matrix with a complementary active shielding electrode array and a grounded backplane represents the best approach in terms of generating electric field generation. The shape of an approaching object causes the most uncertainty in capacitive responses with respect to vertical distance. This ambiguity leads to the inaccuracy in distance evaluation. Identifying the shape of an object first, and then choose the most suitable connection type (i.e. the one causes the least capacitance variation among different horizontal positions) to measure the capacitance can be a good strategy to improve the sensing performance.

3.2.2. Electrode structure based on flexible substrate

Flexible circuits are a high-growth technology in the area of electrical interconnectivity and look set to deliver improved performance against the demands of a wide range of electronic products [134]. Flexible sensors have emerging applications in biomedicine, artificial skin, and wearable electronics. A lot of factors contribute to the allure of flexible electronics: they are typically more rugged, lighter, and portable compared to their rigid substrate counterparts [135]. Flexible sensors are able to offer a wearable and longtime non-intrusive monitor. The electrodes can be cut and unobtrusively integrated into a piece of working garment to be worn during the whole working process. For human-robot interaction applications, in order to realize a convenient, comfortable, and reliable measurement that results in a sufficient protection for human workers, the use of

sensorized garments is a suitable strategy. The design concerns and the sensing properties of the flexible capacitive sensor are investigated in this section.

One of the most commonly used methods to obtain a sensor flexible is to fabricate the device directly on a flexible substrate. This is the approach adopted in this work to transform the designed sensor into a flexible version. There are some challenges that tie to the flexible substrate: very low amplitude signals, and complex relationship between signal and measured phenomena [136]. As the thickness of isolation layer for a flexible substrate is much smaller than a traditional rigid substrate, thus, the strength of electric field between the electrode and the backplane is stronger. That leads to a smaller mutual capacitance between an electrode pair. Moreover, the capacitive responses for a flexible sensor not only depending on the approaching object but also dependence on the nature of the structural changes (bent angles) of the sensor itself. Instead of simply duplicating the whole design into a flexible substrate, more emphasizes are put to cope with these new challenges.

As a fringing electric field sensor, the geometrical parameters including size, shape, and the separation of the electrodes are the key factors that determine the sensing properties. The investigation of these parameters is the same as the one being performed for the rigid sensor. Based on the simulations that investigate the relationship between the acquired mutual capacitance, electrode shape, size, and shielding strategies (the results were demonstrated in Figure 3.4, Figure 3.5, and Figure 3.7), two hollow square-shaped electrodes with the side length of 1cm and separation of 0.5cm are selected as the building unit for this flexible design. Again, driven shield and grounded backplane are applied to shield out the detection from undesired direction. The attractive properties of the hollow square shape include its ability to hold more charge and its smaller area that reduces the parasitic capacitance to the backplane. Another essential modification being made to the flexible sensor is to shrink the array structure to a band-structure with just one row of electrodes. This change is made under the consideration of practical applications: the flexible sensor is most probably being used as or integrated into a soft wristband. By adopting multiple sensors, it is possible to cut and rearranged the electrodes to create an electrode-matrix if necessary. One band contains five electrodes that can be connected differently for the purpose of different detecting requirements.

In order to gain a better insight into the characterizations under different bending angles, a set of simulations are performed. The basic simulation setups are shown in Figure 3.12 (a): the electrode-band is attached on the surface of a cylinder, and a plate which is used to mimic the detection target moves from 1cm to 20cm away from the sensor. Different bending angles are realized by changing the radius of the cylinder from 3cm to 8cm. This range is decided according to the general sizes of human arms and working garments. Based on the study about dielectric properties of the human body in [137], water is chosen as the material of the cylinder to imitate the human body. The area of the plate is 7cm × 7cm, and the material used for the object and the electrodes is copper that is conductivity.

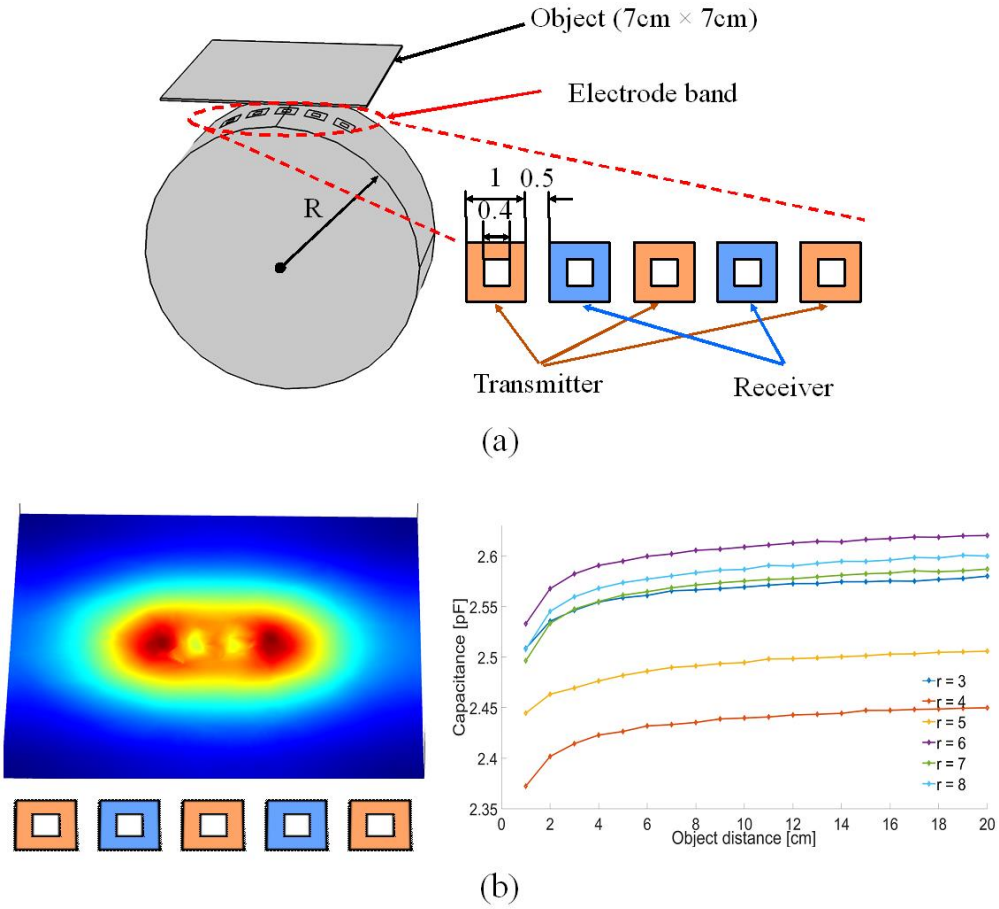


Figure 3.12. Simulations of the sensor on the flexible substrate. (a) Simulation setups. (b) Simulated results in terms of electric field distribution and capacitance as a function of object distance and radius of the attached cylinder

The inner connection of the five electrodes is the comb-type, in which three electrodes are connected as the transmitter and the rest two electrodes work as the

receiver (also described in Figure 3.12 (a)). The simulated electric field distribution and the obtained capacitance vs. different cylinder radiuses are plotted in Figure 3.12 (b). It can be seen different bending angles would result in quite different capacitive responses: Not only in terms of absolute capacitive values but also the changing rate with respect to the vertical distance of the object. This finding provides a guidance in practical application of the flexible sensor: Some pre-tests include attaching the electrode-band to different parts of the human body and/or to different persons should be performed to calibrate the raw capacitive sensing data. This procedure would help in improving the accuracy of the desired vertical distance information.

3.2.3. Sensing method

The designed sensor has 16 independent sensing electrode, providing many possible configurations to generate the fringing electric field so that multiple functionalities can be achieved by the same sensing platform. The aiming sensing capabilities include proximity sensing capability, surface profile recognition, improved distance estimation, and parallel motion tracking. Selecting the most suitable connection patterns for each functionality plays a significant role in applying the designed sensing system most effectively. All the potential connection configurations are demonstrated in Figure 3.13 in the form of matrices: each letter represents the connection of each electrode (“T” represents the transmitter, “R” stands for the receiver, and “G” means ground connected), and electrodes labeled by the same letter are connected.

Penetration depth that determines detection range is an intrinsically important factor when employing the system as a proximity sensor. Four promising candidates corresponding to the four connection types investigated in the simulations (Section 3.21) are illustrated in Figure 3.13 (a). Among those types, Type III that utilizes only the two most remote electrode strips as transmitter and receiver is chosen for the basic proximity feature. This type stands out because it generates the largest spatial wavelength λ and it has been verified in the previous simulation to have the largest penetrate depth.

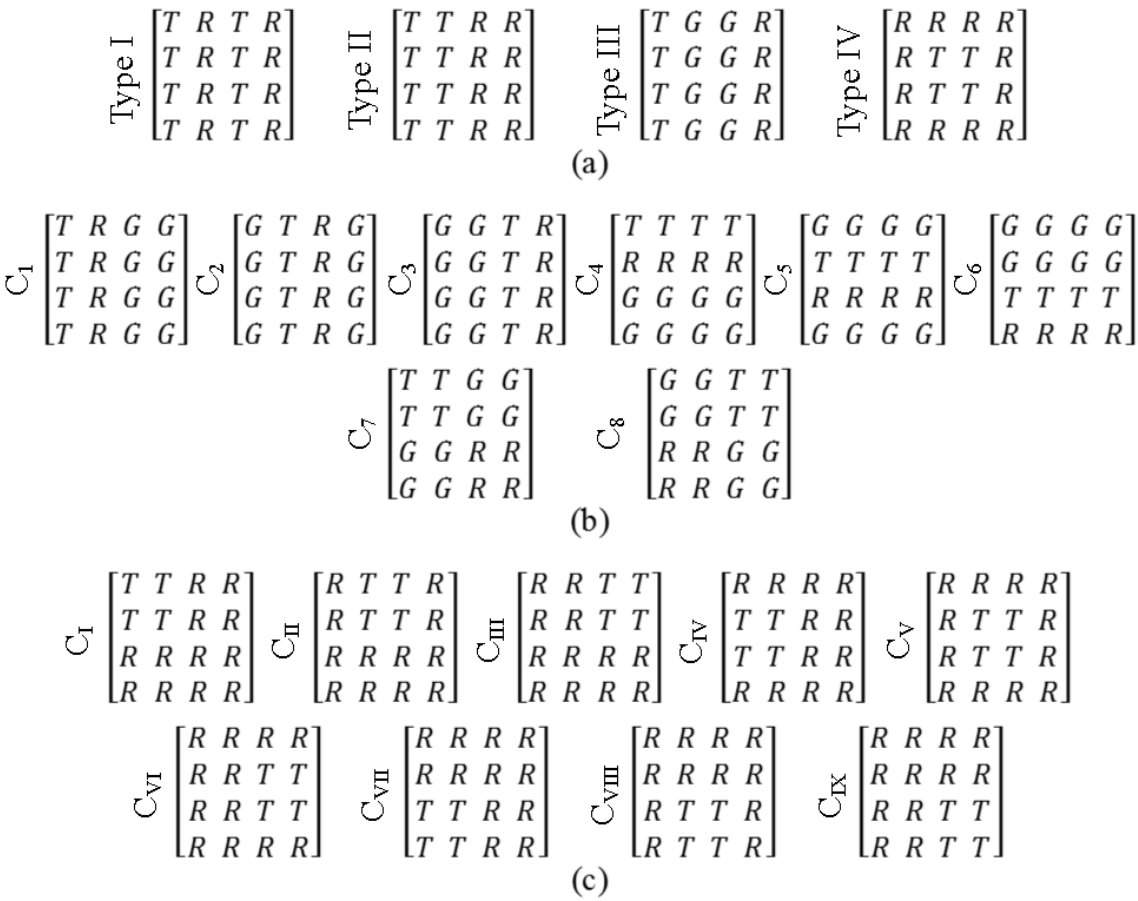


Figure 3.13. Electrode connection patterns for different operation modes. "T": transmitter; "R": receiver, "G": ground. All the capacitors are formed between transmitter and receiver. (a) Four connection types used in basic proximity sensing and improved distance estimation. (b) Surface profile recognition configuration I: eight independent capacitors (C1 – C8) based on the connections to mesh the nearby surrounding. The first three generated capacitors (C1 – C3) are measured for parallel motion trajectory detection. (c) Surface profile recognition configuration II: nine connection configurations that can provide electric field with various parameters.

For the purpose of distinguishing different object shapes, the inputs of the classification tools are the readings of several distributed proximity sensor from the electrode array. Management of electrode configurations plays an essential role in acquiring the shape information. The key point in reconstructing the desired obstacles is to distinguish the differences in terms of sensor responses among different circumstances. Two different approaches that can be performed to mesh the close surroundings are given and explained in detail as follows.

The electric field between adjacent columns and adjacent rows can be generated by connecting the electrodes in a way as shown in Figure 3.13 (b), the upper part. Six individual capacitors are formed and the nearby space can be well meshed along the X-Y plane. Based on the surface topography, cylinder tends to have similar capacitive responses as sphere along X-axis. To reduce this obscurity, two more capacitance between diagonal eight electrodes as depicted in Figure 3.13 (b) (i.e. C_7 and C_8) are measured. Totally, eight sets of independent capacitances (C_1 to C_8) are acquired in this approach method for surface profile recognition.

The inspiration for the second approach comes from the ring-shaped connection structure: the electric flux emits from the central four electrodes in every direction and falls to the receivers on the outer circle of the sensor. The environment around the sensor can be meshed comprehensively when employing this mode to the whole sensing matrix as shown in Figure 3.13 (c). Each of the nine connection patterns generates multiple electric fields with different spatial wavelength λ and different penetration depths. By combining these nine capacitive responses (C_I to C_{IX}) together, the surface profile of the nearby object can be estimated.

With the shape information, the primary goal of the project, estimating the distance with high accuracy, is put back on the agenda. All the four connection types demonstrated in Figure 3.13 (a) are considered as each of them processes different advantages in terms of signal strength, penetration depth, and horizontal variation.

The last functionality that can be achieved by the same sensing system is parallel motion tracking. More attention is paid to the moving direction in this scenario, whereas the absolute vertical distance is of secondary interest. Thus, the trajectory history of the target object has to be analyzed continuously. In this context, parallel movement assumes an object moves parallel to a reference axis with low deviations to the other two orthogonal axes in 3-dimensional space. Electrodes are connected to four strips along the reference axis as visualized in Figure 3.13 (b), the first three matrices. Thus, three mutual capacitors between two adjacent columns can be formed. These capacitors are measured in turn periodically to scan the surroundings. The interactions between the moving object and each capacitor would have the same shape but with certain delays in the moving direction. The parallel motion of an object can be discriminated against its past cluster centers

obtained by any of the three mutual capacitors, and the moving direction can be visually recognized by combining the three sets of measurements together.

The performance of the sensing modalities of all the electrode connection patterns for every functionality will be experimentally assessed in Chapter 5.

Chapter 4.

Data Processing

Capacitive sensors are difficult to model analytically due to their inherently nonlinear characteristics. As explained in Chapter 3, the analytical models for fringing electric field sensors are based on idealized assumptions and restrict boundary conditions that limit their accuracy for real-world applications. So that acquiring desired shape and distance information relies heavily on numerical experiments and data processing. Statistical learning that refers to a vast set of tools for understanding data is used as the data processing tool. It can be further characterized as either qualitative or quantitative. Quantitative regression models are built to seek out distances while adaptive classification tools are employed to recognize the surface profiles.

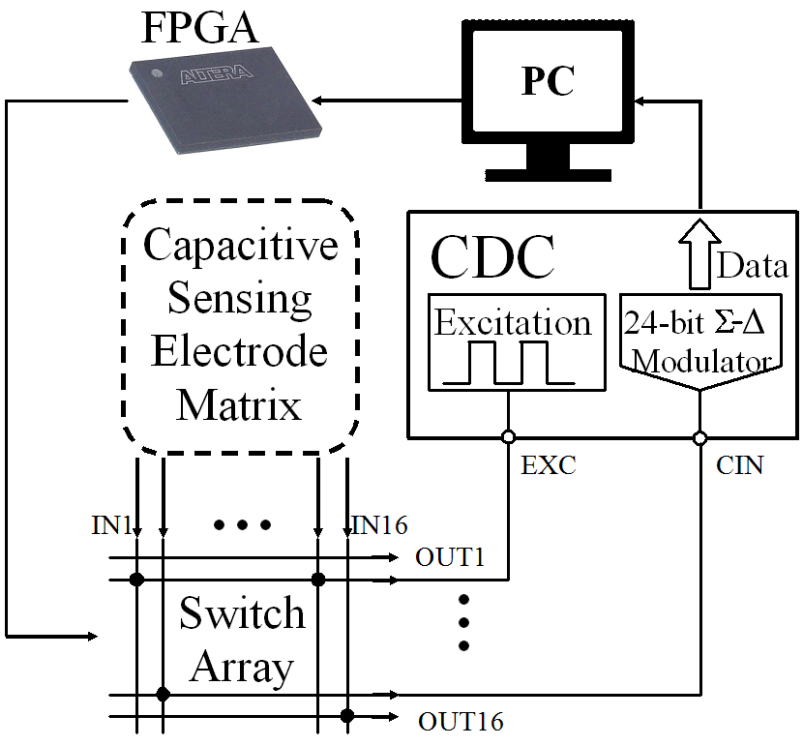


Figure 4.1. Block diagram of the designed capacitive proximity sensing system.

In order to implement the different inner electrode connection configurations and extract the capacitive responses, the capacitive proximity sensing system is composed of five building blocks as described in Figure 4.1. The 4×4 electrode matrix together with its active shielding electrode array and the grounded back-plane constitute the core sensing

section. Two analog switch arrays are used to generate physically connections within sensing and driven shielding electric matrices. A digital controlling module that is mapped on a field programmable gate array (FPGA) creates desired signals to program the switch arrays. A capacitance to digital converter is obtained to measure the capacitive responses and convert them to strings of digital data. Finally, an external personal computer (PC) is involved to collect data and perform signal processing.

In the human-robot interaction applications, we are primarily interested in the capacitive responses related to the approaching object, it is important to separate the distance/shape/motion variations from obscuring sources of variability introduced by the interference. The quality of data collecting and processing is, therefore, an essential step to interpret capacitive responses properly.

Before delving into the question of how the signal processing methods operation, we begin by understanding the input data. The input takes the form of concepts, instances, and attributes [138]. The idea of a “concept” is the result of the signal processing, which in this case be the distance value or the shape of the object. The information that is fed to the processing tools takes the form of a set of “instances”. One instance represents one individual, independent example of the target concept. Each instance is characterized by the values of “attributes” that measure different aspects of the instance. In this project, the values of mutual capacitors under known concepts are attributes. Based on different processing purposes, these attributes should be fused differently to form the instances. The generated instances together with the concepts are utilized to build the regression models and to train the classifiers. In practical applications, the obtained statistical tools will be applied to the actually measured capacitances so that the desired information can be extracted directly.

4.1. Data preparation

The goal of data processing is to obtain desired information from the raw measurement results. Such a task is inherently interactive and iterative [139]: one cannot expect to gain useful knowledge simply by feeding a lot of data to a random tool. A solid understanding of the problem domain is essential in collecting data and selecting suitable classes of patterns. In this project, the ultimate goal is to derive the distance between an object and the sensor platform accurately. As specified in the last chapter, recognize the

surface profile of an approaching object and then extract the distance information stands for a good strategy. To support the hypothesis about the problems that intend to be solved, data collection forms the core to present any reasonable predictive analytic solution. However, until actually train a predictive model, it is hard to know which attributes will contribute the most and provide the best results.

Depending upon the domain of this project, the input attributes are measurement results. The experiments are designed to restore the scenarios in realistic applications: the metallic plate, sphere, and cylinder are used to mimic the three surface profiles that are most likely to be encountered during a manufacturing process; the object is moving from 1cm to 20cm away from the sensor to cover the interested distance range. All the electrode connection patterns described in Chapter3 are implemented through programming the switch array, and the capacitive responses are collected to form the data sets. More detailed data acquiring methodologies will be explained together with the experimental setups in the next chapter.

Though the data collection process is under adequate consideration, the experimental procedures might still highly liable to inner parasitic of the sensor, environmental interferences, noises, and unexpected mistakes. These factors result in missing values, inconsistency data, drifting, noises etc. [140], and they are the main sources of ambiguities in data processing results. Therefore, data pre-processing includes cleaning, normalization, integration and reduction [141] is the foremost step before running the analysis. More specifically, data cleaning is used to remove noise and correct drifting in the data set. Data normalization (also known as transformation) that scales numeric attributes to a smaller range like 0 to 1 to improve the accuracy and efficiency of data processing algorithms. Data integration merges data from multiple measurements into a coherent data store, whereas data reduction reduces data size by performing principal component analysis (PCA). These techniques are not mutually exclusive; they work together in many cases [142].

Among the mentioned pre-processing techniques, except for data reduction, all the rest are straightforward in terms of understanding and conducting. Principal component analysis (PCA) is the general name for a technique that involves sophisticated underlying mathematical principles to transform the instances that are composed of several inter-correlated quantitative dependent attributes into smaller dimensional instances [143]. The

goal of PCA is to extract the most important information from the data; compress the size of the data set by keeping only the new orthogonal attributes called “principal components”; simplify the description of the data sets, and analyze the structure of the instances and attributes [144].

The principal components are the linear combinations of the original attributes. Covariance is considered to be a measure of how well correlated two variables are, and PCA method makes the fundamental assumption that the components in the transformed instances should be as uncorrelated as possible. The first principal component represents the one with the largest possible variance. The second component is computed under the constraint of being orthogonal to the first one with the largest possible inertia. This procedure is continued until all the principal components are acquired. The values of these new attributes are called factor scores, and these factors scores can be interpreted geometrically as the projections of the instances onto the principal components [145]. Depending on the rank of the factor scores, the components that are most related to the others can be discarded. So that the X attributes in original instances can be replaced by Y variables (Y<X). In this way, the influence of noise is minimized as the original attributes are replaced with weighted averages; the interpretation and visualization are greatly aided by having a simpler view of all the variations [146]. The way for finding the principal components can be explained with five steps: Firstly, all the data is standardized to the same scale. Then calculate the covariance matrix of the data. After that, the first principal component can be decided as the sum of the squared distances from the points to this component is minimized. Consequently, a second component that is orthogonal to the first one is obtained. Finally, rotate the figure of all the scaled data in order to plot all the points into the PCA coordinate system and repeat the last three steps for all the rest principal components.

In summary, real-world experimental data tend to be dirty, incomplete, inconstant, and sometimes redundant. Data pre-processing techniques can improve the quality of the raw data, thereby helping to improve the accuracy and efficiency of the subsequent processing procedure. Typical data processing pipeline usually proceeds through multiple stages as shown in Figure 4.2, and it should always be remembered that those methodologies are not mutually exclusive.

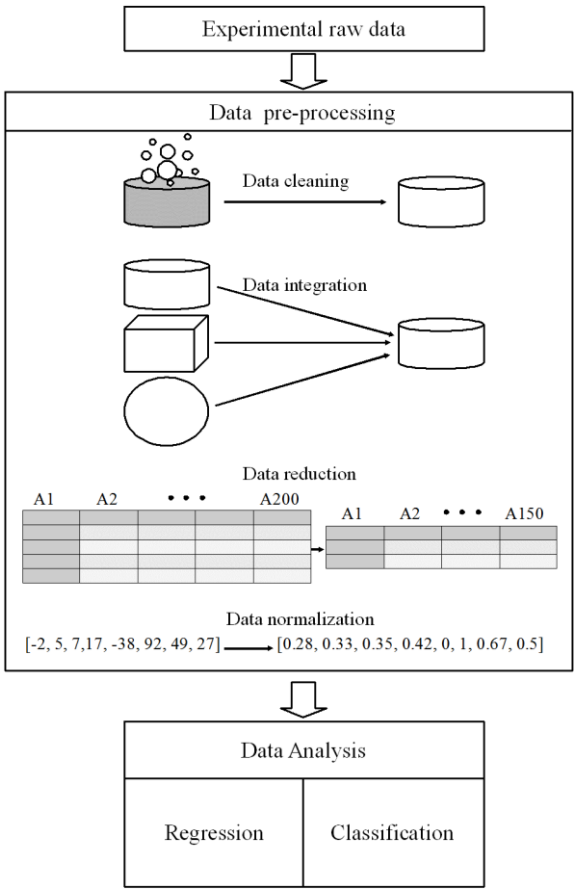


Figure 4.2. Summary of data processing workflow.

4.2. Distance estimation

The distance estimation task for this project is quantitative: the results take on numerical values. The problems with a quantitative response are referred to as regression problems [147]. Regression analysis is a branch of statistical methodology concerned with relating a response to a set of input variables. The goal is to build a regression model, which is a prediction equation, that enables us to predict response for given values of inputs and to do so with a small error [148]. Regression analysis of experimental data is a powerful statistical tool that provides a technique for building a predictor of a piece of interested information (i.e. the distance value in this project) and enables users to place a bound (an approximate upper limit) on the error of prediction.

In regression analysis, the outcome of interest is called the dependent variable and demoted with Y, and the subscript on Y identifies the particular unit from which the experiment is taken. Most commonly, the regression is aimed at reflecting how the mean

of the output changes with changing inputs assuming the variance of Y is unaffected by different inputs [149]. The input variables that provide information on the behavior of dependent variables are incorporated into the model as predictors and being demoted by X. In addition, all the unknown constants in regression models are called parameters, usually being denoted by Greek letters. They control the behavior of the model. In this specific project, Xs are the capacitive responses obtained experimentally and Ys represent the corresponding distance values. The parameters are estimated to give out the best fit of each predictor to its corresponding output. In practical, when the newly measured capacitance is reported, the distance value can be directly calculated.

Among all the regression approaches, linear regression is the basic approach for supervised learning. Though it seems dull compared to most of the modern regression methodologies, it is a useful and widely used tool, especially in engineering problems. The simplest linear model, straight-line model, involves only one predictor X and states that the true mean of the dependent variable Y changes at a constant rate with the changing of X. The functional relationship between the true mean of Y_i, denoted by $E(Y_i)$, and X_i is the equation of a straight line as in (4.1). In which β_0 is the intercept, the value of $E(Y_i)$ when X = 0, and β_1 is the slope of the line, the changing rate in $E(Y_i)$ per unit change in X. The subscript i indicates the particular experimental unit, i = 1, 2, …, n.

$$E(Y_i) = \beta_0 + \beta_1 X_i \qquad (4.1)$$

The observations on the dependent variable Y_i are assumed as random outputs from populations of random variables with the mean of each population given by $E(Y_i)$. The deviation of an observation Y_i from its mean $E(Y_i)$ is taken into consideration by adding an error term ε_i. ε_i represents all unexpected variations caused by important but omitted variables or by unexpected random phenomena. Consequently, the probabilistic model is modified as shown in Equation (4.2).

$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i \qquad (4.2)$$

The random error ε_i plays an important role in the test of the hypotheses or finding confidence intervals for the deterministic portion of a regression model. It also makes it possible to estimate the magnitude of the error of prediction when the model is applied to predict the value of Y based on the new observations in the future [150]. ε_i has zero mean and is assumed to have common variance σ^2 and to be pairwise independent. For purposes of making tests of significance, ε_i is also assumed to be normally distributed as

described in Equation (4.3), which implies that Yi is also normally distributed. NID stands for “normally and independently distributed”, and “0” and “σ^2” inside the parentheses denote the mean and the variance respectively.

$$\varepsilon_i \sim NID(0, \sigma^2) \qquad (4.\ 3)$$

The random variation makes each pair of inputs provides different results so that a method is needed to combine all the information and finally give out one solution which is the “best” by some certain criterion. One way to quantitatively decide how well a model fits a set of data is to determine the extent to which the data points deviate from the line as illustrated in Figure 4.3. The magnitude of deviations, also be expressed as errors of prediction, are the differences between the experimental and the predicted values of Y. The sum of squares of the errors (SSE) that is described in Equation (4.4) gives greater emphasis to larger deviations of the points from the line.

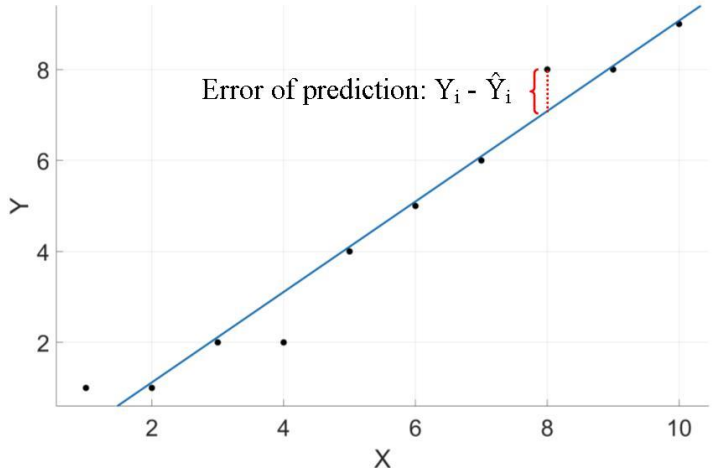


Figure 4.3. Visual straight-line fit and the variation.

$$SSE = \sum_{i=1}^{n}(Y_i - \widehat{Y_i})^2 \qquad (4.\ 4)$$

$$\widehat{Y_i} = \widehat{\beta_0} + \widehat{\beta_1}X_i \qquad (4.\ 5)$$

The “hat” is treated as “estimator of”. Thus, $\widehat{\beta_0}$ and $\widehat{\beta_1}$ are the numerical estimates of the parameters β_0 and β_1; $\widehat{Y_i}$ is the estimated mean value of Y. The deviation of Y_i from its predicted value is the ith residual. The least squares estimation method uses the criterion that the solution gives the smallest possible sum of the SSE.

Another indicator to measure the utility of a regression model is to quantify the contribution of the independent variable X in predicting the output Y. The indicator is the

coefficient of determination, denoted by R^2. It reflects how much the errors of prediction of Y_i can be reduced by using the information provided by X_i. If the value of Y_i doesn't correlate to the value of X_i, the best prediction of Y_i is the sample mean $\bar{Y}$. The sum of squares of deviation for the model $\hat{Y} = \bar{Y}$ is given as SS_{yy}:

$$SS_{yy} = \sum(Y_i - \bar{Y})^2 \qquad (4.\ 6)$$

If X_i contributes little or no information for the prediction of Y_i, SSE and SS_{yy} will be nearly equal. The SSE will decrease with more information provided by Xi, until the best case when SSE = 0. The coefficient of determination (R^2) is the quantity represents the reduction in the SS_{yy} that can be attributed to Xi, expressed as a proportion of SS_{yy}. R^2 is always between 0 and 1 as SSE $\leq SS_{yy}$. An R^2 value closer to 1 means a greater proportion of variance is accounted for by the model.

$$R^2 = \frac{SS_{yy} - SSE}{SS_{yy}} \qquad (4.\ 7)$$

As the functional relationship between the capacitive responses and the predicted vertical distance in this project is highly non-linear, more complicated regression models such as high-order polynomial model and exponential model will be applied and compared with experimental data. These two methodologies can be treated as extensions of the simple linear model, and the indicators being used to evaluate the performance of the models are SSE and R^2.

4.3. Profile recognition

The regression methodologies as explained in last section are able to build quantitative equations based on the numerical experimental capacitive responses. The regression model could be applied to the practically obtained capacitances, and consequently makes a prediction on the distance information. It provides an efficient way to acquire the desired distance information accurately and continuously. However, the ambiguity of the sensor readings, which is a common problem for capacitive sensing, makes it difficult to infer the distance without knowing additional object characteristics. Considering the generated electric field, a small object that is close to the sensor might have the same capacitive reading as a larger object at an increased distance. It has been verified during the simulation stage that even if placed at the same distance, objects with different surface profiles could result in quite different capacitive responses. Typically, the

actual shapes that might be encountered in a manufacturing process can be approximated by simple geometric shapes that are easier to process, e.g. plate, sphere, and cylinder. If the approaching object can be pre-classified into one of the shape categories, the most suitable regression model can be selected so that more accurate distance values can be derived. The task of profile recognition is qualitative: the variables being predicted are discrete rather than continuous. Therefore, the adaptive classification tools are involved to solve the problem.

Interpreting the experiments taken by the proposed sensing system requires the analysis of complex, multivariate and multi-dimensional data. In recent years, one analysis approach that has huge growth in popularity is the use of machine learning algorithms to train classifiers to decode behaviors and information of interest from the experimental data [151]. Machine Learning (ML) was firstly introduced in the late 1950's as a technique for artificial intelligence (AI) [152]. Over time, it has been used extensively for a wide range of tasks including classification in various application areas.

In this project, the promise of machine learning lies in its ability to exploit historical data to make an accurate prediction without the need of re-programming. More specifically, machine learning is essential in the application of our sensor for the following reasons: The sensor is usually deployed in a complicated and dynamic environment so that it is impossible to build an accurate mathematical model to describe the system behavior. A large amount of experimental data can be accessed but extracting important correlations within them can be difficult.

Existing machine learning algorithms can be categorized by the intended structure of the model, and most of them fall into one of the categories of supervised, unsupervised and reinforcement learning [153]. Supervised learning is the most studied and utilized type of learning: a labeled training set (i.e. predefined inputs and known outputs) is used to build the model. This model represents the learned relation between the input, output, and the system parameters. In the unsupervised setting, the training data does not contain any output information at all. Unsupervised learning can be viewed as the task of spontaneously finding patterns and structures in input data. The method being used is to classify the sample set into different groups by investigating the similarity between them. When the training data does not explicitly contain the correct output for each input, it falls into the category of reinforcement learning. The classifier will learn by interacting with its

environment and learn to take the best actions that maximize the long-term rewards of its own experience.

In this project, all the training data is acquired by performing experiments with the sensor. Thus, input data (capacitive responses) together with its target (shape classes) are both known. Therefore, in this project, supervised learning algorithms including k Nearest Neighbor (KNN), Neural Networks (NNs), and Support Vector Machines (SVMs) will be investigated and applied.

4.3.1. K nearest neighbors (KNN)

The k-nearest neighbors (KNN) algorithm is one of the most basic supervised learning techniques. KNN problems can be defined as: given a collection of data points and a query point in an m-dimensional metric space, find the data point that is closest to the query point [154]. The KNN algorithm predicts the outcome of a new observation by comparing it to k similar cases in the training data set, where k is defined by the analyst.

The KNN approach "looks at" the training set X, estimates the conditional distribution of Y given X, and then classify an observation with highest estimated probability [155]. Given a positive integer k and an experimental observation x0, the KNN classifier identifies the k points in the training set that are nearest to x0, counts how many members of each class are included in the set, and returns the empirical fraction as the classifying result [156]. More formally:

$$P(Y = j|X = x_0) = \frac{1}{k}\sum_{i\in N_0} I(y_i = j) \qquad (4.\ 8)$$

$$I(e) = \begin{cases} 1 & if\ e\ is\ true \\ 0 & if\ e\ is\ false \end{cases} \qquad (4.\ 9)$$

Where N0 is the collection of the k closest points to x_0, and *I(e)* is the indicate function as defined in Equation (4.9).

Figure 4.4 provides a demonstration of how does the KNN algorithm work. The left-hand panel plots a two-class training dataset consists of a few observations that are denoted by blue and red dots. The point labeled by a black cross is the new observation needs to be classified. Suppose the value of k is 3. The algorithm will firstly identify three nearest data points to the black cross, as shown in the green circle. Two blue dots and

one red dots are included, resulting in the estimated probabilities of 66.7% for the blue class and 33.3% for the red class. Therefore, the KNN classifier gives out its prediction that the new observation belongs to the blue class. Furthermore, by applying the KNN approach to all the training data points for both classes, the corresponding KNN decision boundary can be drawn as shown in Figure 4.4 (b). Despite the fact that it is a simple approach, KNN can often produce a quite optimal classification result, and it is treated as a suitable distributed learning algorithm for wireless sensor networks [157].

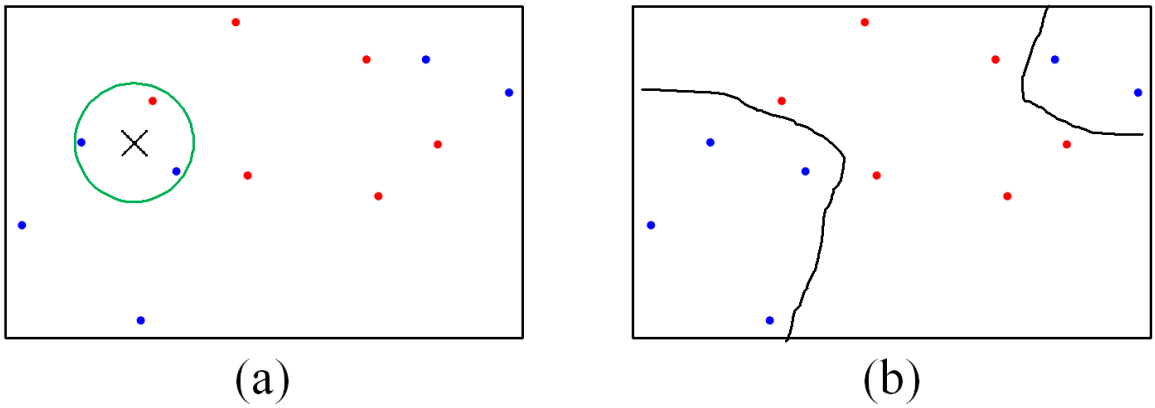


Figure 4.4. The KNN approach, using k = 3. (a) A test observation that is labeled with the black cross belongs to the most commonly occurring class: the blue class. (b) The KNN decision boundary for this specific data set is drawn with the black line.

The choice of “k” has a drastic effect on the performance of the KNN classifier obtained. Figure 4.5 displays two KNN fits two sets of random data with different offset values using the values for k as 1 and 25 respectively. When k =1, the decision boundary is overly flexible and induces a Voronoi tessellation of the points. This corresponds to an overfit classifier that has low bias but very high variance. With the growth in the value of k, the algorithm tends to be less flexible and the generated boundary becomes closer to linear. This corresponds to a low-variance but high-bias classifier. As any regression setting, there does not exist a strong relationship between the training error rate and the testing error rate. Though the training error rate with k = 1 is 0, the testing error rate might be very high. In general, the usage of a more flexible classification methodology, the training error rate will decline but the testing error rate not necessarily decreases. The testing error rate exhibits a characteristic U-shape, decreasing at the beginning before raising again when the algorithm is getting excessively flexible and overly fit. In this example, the first approach in which k = 1 would not be able to give good predictions: it will result in higher testing error rates due to the overfitting. Choosing the proper level of

the flexibility is critical to the success of any statistical learning method. The best tradeoffs between bias and variance should be taken good care of in order to obtain the optimal classification result.

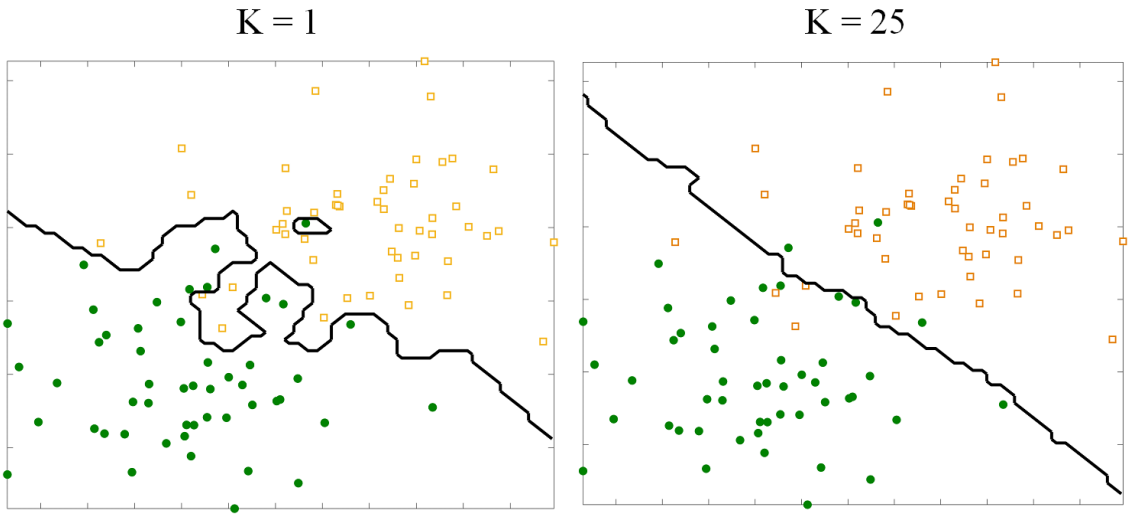


Figure 4.5. A comparison of the KNN decision boundaries (solid black lines) obtained with different values of "k". With k = 1, the decision boundary is overly flexible; whereas with k = 25, the generated boundary is closer to linear.

The KNN classifier is simple and works quite well, provided it is given a good distance metric and enough labeled training data [158]. However, the main problem with KNN classifier is that it might not work well with high dimensional settings. This poor performance when dealing with high dimensional data is due to the curse of dimensionality: the method is no longer local. The trouble that the neighbors that are generally far away make KNN not a good predictor of the behavior of the input-output function at a given point.

4.3.2. Neural Networks (NN)

The term "neural work" has its origins in attempts to find mathematic representations of information processing in biological systems [159]. Inspired by biological neural networks, an artificial neural network, also referred to as "neural networks (NN)" has been applied broadly as effective models for statistical pattern recognition [160].

A neural network algorithm derives its computing power through its massive parallel distributed structure and the ability to learn and generalize. Generalization refers

to the production of reasonable outputs for inputs that are not encountered during the learning procedure. Neural networks offer the following useful properties: A neural network, made up of an interconnection of nonlinear neurons, is itself nonlinear. Nonlinearity is highly important as most physical mechanism responsible for the practical input is inherently nonlinear. It is a nonparametric approach, an NN classifier estimates arbitrary decision boundary in the training data space by constructing an input-output mapping without invoking a probabilistic distribution model [161]. It is adaptive: NN has a built-in capability to adapt its synaptic weights to change according to the operating environment conditions. In the context of pattern classification, an NN provides information not only about which class an observation belongs to, but also about the confidence in the decision made. The massively parallel nature of the NNs makes them potentially fast in terms of computation, and thus well suited for implementation using very large scale integrated (VLSI) technology. These capabilities make it possible for NNs to find good approximate solutions to complex problems including wireless sensor networks [162], nonlinear hydrological processes [163], power electronics [164], and much more.

In this project, the attention is restricted to one specific type of NNs that is proven to have greatest practical value, namely the multilayer perceptron [165]. The basic features of multilayer perceptron can be highlighted with the following three points. The model of each neuron in the network involves a nonlinear activation function that is differentiable. The network contains one or more hidden layers from the input nodes to the output nodes. The network exhibits a high degree of connectivity and the extent of which is determined by its synaptic weight.

Figure 4.6 is the architectural graph of a multilayer perceptron with two hidden layers. To describe the multilayer perceptron in a more general form, the network shown in the figure is fully connected, that means any neuron located in any layer of the network is connected to all the nodes in the previous layer. The signal flow through the network progresses in a forward direction, from left to right on a layer-by-layer basis. In this network, there are two types of signals: function signals and error signals. A function signal is an input signal comes in at the input end of the network. It propagates forward through the network and finally emerges as an output signal. At each neuron of the network through which a function signal passes, the signal is calculated as a function of the input and the associated weights applied to that neuron. Thus, a useful function is performed at the output of the network. On the contrary, the error signal originates at the output neuron

of the network and propagates backward layer by layer throughout the network. The computation of an error signal by every neuron involves an error-dependent function. So each neuron performs two sorts of computations: the computation of the function signal appearing at the output side of the neuron and the computation of an estimate of the gradient vector. The hidden neurons act as feature detectors that are critical in the operation of the multilayer perceptron. They perform a nonlinear transformation on the input data into a new space called feature space, in which the classes of interest for a pattern-classification task might be more easily separated from each other.

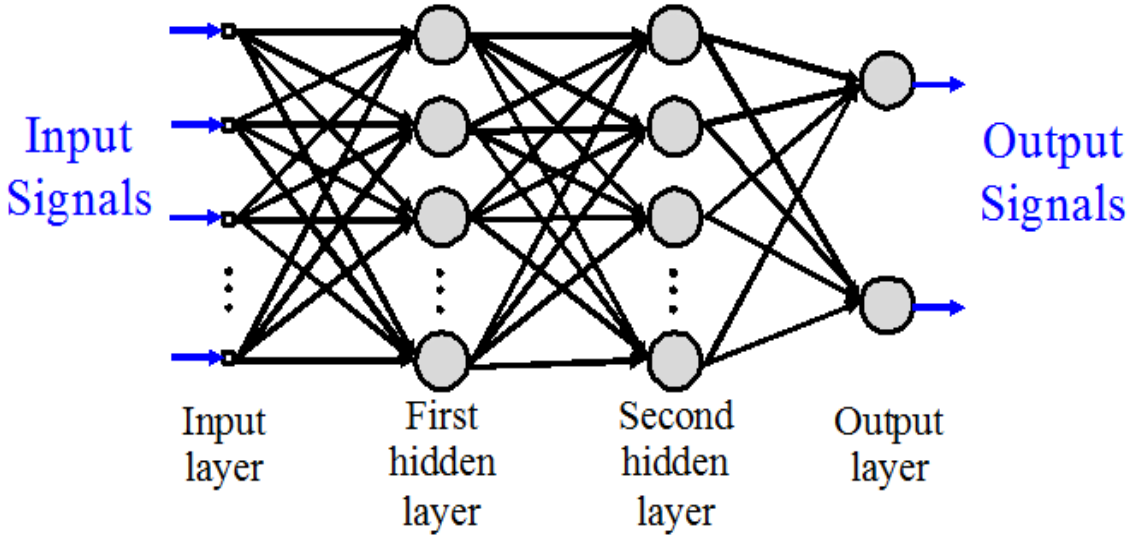


Figure 4.6. Architectural graph of a multilayer perceptron with two hidden layers

Training a multilayer perception is a procedure by which the values for the individual synaptic weights are determined so that the relationship between an output and an input is accurately resolved. Many training methodologies are available in training the multilayer perceptron, among them, the back-propagation algorithm is a popular one. The term “back-propagation” was popularized in the late 1980s through the publication [166]. The training process can be divided into two phases: the forward phase and the backward phase. In the forward phase, the synaptic weights are calculated and the input signal is propagated through the network until it reaches the output. In this way, the changes are confined to the activation potentials and outputs of the neurons in the network. In the backward phase, the error signal is generated by comparing the output with the desired response. The resulting error signal is propagated through the network in the backward direction from the output to the input. During this process, successive adjustments are made to optimize the synaptic weight. The development of the back-propagation algorithm

represents a landmark in neural networks as it provides a computationally efficient method for the training of a multilayer perceptron.

The implementation of the back-propagation algorithm is known as online training where the weights are adapted after each pattern has been presented. One alternative is batch training, where the weights are updated according to the summed error of all the patterns. In practice, there are much more training iterations that can be acquired.

Though the multilayer perceptron illustrates a lot of benefits and is broadly used, one reason that prevents it from being applied in practice is it is difficult to implement and interpret. The initial problem being faced when using a multilayer perceptron is deciding the network structure: the number of layers and the number of nodes located in each layer. Technically, only one hidden layer is required if the function between the input and the output is smooth [167]. However, the optimum number of the hidden layers and the neurons is highly problem dependent, and there are no rules to help. Another factor that needs sufficient attention is given too many hidden layers and nodes, it is highly probable that the perceptron will eventually learn all the training patterns of overfitting. This problem becomes severer with noisy training date. When such an overtrained network is presented with new patterns, high error rates will show up. It is also to remember that the purpose of the training process is to achieve the best generalization on unseen data. The training procedure should be stopped when the performance of the multilayer perceptron on an independent test data reaches a maximum, which is not necessarily when the network error is minimized. Maximum generalization performance always occurs before the overall network training error reaches a minimum.

4.3.3. Support vector machine (SVM)

Another machine learning algorithm that became popular within recent a few decades for solving problems in classification, regression, and novelty detection is Support Vector Machine (SVM). It is originally developed and refined by Vapnik and his co-workers in 1990s [168]. In the context of pattern classification, the SVM constructs a hyperplane as the decision surface based on the training samples, in such a way that the margin of separation between examples from different classes is maximized [161]. This basic definition can be extended in a principled way to deal with more difficult cases with non-linear separable patterns.

Initially, the SVM is a binary learning machine with some highly elegant properties. SVM is a kernel-based [169] algorithm that has sparse solutions so that the prediction for new inputs depend only on the kernel function evaluated at a subset of the training data points. So that the learning algorithm involved in the construction of support vector machine is also referred to as a kernel method. The SVM can be used to solve both classification and regression problems, however, it is proven to be able to make its most significant impact in solving difficult pattern-regression problems such as human action recognition [170], neuro-image analysis [171] and much more.

The aim of support vector classification is to devise a computationally efficient way of obtaining optimal separating hyperplanes in a high dimensional feature space. Different generalization bounds exist, motivating different algorithms: optimization of the maximal margin, the margin distribution, the number of support vectors, etc [172]. The most common and well-established approach is the maximal margin classifier. It works only for data which are linearly separable in the feature space, and hence cannot be used in real-world situations. Nonetheless, it forms the main building block for SVM.

The maximal margin classifier optimizes the bound by separating the data with the maximal margin hyperplane, and the geometric construction of an optimal hyperplane for a two-dimensional input space is visualized in Figure 4.7. The training samples can be expressed as (x_i, d_i), where x_i is an input sample and d_i is the corresponding output (i. e. either +1 or -1). The equation of a decision surface in the form of a hyperplane that does the separation is:

$$w^T x + b = 0 \qquad (4.\ 10)$$

where x is the input vector, w is the adjustable weight vector, and b is a bias. For a given pair of weight vector and bias, the separation between the hyperplane and the closest data point is defined as “margin of separation”, denoted by ρ. The goal of an SVM is to find the hyperplane for which the margin of separation is maximized. In this case, the decision surface is called the “optimal hyperplane”, represented by the area in between the two dashed lines in Figure 4.7.

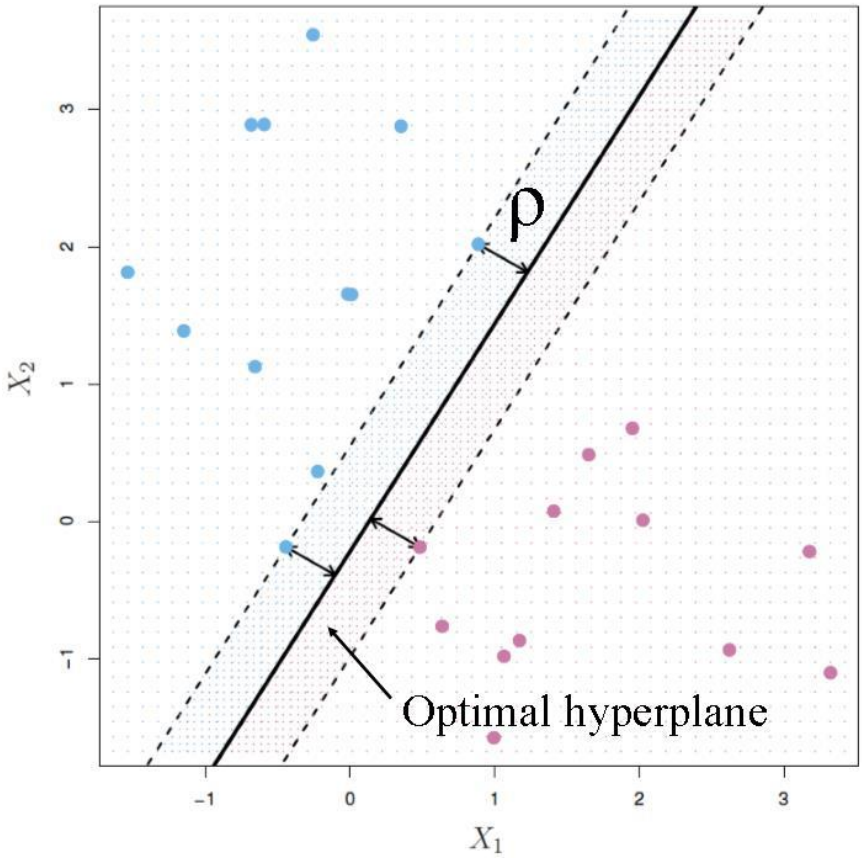


Figure 4.7. Illustration of the optimal hyperplane for linearly separable patterns

To extend this idea to the multidimensional problems, suppose w_0 and b_0 represent the optimum values of weight vector and bias respectively. Consequently, the optimal hyperplane, now representing a multidimensional linear separation surface can be defined with:

$$w_0^T x + b_0 = 0 \qquad (4.\ 11)$$

The discriminant function as expressed in Equation (4.12) provides an algebraic measure of the geometric distance from any x to the optimal hyperplane [173]. Then, the input x can be described with Equation (4.13), where x_p is the normal projection of x onto the optimal hyperplane and r is the desired algebraic distance. R is positive when x_i belongs to the class of +1, and negative when x_i is on the negative side.

$$g(x) = w_0^T x + b_0 \qquad (4.\ 12)$$

$$x = x_p + r\frac{w_0}{||w_0||} \qquad (4.\ 13)$$

Therefore, by definition, $g(x_p)$ should be 0, or, equivalently,

$$r = \frac{g(x)}{||w_0||} \qquad (4.\ 14)$$

Given the training data set (x, d), the parameter pair of (w_0, b_0) must satisfy the following constraints:

$$w_0^T x_i + b_0 \geq 0 \quad for\ d_i = +1 \qquad (4.\ 15)$$

$$w_0^T x_i + b_0 \leq 0 \quad for\ d_i = -1 \qquad (4.\ 16)$$

The particular training data points for which one of the constraints are satisfied with the equality sign are the “support vectors”. All the remaining samples in the training data set are completely irrelevant. The conceptual understanding of support vectors is those data points that lie closest to the optimal hyperplane and therefore the most difficult to classify. From the equations, maximizing the margin of separation between classes is equivalent to minimizing the Euclidean norm of the weight vector w. This highly distinct property makes the SVM a prominent tool in pattern recognition applications. The optimal hyperplane is unique in the sense that the weight vector provides the maximum possible separation between the data points from different classes.

SVM originally separates the binary classes with the maximized margin criterion. However, the real-world problems including the problems need to be solved in this study require the discrimination for more than two categories. So extending SVM so that it can be suitable for more general cases where an arbitrary number of classes is important. Among all the proposals for modifying the SVM to the K-class case, the two most popular approaches are the “one versus one” and “one versus rest” [174].

The one versus one (1V1) approach is a pairwise decomposition. It evaluates all possible pairwise classes and therefore constructs K(K-1)/2 individual binary classifiers. Applying each classifier to a test sample will generate one vote to the winning class. Finally, the test example will be classified to the class with more votes. Though the size of classifiers generated by 1V1 is large, it is still possible to achieve a fast train process due to the computation cost for each classifier is small. Moreover, the one versus one method is symmetric.

The one versus rest (1VR) approach only constructs K separate binary classifiers for a K-class problem. Each time, comparing one class to all the remaining classes as a whole. The i-th class is trained using the data from the i-th class as positive examples and

all the other data as negative samples, and the resulting parameters can be demoted with β_i. During the testing procedure, a test observation X is classified to the class for which $\beta_i X$ is the largest. As this amounts to a high level of confidence that the test sample belongs to the i-th class rather than any other classes.

There is no clear evidence that one versus one method can achieve higher accuracy compared to alternative multi-class SVM methods, but Hsu and Lin claim in [175] that 1V1 is more practical due to its quicker training process.

4.4. Cross validation

As mentioned a couple of times in the preceding sections of this chapter, a minimum training error rate does not necessarily result in an optimal testing error rate. The test error is the average error that results from using a learning method to predict the response on new observations: the data that is not used in the training procedure. The training error can be easily calculated with the pre-known training samples, however, in contrast, the real-world testing error can be difficult to evaluate and be quite different from training error. In particular, the training error rate can dramatically underestimate the testing error rate [147]. A common mistake when applying the machine learning tool is about overfitting. Overfitting is the phenomenon where fitting the training data point overly well so that the training performance no longer indicates a decent testing performance, and might actually lead to the opposite direction [153].

A very general and most widely used strategy that is able to provide a good estimation of the out-of-sample error rate and avoid overfitting problems at the same time is cross-validation (CV). The main idea behind CV is to split the data, one or more times, for optimizing the learning algorithm [176], each time by holding out a subset of the training examples and using the remaining samples to train the algorithm. After the training, the algorithm is applied to the held-out observations for estimating the testing error of the algorithm. These steps can be repeated for several times, and finally, the learning method that shows the best-estimated error will be selected.

A general description of the CV methodology is given in [177]: CV generates several hold-out estimators of the testing error corresponding to different data splits. While one of these non-empty subsets is used for testing, all the rest examples are used for

training the model. So that the training data and the estimators are totally independent of each other. There are several CV approaches, with the k-Fold cross-validation being an efficient and popular one.

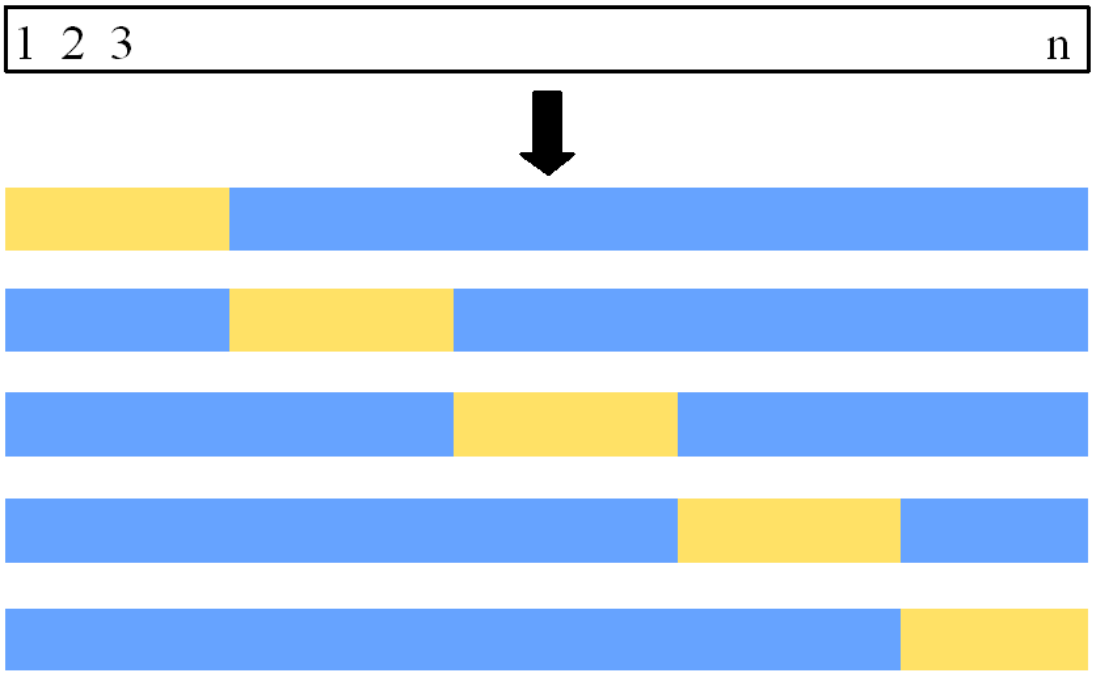


Figure 4.8. A schematic explanation of a 5-fold CV. The set of observations from 1 to n is randomly split into five non-overlapping subsets as denoted by the yellow bars. Each of them acts as a validation set once when all the remand data points denoted by the blue bar) acting as the training samples.

Figure 4.8 provides a schematic display of a 5-fold CV. The k-folds approach involves randomly diving the training observations into k groups (i.e. k folds), of approximately equal size. The method begins by treating the first subset as the validation set, and the learning algorithm is fit on the remaining k-1 folds. The first mean squared error, MSE_1, is computed on the samples in the first held-out fold. This procedure is repeated for k times, and each time, a different group of the observations is treated as the validation set. This process results in k estimates of the testing error, and the overall k-fold CV estimate can be calculated by averaging all these values:

$$CV(k) = \frac{1}{k}\sum_{i=1}^{k} MSE_i \qquad (4.\ 17)$$

In practice, the rule of thumb is setting k = 5 or k = 10. The most obvious advantage of choosing such values is computation efficiency. There also can be other non-computational but potentially more important advantages to performing 5-fold or 10-fold

CV, that involve the bias-variance trade-off. Performing a k-fold CV results in each training dataset contain (k-1)n/k observations. With the increasing of the value of k, the bias of estimates is getting smaller until an approximately unbiased estimate of the testing error since each training set contains almost all the samples. In addition to the bias, another source for concern in a validating procedure comes from the variance. In contrast, the variance will increase with a larger value of k. When performing k-fold CV, the estimated error is the average of the output of k fitted models. The outputs are less correlated when the overlap between the training sets is smaller, and the mean of highly correlated quantities has higher variance.

In summary, the choice of the value for k is associated with the computational efficiency and the bias-variance trade-off. Practically, given these considerations, a 5-fold or 10-fold cross validation shows empirically to yield testing error rate estimates not suffering from high bias nor high variance problems.

Chapter 5.

Evaluation results

In the previous chapters, the detailed theoretical analysis of the fringing electric field sensor was explained. Finite element methods (FEM) simulations were performed to model the most suitable array structure for the capacitive sensor. The extended characterization of operation and the corresponding statistical learning methods were described.

In this chapter, the hardware implementation of the sensing system with discrete components will be described in the beginning. In order to investigate the capabilities of the sensor and to observe its behavior properties in real application scenarios, a wide range of experiments are conducted. So that the performance of the sensing modalities can be experimentally assessed. Quantitative and qualitative statistical learning methods will be applied to the experimentally obtained capacitive responses to extract desired distance and surface profile information. The performance of different machine learning algorithms will be provided and compared in terms of classification accuracy and computation cost.

5.1. Hardware implementation

As described previously, the core component of the sensor is the sensing matrix that consists of a 4 × 4 electrode matrix, a 4 × 4 active shielding array, and a grounded backplane as shown in Figure 5.1 (a). The electrode matrix can be configured to form multiple mutual capacitors so that the approaching objects can be detected by the generated shunt electric field. The core structure of the sensor together with the complementary electronics are fabricated on a four-layer printed circuit board (PCB). The top three layers of the PCB serve as carriers for sensing matrix, active shielding matrix, and backplane shielding. Two analog switch matrix chips (AD75019 from Analog Devices) and required supplementary electronic components including bypass capacitors and resistors are fabricated on the bottom side. The photographs from both the top and the bottom view of the manufactured PCB are shown in Figure 5.1 (b). The two switch matrix chips provide independent switching for every sensing and active shielding electrode, so

that each of them may serve as the transmitter, receiver, or ground electrode. The runtime matrix reconfiguration is implemented through the two switch chips with each containing 256 unbuffered analog switches in the form of 16×16 array. The chip can be digitally programmed so that any or all of the input terminals can connect to any or all of the output terminals [178].

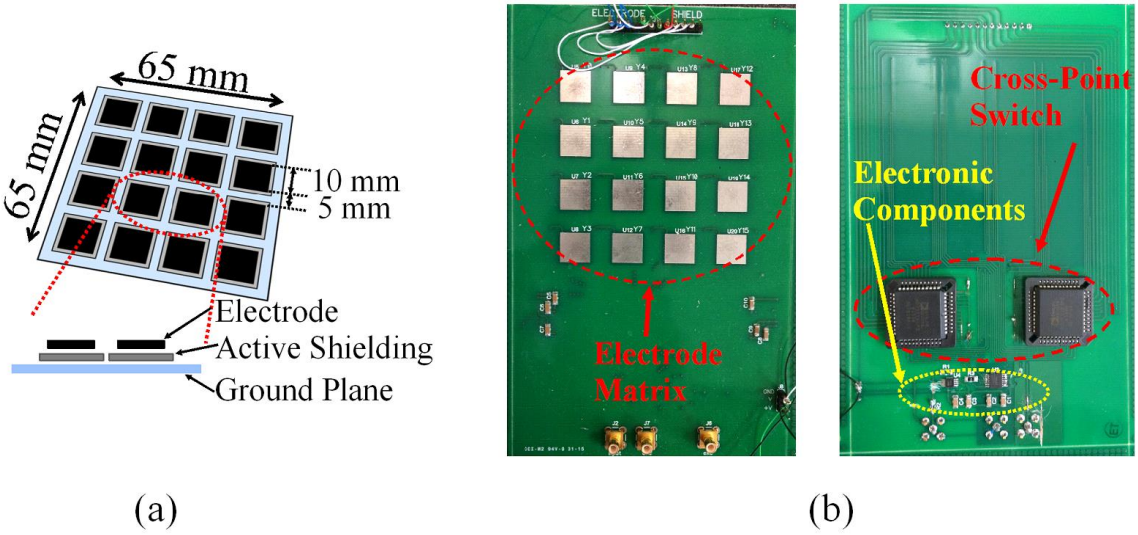


Figure 5.1. Hardware implementation of the sensor. (a) The mesh structure of the core sensor and magnified cross-section view close one electrode pair. (b) Photographs of the fabricated PCB from both top and bottom side.

In order to provide proper electrode connections for desired detection functionalities, the switch matrix chips (AD75019) need to be appropriately programmed. There are three digital input signals that need to be generated from an ad-hoc controlling digital logic: SIN, SCLK, and PCLK. In particular, the control instruction, which determines the configuration of all the 256 switches inside AD75019 chip, is clocked serially into a 256-bit shift register and then transferred in parallel to 256 memory bits driving the 256 analog switches: the instruction is loaded serially via the SIN pin. At the rising edge of the serial clock input SCLK, data is loaded into the shift register. After the shift register is filled with the new 256 bits of control data, the parallel clock input PCLK will be activated (pulse low) to transfer the data to the parallel memory driving the switches that will hold their data as long as power supply is applied.

To generate the control signals described above, a Field Programmable Gate Array (FPGA) board (Terasic DE1 board featuring ALTERA Cyclone II 2C20 FPGA device)

was added to the system. The FPGA was chosen as it allows a better control of the system timings as opposed to a microcontroller, its inherent parallelism matches well with the size of the control word, and its ready programmability ensures that future upgrades could be easily integrated into the system.

VHSIC Hardware Description Language (VHDL) is a programming language used for modeling the behavior of physical devices and processes [179]. It was developed to address a number of recurrent problems in the development, exchange, and documentation of digital hardware [180]. It is involved in our project as the programming language due to its public availability, technology and process independence, large-scale design, and design reusability. The goal of the VHDL code is to use the FPGA internal clock to generate proper control signals for the cross-point switches, in particular, the 256-bit control instruction word determining the sensing arrangement. Three input signals are applied to the FPGA, and three output signals are fed from the FPGA to the switches. The three inputs are clock (produced by the oscillator embedded in the FPGA), reset (which is active high), and serial data input. The outputs are, as described above, PCLK, SCLK, and SIN.

SCLK is the serial clock input of the cross-point switch chip, and according to the AD75019 specifications, its frequency range is between 20 KHz and 5 MHz. As the FPGA internal oscillator works at 50 MHz, a clock divider is built by the code. The 256-bit control word is exported serially through SIN port following the AD75019 specifications. The setup and hold time of SIN with respect to SCLK are taken into consideration by a dedicated finite state machine (FSM) that utilizes the 50 MHz clock as a discrete interval to implement the looser timings of the AD75019. PCLK is the parallel clock signal for the cross-point switch chip, it will only be active (pulse low) after the entire 256-bit instruction word has been loaded, and then come back to high within 5ms.

The generated digital signals usually suffer from glitches and unknown states during the transition, that may lead to some undesired values in the outputs. In order to get rid of these effects, three synchronized registers are added as the last stage of the signal path. Moreover, SIN_enable is used as an enable signal of a register R1, at each rising edge of SIN_enable, one single bit of the predefined data will be sent to SIN_internal signal. A set of predefined instruction words are configured into the FPGA to perform desired electrode connection patterns. There are two working modes: the user is enabled

to select any of the configurations by operating the switches on the FPGA board; all or a portion of the instruction words can be scanned automatically with a user-defined interval. The block diagram of the digital design is depicted in Figure 5.2.

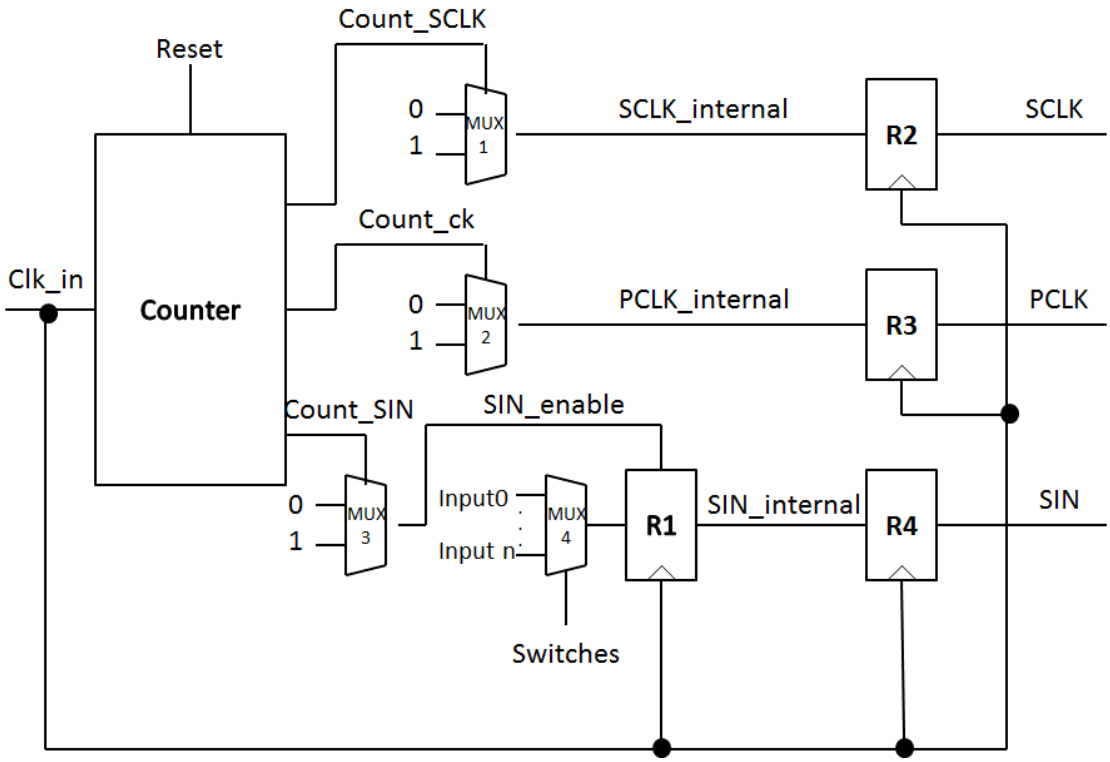


Figure 5.2. Block diagram of the designed digital controlling block.

The verification of the designed digital module is performed in three stages. The RTL VHDL described in the previous subsection is initially simulated utilizing Mentor ModelSIM. The VHDL code is then synthesized and mapped on the available FPGA chip and the resulting post-synthesis gate-level netlist is simulated again with Mentor ModelSIM, back-annotating timing delays extracted from the QuartusII timing analysis engine. This second step helps ensure that no glitch or delay would affect the expected behavior already verified at RTL level. Finally, the resulting bit-stream was loaded to the FPGA chip and the FPGA board was connected to the PCB featuring the sensors array as described previously. Measurements were performed using the different sensor arrangements specifies in the AD75019 control words proving the solidity of the whole system. When simulating the VHDL code both at RTL and gate-level, a testbench was prepared according to expected functionality. According to both RTL and Post-Synthesis back-annotated simulations, the proposed code achieved proper results.

In addition, timing and power analysis are also performed under the power supply of 1.25V at the temperature of 125°C. For the timing performance, the clock period is set as 20ns, while all the delay and transition times were set as half-period. Synthesis was carried out with the “compilation” function inside Quartus. In terms of power analysis, it is done by both vector-less and VCD-based methods. Value Change Dump (.VCD) file in which every signal and every change of state during the simulation is recorded for every wire in the design can be generated during the simulation process. This VCD file can be fed to Quartus power analyzer for a detailed and accurate power analysis called VCD-based analysis. The power analysis without applying this file is known as vector-less analysis. The results for timing and power analysis are shown in Table 5.1. Compared to a vector-less power analysis, static power and I/O power hold almost the same value, but the dynamic power decreases a little bit as vector-less analysis is based on statistical, pessimistic approximations.

Table 5.1. Analysis results of the digital block on Altera FPGA @1.25V_125°C

Timing	Setup time slacks	Hold time slacks	Power	Vector-less (mV)	VCD-based (mV)
Slow model	4.297	0.621	**Static**	47.36	47.36
Fast model	6.801	0.241	**Dynamic**	2.47	2.43
			I/O	20.45	20.24

To obtain the capacitive responses, in the experimental process, an AC signal is applied to the transmitter and the mutual capacitance is determined by measuring the displacement current on the receiver side [181]. AD7746 CDC chipset manufactured by Analog Devices is acquired to conduct a comprehensive study on capacitive sensing performance. The chip kit is used to achieve capacitance measurement, quantize the measured capacitances, and exhibit results in terms of both capacitance values and digital strings via its standard communication interface [182]. A square-wave excitation signal with the frequency of 32 kHz and voltage of 2.5 V is applied on the transmitter during the conversion and the charges going through the mutual capacitor is sampled. Its sampling rate is set as 90.9 Hz, which means the time consumed for one single measurement is 11 ms. Ultimately, measured data is stored and processed on an external computer.

In order to investigate the capabilities of the designed sensing system as a whole and to observe its behavior properties in real application scenarios, a variety of prototypical experiments are conducted. Figure 5.3 shows the experimental setup and the

apparatus established in this study: a height-adjustable frame is used to support different vertical positions, parallel movement, and change of objects with different shapes. The value *d*, which respects the vertical distance from the object to the sensor, is controlled by the positioning stage. Objects are held by a rotatable beam so that a parallel motion can be realized. The plate can be replaced by simply attaching alternative objects to the replaceable joint for surface profile classification. The three objects being used are a plate, a ball, and a cylinder. Different object materials would have different influences on the sensor's capacitive responses. However, the main working scenario of the designed sensor is detecting robot parts that are usually metallic in manufacturing processes, so the focus is put on investigating the effects caused by metallic objects. Moreover, the human body and tissues are mainly composed of water, which is conductive. So a metallic object can also be a good surrogate for a human body. Though studying the capacitive responses from different materials is not the purpose from the perspective of this work, some other group members did this research previously. All the three objects involved in the experiments are made of the aluminum. The area of the plate is 16 cm × 13 cm, the diameter of the ball is 6.5 cm, and the radius and the height of the cylinder are 1.5 cm and 10 cm respectively.

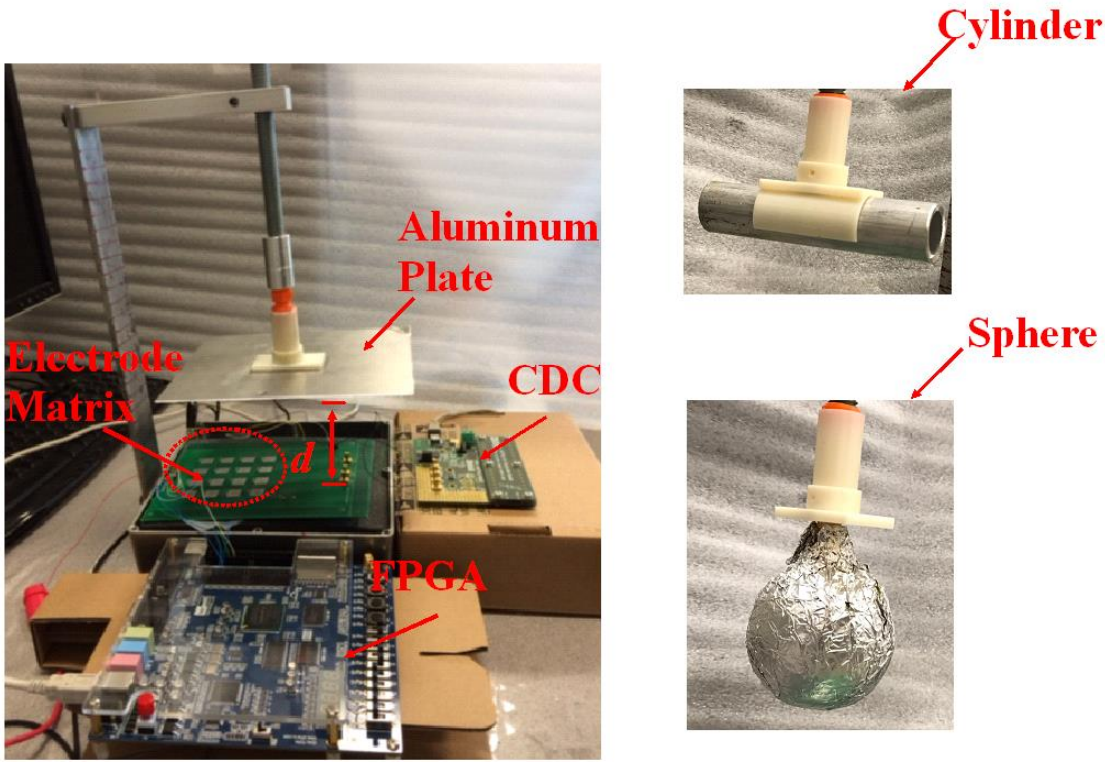


Figure 5.3. Experimental setup.

5.2. Signal pre-processing

One of the main drawbacks when applying the capacitive sensors is their detection range and resolution are sensitive to environmental interference. The capacitance will change with the temperature and the humidity because the dielectric constant depends on the temperature, and the moisture in the air leads to a larger dielectric constant [183]. In addition, random disturbance from various sources including human factors will cause unexpected influences to the capacitive responses. It is impossible to maintain the optimum working condition for the sensor.

The effects of the interferences should be explored and accounted for in order to minimize the uncertainty of the results. Signal pre-processing procedure that deals with the “dirty” raw data is primarily intended to compensate for these non-ideal effects. One common type of interference that is added to the signal is high-frequency noises generated from the environment. Another important aspect of raw data processing is about random-walk canceling. A possible cause of the drift is the impact of temperature variations on the capacitance [184]. In practical, it is essential to provide a well calibrated and adaptive baseline, which is defined as the capacitance acquired from the environment without the presence of an object [185].

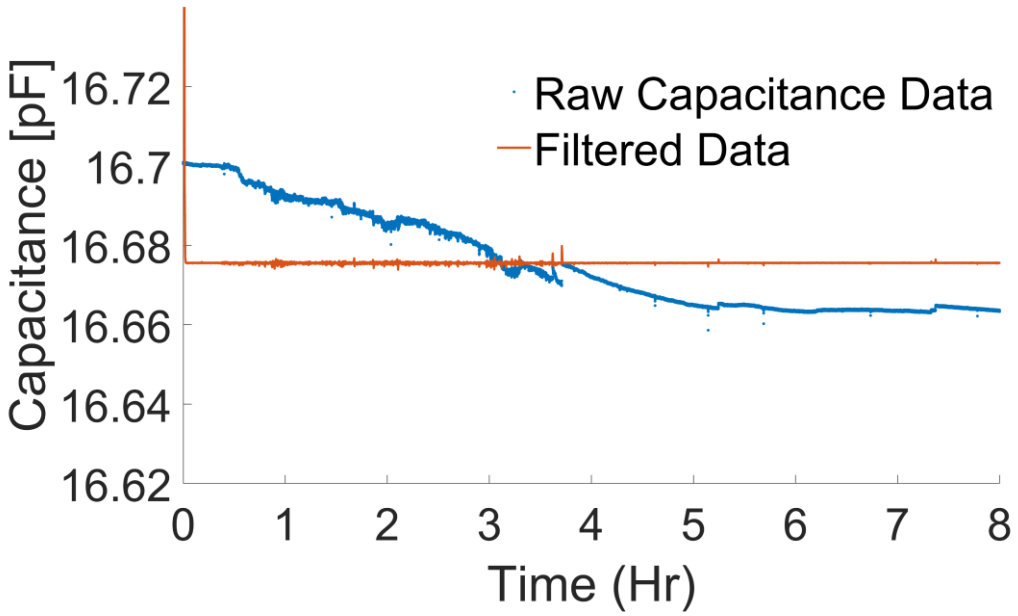


Figure 5.4. Comparison of raw capacitive data from long-term measurement and filtered result with both a dc notch and a low-pass filter.

Both high-frequency interference and random-walk should be handled in the raw-data processing stage. A long-term measurement was carried out to monitor the intrinsic capacitance: the sensor was kept working without any approaching object. The experiment lasted for a whole week. A portion of data that consists of measured capacitance of 8 hours from a typical working day have been captured and plotted in Figure 5.4 with blue dots. From which it can be found that the detection signal is drifting at a rather low frequency compared to the moving speed of an object whereas glitches that represent interference are at a much higher frequency. Therefore, a dc notch filter is used to cancel random-walk. Additionally, a low-pass averaging filter with the cut-off frequency of 20 Hz that takes various samples in the close neighborhood and calculates an average value will be employed in this work to deal with high-frequency noise issues. The data filters are applied using Matlab [186] and the filtered signals are plotted on the same graph in red. After proper filtering, the drift issue is fully eliminated and the waveform becomes much smoother.

5.3. Proximity sensing capability

The primary task of the designed capacitive sensor is to provide vertical distance measurements with large detection range and enhanced accuracy. The proximity sensing capability is measured with respect to the four electrode connection patterns (i.e. Type I, Type II, Type III, and Type IV as described in Chapter 3) and the three objects with different shapes (i.e. the plate, sphere, and the cylinder). The vertical distance range of interest is from 1 cm to 20 cm.

The experiment process is explained by taking the plate as the example. The implementation of basic distance estimation can be divided into three steps: Select one of the four electrode connection patterns by configuring the switches on the FPGA board. Then move the object continuously from 1cm to 20cm away from the sensor, and record the capacitance values at the same time. Lastly, feed the acquired original experimental capacitive responses and their corresponding vertical distance values to MATLAB curve fitting tool to generate the desired regression model. This procedure is repeated for every electrode connection types for the sake of comparison. The original experimental samples for Type I, II, III, and IV are plotted with black dots in Figure 5.5 (a), (b), (c), and (d) respectively. Fitted curves that describe the mathematical relationship between a capacitive value and its corresponding distance are denoted by the red lines. After trying

different regression methodologies including linear models, polynomial models, and exponential models, it can be figured out that the two-term exponential regression models are able to provide high-degree of fittings.

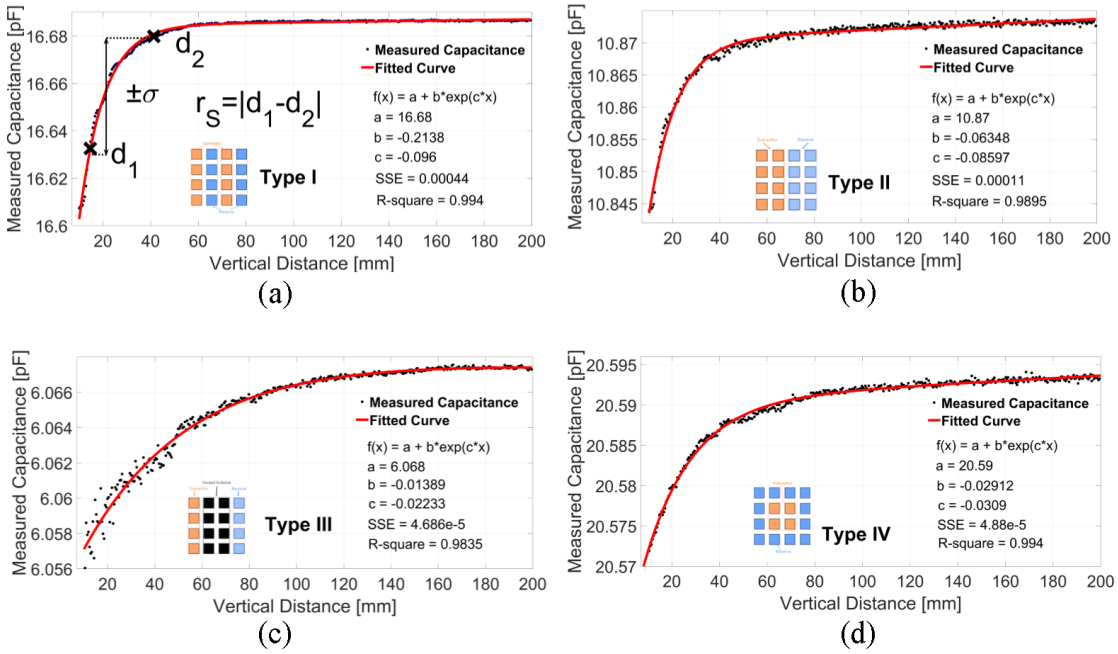


Figure 5.5. Use the plate as the object. Measured capacitive raw data (shown with the black dots) and the fitting curve (denoted by the red lines) with respect to four electrode connection types. (a) Type I. The approach to determine the spatial resolution at a given point is explained: it is based on a linearly interpolated measurement series and the standard deviation. (b) Type II. (c) Type III. (d) Type IV.

Regression model expressions together with corresponding coefficients and goodness indices (i.e. SSE and R-square) are also denoted in the figures. The sum of squared errors of prediction (SSE) measures the total deviation of the response values from fit values. An SSE value closer to 0 indicates the regression model has a smaller random error component and is more useful for prediction. R-square, also called the coefficient of determination, is defined as the square of the correlation between the response values and predicted values. It stands for how successful the fitting model is in terms of explaining the variation of the data, and a value closer to 1 means a greater proportion of variance is accounted for by the model. New data from real-world applications can be directly inserted into the desired model and the vertical distance can be predicted by calculation.

The measured initial capacitance (at the distance of 1cm) of Type I, II, III, and IV are 16.607 pF, 10.844 pF, 6.055 pF, and 20.572 pF, respectively. Their absolute capacitive values are quite different. It worth the effort to make a comparison between them in a normalized form as in practice, the capacitive changing ratio and the linear detection range are with more interest. This process is known as a data normalization which means adjusting the data to a notionally common scale. The applied method is linearly rescaling the range of data from each type to the range in [0, 1]. More specifically, given the maximum capacitive value C_{max} and the lower bound C_{min} for one data set, a normalized value is calculated by $C_N = (C-C_{min})/(C_{max}-C_{min})$ to keep the values in the [0, 1] range [132]. In addition, different connection types result in different levels of immunity to noises due to their different signal strengths. To enable a more precise comparison, two-term exponential regression model as a function of distance f(d) is created to generate fitting curve for each set of the normalized data. By this way, effects brought by high-frequency noises and outliners can be eliminated. However, the maximum and minimum values of f(d) are slightly off 0 and 1, so the last step is to shift and stretch the obtained curves to the range of [0,1]. The concept used here is also linear scaling: finding the minimum value f_{min} of the function that obtained at the minimum distance f(10) and the maximum output f_{max} that happens at the longest distance f(200). The normalized fitting function can be expressed as $f(d) = (f(d)-f_{min})/(f_{max}-f_{min})$. By plotting this normalized fitting curves for all the electrode connection types, an intuitionistic and precise comparison can be conducted.

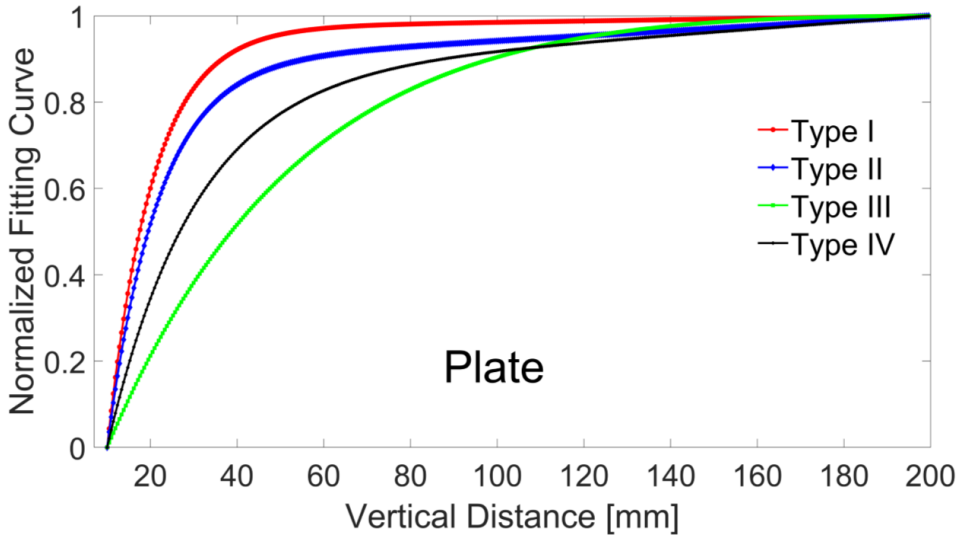


Figure 5.6. Use the plate as the object. Normalized fitting curve as a function of vertical distance for all the four electrode connection types.

Normalized capacitive fitting curves with respect to distance values for all the four types are represented by the lines in Figure 5.6, with red line represents Type I, blue line stands for Type II, green line describes Type III, and black line for Type IV. Though with the smallest absolute capacitance, Type III exhibits the longest linear detective range as expected whereas Type I tends to have largest capacitance change in short distance.

One most important aspect of this study is to offer the distance measurement with improved accuracy. In order to provide a quantitative analysis of the sensing accuracy, one more set of experiments are carried out and the principle to determine a system's spatial resolution introduced in [187] is applied. The spatial resolution for a given distance d can be used as an indicator: the sensor is able to detect the defined object at this specified distance with the precision of $r_s(d)$.

Table 5.2. Spatial resolution summary for the plate

d(mm)	$\sigma_n(d)$ (fF)				rs(d) (mm)			
	Type I	Type II	Type III	Type IV	Type I	Type II	Type III	Type IV
30	0.148	0.134	0.053	0.117	0.285	0.571	0.304	0.475
60	0.165	0.172	0.061	0.123	4.469	7.123	4.179	1.974
90	0.109	0.109	0.056	0.119	13.408	11.031	6.463	5.226
120	0.105	0.111	0.062	0.116	16.356	12.648	14.444	13.525
150	0.124	0.124	0.061	0.102	19.590	14.264	24.321	19.706
180	0.133	0.133	0.055	0.108	21.111	15.215	32.301	22.330

A group of discrete distance values in between 1cm and 20cm (i.e. 3cm, 6cm, 9cm, 12cm, 15cm, 18cm) has been chosen for estimation. For every single distance d, the plate is fixed for measurement, and a series of experimental capacitive samples are recorded for 10 seconds. This allows calculating basic statistical values of the sampled sensor measurements. The most important values are the arithmetic mean $S_n(d)$ and the standard deviation $\sigma_n(d)$, the latter being the main criterion to determine the sensor's signal-to-noise ratio. Figure 5.5 (a) outlines the way of obtaining the spatial resolution $r_s(d)$ for any given distance d by using Type I as an example. At a certain distance d, the standard deviation is used to look for distances d_1 and d_2 that deviate from the mean value with $\pm\sigma_n(d)$, respectively. This lead to an estimation of $r_s(d)$ which is calculated as the absolute difference between d_1 and d_2. The calculated spatial resolutions for some discrete distances are summarized in Table 5.2. The sensing precision is deteriorating with the

increasing distance, as the capacitive changes are becoming small and noises have a greater influence. Type IV provides a better performance in terms of resolution at short distances, whereas Type II shows better immunity to noises at longer distances.

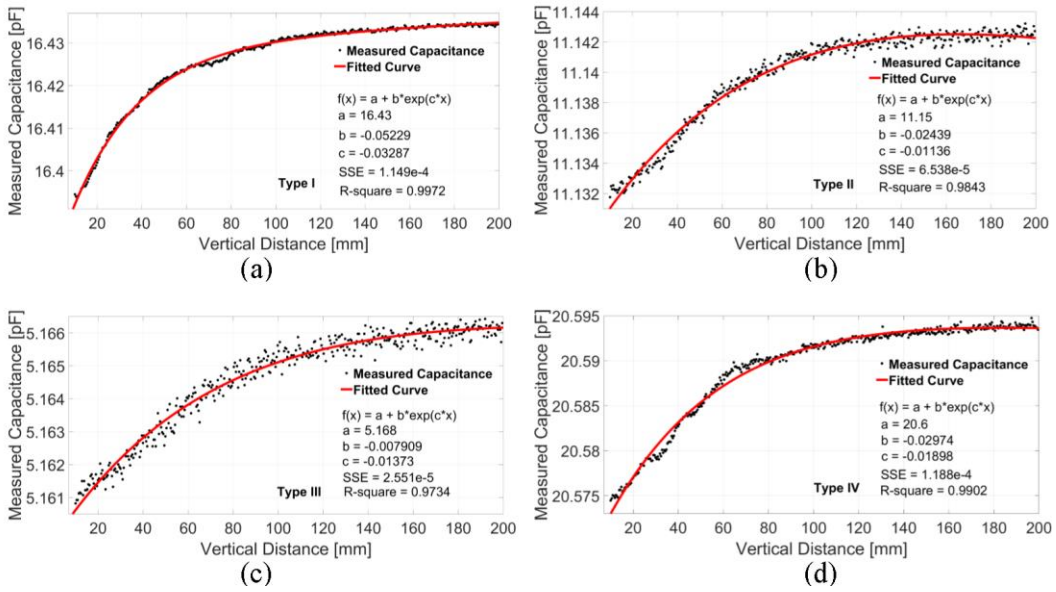


Figure 5.7. Use the sphere as the object. Measured capacitive raw data (shown with the black dots) and the fitting curve (denoted by the red lines) with respect to four electrode connection types. (a) Type I. (b) Type II. (c) Type III. (d) Type IV.

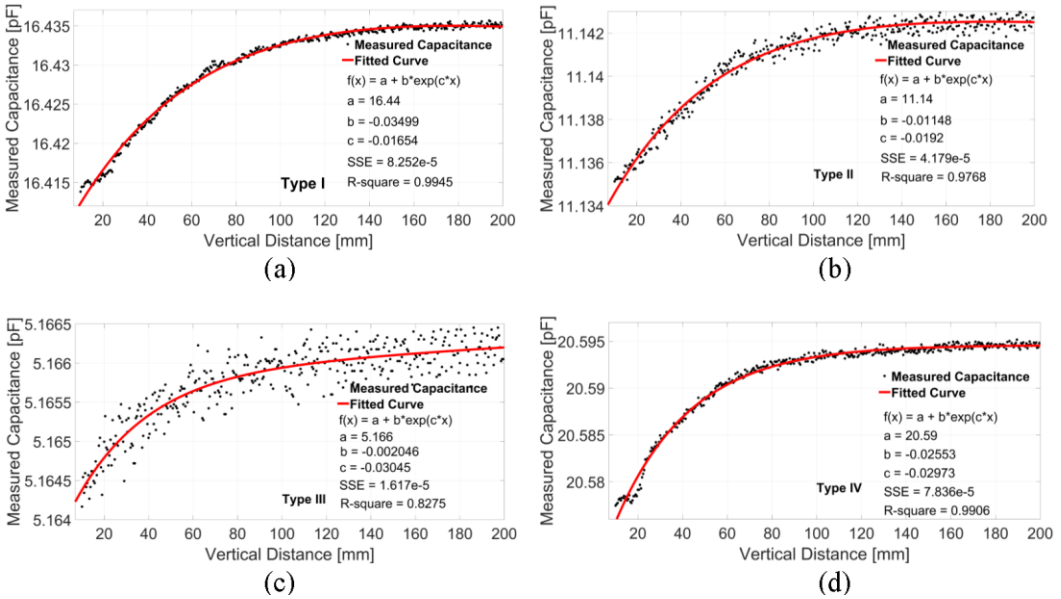


Figure 5.8. Use the sphere as the object. Measured capacitive raw data (shown with the black dots) and the fitting curve (denoted by the red lines) with respect to four electrode connection types. (a) Type I. (b) Type II. (c) Type III. (d) Type IV.

The previous two sets of experiments provide a comprehensive analysis of the sensor's proximity evaluation performance with respect to the electrode connection patterns. The same measurements are repeated with the other two objects for the study of the influences brought by different shapes. The experimental capacitive values together with the corresponding fitted curves are depicted in Figure 5.7 and Figure 5.8 for the sphere and the cylinder, respectively. In these two cases, the two-term exponential regression models also provide good fittings for the acquired data and are able to predict the distance information with future measured capacitance values.

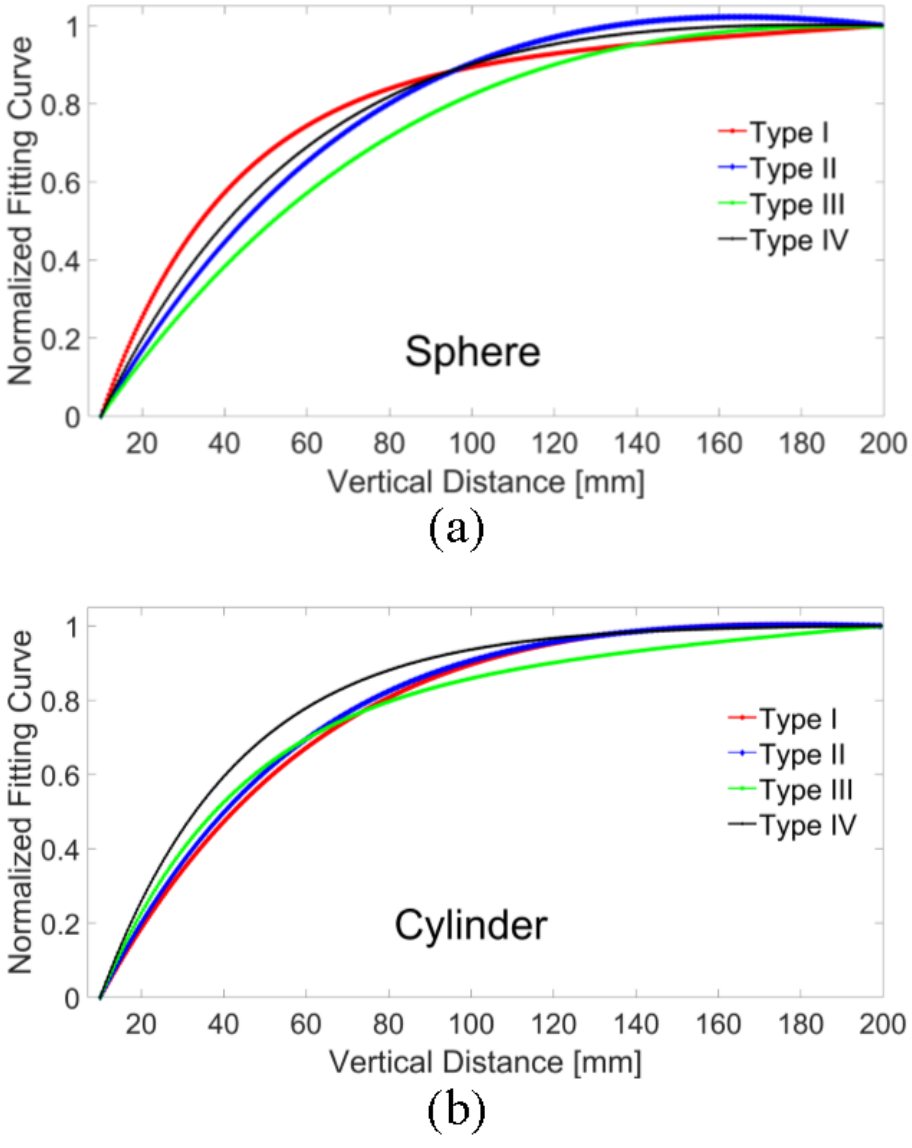


Figure 5.9. Normalized fitting curve as a function of the vertical distance. (a) Use the sphere as the object. (b) Use the cylinder as the object

The same as the plate, the absolute capacitive values and noise levels acquired from the four electrode connection types are quite different. To get a clearer understanding of the influences caused by different electrode connection patterns, the raw capacitance data was normalized for creating regression models. Figure 5.9 (a) denotes the normalized fitting curve as a function of vertical distance for the sphere with all the four electrode connection types. The same plot for the cylinder is shown in Figure 5.9(b). One common observation exits for all the three shapes is that though with the smallest absolute capacitance values, Type III which only involves the two columns located at the edge of the sensor brings out the longest detection range. For both the plate and the sphere, the Type I which stands for an interdigital connection pattern exhibits the largest capacitance change when the object is relatively close to the sensor. Whereas for the sphere, Type IV that represent the ring-shaped connection has the similar character in short distances.

Table 5.3. Spatial resolution summary for the sphere

d(mm)	σ_n(d) (fF)				rs(d) (mm)			
	Type I	Type II	Type III	Type IV	Type I	Type II	Type III	Type IV
30	0.183	0.163	0.129	0.142	1.425	0.95	0.475	0.539
60	0.156	0.173	0.192	0.153	1.425	3.334	2.154	2.375
90	0.168	0.160	0.216	0.121	5.701	3.890	7.136	6.179
120	0.183	0.207	0.117	0.110	9.987	12.963	15.675	9.5
150	0.165	0.196	0.167	0.132	14.725	9.037	19.011	12.351
180	0.173	0.222	0.148	0.158	16.825	19.938	23.125	21.842

Table 5.4. Spatial resolution summary for the cylinder

d(mm)	σ_n(d) (fF)				rs(d) (mm)			
	Type I	Type II	Type III	Type IV	Type I	Type II	Type III	Type IV
30	0.138	0.144	0.171	0.116	3.325	1.907	1.830	3.324
60	0.136	0.163	0.158	0.126	3.857	8.075	4.275	6.671
90	0.134	0.187	0.235	0.145	6.637	9.025	9.978	11.873
120	0.162	0.212	0.138	0.124	11.877	13.333	16.625	13.759
150	0.160	0.185	0.121	0.122	12.832	18.05	17.579	21.843
180	0.137	0.183	0.201	0.143	13.47	22.815	23.042	24.207

The above-mentioned spatial resolution calculation procedure is performed for the sphere and the cylinder to investigate the different performance achieved by the different

connection types in terms of resolutions. The calculated standard deviation $\sigma_n(d)$ and the estimated spatial resolution $r_s(d)$ are summarized in Table 5.3 and Table 5.4 respectively for the sphere and the cylinder.

In summary, for every object, different electrode connection patterns can provide quite different distance detecting efficiency in terms of linear detection range, capacitive changes, and the spatial resolution. The most suitable connection types can be selected depending on the primary requests in the most desired application scenarios. Overall, for a pre-defined object, the proposed sensor is able to detect any of the three objects at distances up to 18cm with relatively high resolutions.

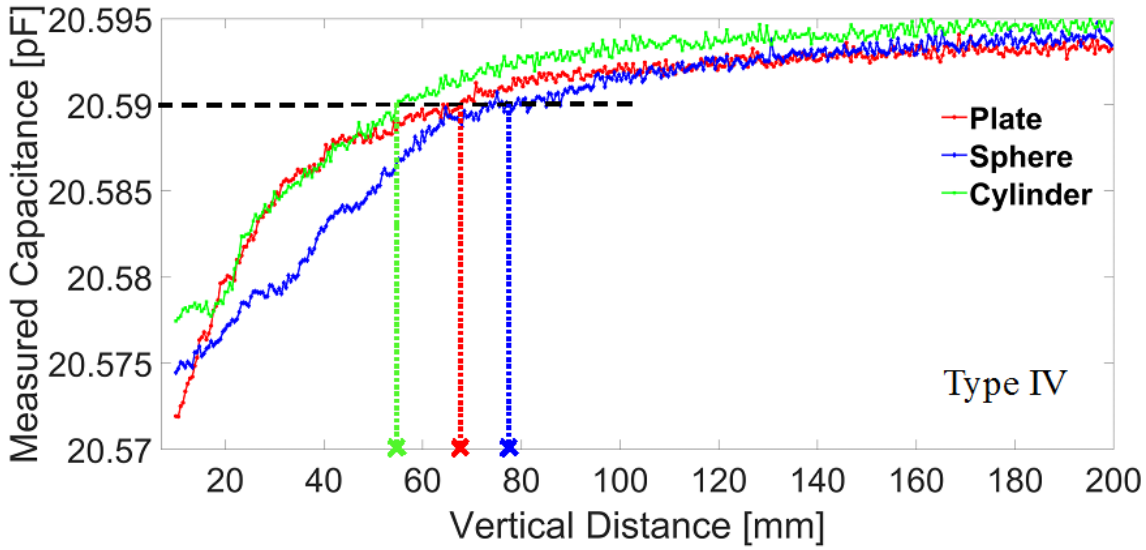


Figure 5.10. Measuring error caused by the confusion of the object's shape. Measurement is performed with the electrode connection Type IV.

Based on the obtained experimental results, the influences on the capacitive results caused by different object shapes can also be investigated. One common observation for all the three objects is the measured capacitance is rising with the increase of the object's vertical distance. However, when looking into the detailed numbers it becomes clear that their absolute capacitive values and the changing slopes are quite different. After comparing all the obtained data, it is found that the electrode connection Type IV demonstrates the least measurement difference among those three objects. For these three objects, the experimental capacitive response as a function of vertical distance using Type IV is summarized to the same plot for comparison as shown in Figure 5.10. Even though better than the other three electrode connection patterns, it still generates

high-degree of uncertainty to the relation between capacitance and distance. That will lead to a decreased measurement accuracy or even a wrong distance estimation. This fact motivates the requirement for surface profile recognition.

5.4. Profile recognition

Accomplished the comprehensive evaluations on the proximity sensing capability, it can be found that with the shape information pre-known, the sensor is able to detect a metallic object with high resolutions. However, the relation between the vertical distances and the capacitive responses is highly dependent on the shape of the approaching object. Therefore, the spatial resolution drops dramatically if the sensor is confused about the shape. To improve the distance detecting accuracy, it is essential to eliminate the ambiguity on the surface profile of an object before or at the same time of performing the distance measurement. In this section, dynamic measurements are carried out with the same sensing platform, and multiple statistical learning methods are applied to extract finer information from the experimental data.

We attempted two different paths to the ultimate goal of this study: estimate the distance of an approaching object with high accuracy. The first technique involves two stages: Firstly, recognize what is the most probable shape of the approaching object, then select the most suitable regression model accordingly and feed the measured capacitive values to it for distance calculation. The second technique sectionalizes the results with the criteria of both distance and surface profile information. In this method, three discrete vertical distances of 5cm, 10cm, and 15cm and three shapes including plate, sphere, and cylinder form the basis of the criteria. The target categories are defined by permutations between the distances and the shapes. The obtained capacitive responses are classified into one of the nine target classes, namely plate with the distances of 5cm, 10cm, 15cm, (P@5, P@10, P@15); sphere with the distances of 5cm, 10cm, 15cm, (S@5, S@10, S@15); cylinder with the distances of 5cm, 10cm, 15cm, (Cy@5, Cy@10, Cy@15). In this way, the consciousness of which object appears at what distance range can be accomplished in one step.

5.4.1. Experiment process

The basic experimental procedure for both the techniques are similar: The FPGA is programmed to run an automatic sweeping of the different electrode connections used for surface profile recognition (as described in Chapter 3, Section 3.2.3), and the capacitive responses are captured and recorded by the capacitance-to-digital converter through an external PC. The effectiveness of the generated data often relies on the well-designed behaviors of the experimental subjects. The "behaviors" under the context of this project represent a series of the object locations that will most likely add confusions to the learning tools but not contribute to the classification. The obtained data is then fused and labeled with the desired classes.

In the first technique, the primary task is to recognize the surface profile of the nearby object while any other non-geometric features including its vertical location and horizontal status are of less or none interest. During the testing, each of the three objects (i.e. the plate, the sphere, and the cylinder) is placed at 90 random locations within the detectable area. Every capacitive measurement will last for a short period of time. That allows the sensing system to have enough time for scanning all the internal electrode connection patterns as well as collecting multiple samples. In this process, the corresponding mutual capacitors are measured in turns resulting in an eight-dimensional or nine-dimensional feature vector (i.e. $\{C_1\text{-}C_8\}$ or $\{C_I\text{-}C_{IX}\}$). Every time, ten feature vectors are generated and collected for completing the input data sets. This process is repeated for every location and every object, resulting 2700 feature vectors that can be divided into three surface profile classes with each consist of 900 feature.

For the second technique, the goal is to reliably distinguish the specified vertical distance together with the object's shape. In this case, the implementation is location-awareness, but the horizontal status of the object is with less priority. A list of horizontal actions is executed by each object at every distance to enrich the dataset for each of the nine categories. The actions being chosen for each object are appropriate metaphors for potential situations that may bring confusions to the classification tools. The actions are also designed to make the interpretations of signals intuitively and as reproducible from time to time as possible. As a symmetrical shape, the sphere affects the sensing responds with its different horizontal locations. For every vertical distance, 30 different relative locations are investigated with the ball. Figure 5.11 (a) demonstrates five of the most

typical locations: above the center and the four corners of the sensor. On the contrary, for the plate, the most likely scenario that may cause confusions is tilting. Instead of only studying the flat position, 30 plate positions with different tilting angles are taken into consideration. Figure 5.11(b) denotes the tilting positions about one of the central axes of the plate, and the plate is also tilting along the other direction during the test. The in-plane rotations of the cylinder could disturb the classification, so a series of rotating positions as shown in Figure 5.11 (c) has been studied. The rotating interval is set as 6° so that 30 different in-plane locations can be obtained by the cylinder. Thus, each class that represents one distance/shape category contains 30 independent data sets.

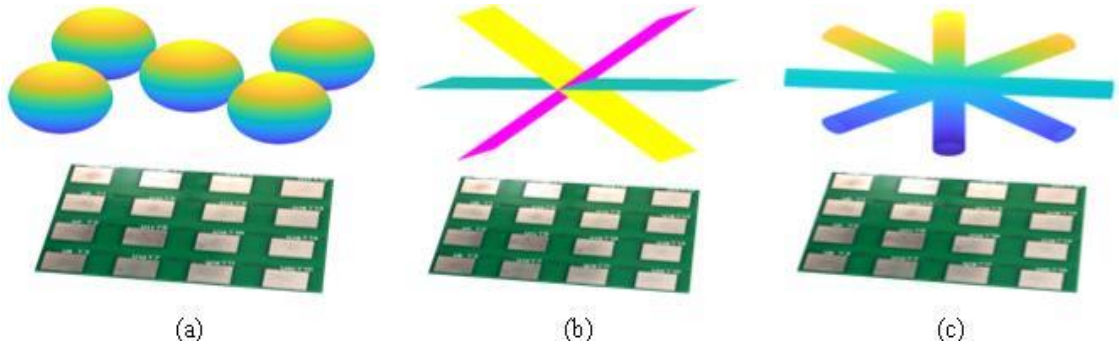


Figure 5.11. Horizontal actions performed by different objects at one certain distance. (a) Several different locations for the sphere. (b) The tilting positions about one central axis for the plate. There are more along the other direction. (c) Some in-plane rotations performed by the cylinder.

Data is collected by performing the capacitive measurements when maintaining the subject at one of the above-mentioned positions. Eight or nine mutual capacitors are measured in turns resulting in high-dimensional feature vectors. Repeat this process for 10 times, as a result, 300 feature vectors are generated for each class and in total 2700 input instances that can be acquired and labeled to nine classes.

5.4.2. Principal component analysis (PCA)

Principal component analysis (PCA) is a dimensional reduction and data compression tool. PCA can be used to reduce the large set of attributes to a smaller set but still contains most of the information. It is a mathematical procedure for identifying the patterns in data and expressing the data in a way that highlights the similarities and differences among variables. In this project, the number of attributes contained in each instance is either eight or nine, thus, compressing the dimension might not have a

significant impact on computational cost. However, PCA provides a good way to analysis the dependency behind the variables, and that will, in turn, help us understand the correlations among the data sets.

	C1	C2	C3	C4	C5	C6	C7	C8
0	8.002803	5.925395	8.086721	5.137771	6.159938	16.81726	15.56249	14.59074
1	7.867874	5.858182	7.984549	5.024867	6.023945	16.15277	14.98124	14.04098
2	8.017511	5.928306	8.090099	5.144789	6.175393	16.90427	15.62783	14.6597
3	8.005397	5.926776	8.085918	5.13219	6.152603	16.7815	15.53253	14.56395
4	7.868789	5.859769	7.983896	5.023598	6.021631	16.12722	14.96185	14.01767
5	8.030315	5.93426	8.09247	5.146525	6.172761	16.89688	15.62197	14.65192
6	7.996401	5.922987	8.082727	5.126983	6.147022	16.75331	15.51008	14.54277
7	7.865	5.858294	7.981133	5.019977	6.017599	16.09653	14.93846	13.99678
8	8.026545	5.932805	8.09135	5.144714	6.170092	16.88613	15.61208	14.64147

(a)

	C1	C2	C3	C4	C5	C6	C7	C8
0	0.039497	0.020392	0.033294	0.037751	0.043814	0.214736	0.190526	0.177759
1	-0.09603	-0.04648	-0.06882	-0.07515	-0.09149	-0.44979	-0.39195	-0.37114
2	0.052992	0.023547	0.037065	0.04518	0.058989	0.302741	0.256675	0.247323
3	0.041084	0.021456	0.033313	0.03217	0.037226	0.180485	0.161465	0.152244
4	-0.09592	-0.04512	-0.06944	-0.07644	-0.0933	-0.47443	-0.40991	-0.38906
5	0.066095	0.028997	0.039378	0.046636	0.057944	0.296189	0.250365	0.240137
6	0.030743	0.016753	0.028777	0.026011	0.029499	0.151686	0.136174	0.128185
7	-0.10066	-0.04699	-0.07209	-0.0806	-0.09826	-0.50629	-0.43469	-0.41609
8	0.062194	0.027448	0.03852	0.044452	0.055574	0.284673	0.24135	0.230637

(b)

Figure 5.12. The data set used for performing PCA. (a) The original experimental data. (b) The data set with the means being subtracted.

The steps for performing a PCA is explained with one of the test datasets. The original data set is found in Figure 5.12 (a), it comes from the second classification technique by scanning and recording the values of the eight mutual capacitors. C_1 to C_8

are the eight variables for each complete scan process, and the number 0 to 8 represents one of the nine categories.

The first step is to subtract the mean from each of the data dimensions. The mean values being subtracted are the averages across each dimension: all the C_1 values have $S_n(C_1)$ subtracted and so on until C_8. This step produces a data set whose mean is zero as shown in Figure 5.12 (b). After that, the covariance matrix is calculated. The covariance is always measured between two dimensions, and for our eight-dimensional data set, all the possible covariance values between all the different dimensions can be summarized in an 8×8 matrix as explained in Figure 5.13, on the left. In this matrix, down the main diagonal, each covariance value, which is between one of the dimensions and itself, represents the variance for that dimension. Since cov(a,b) = cov(b,a), this matrix is symmetrical about the main diagonal. The calculated covariance matrix for our data set is given in Figure 5.13 on the right side. It can be expected that all the capacitance values should increase/decrease together as the non-diagonal elements are all positive.

$$\begin{pmatrix} cov(C_1, C_1) \cdots cov(C_1, C_8) \\ \ddots \\ cov(C_8, C_1) \cdots cov(C_8, C_8) \end{pmatrix} \begin{pmatrix} 0.0055 & 0.0026 & 0.0039 & 0.0043 & 0.0053 & 0.0267 & 0.0230 \\ 0.0026 & 0.0012 & 0.0018 & 0.0020 & 0.0025 & 0.0126 & 0.0108 \\ 0.0039 & 0.0018 & 0.0028 & 0.0031 & 0.0038 & 0.0190 & 0.0164 \\ 0.0043 & 0.0020 & 0.0031 & 0.0034 & 0.0042 & 0.0211 & 0.0182 \\ 0.0053 & 0.0025 & 0.0038 & 0.0042 & 0.0051 & 0.0258 & 0.0223 \\ 0.0267 & 0.0126 & 0.0190 & 0.0211 & 0.0258 & 0.1307 & 0.1128 \\ 0.0230 & 0.0108 & 0.0164 & 0.0182 & 0.0223 & 0.1128 & 0.0973 \\ 0.0219 & 0.0103 & 0.0156 & 0.0173 & 0.0212 & 0.1074 & 0.0926 \end{pmatrix}$$

Figure 5.13. The 8×8 covariance matrix for the data set.

Since the covariance matrix is square, the eigenvectors and eigenvalues for the obtained matrix can be calculated and expressed in Figure 5.14 (a) and Figure 5.14 (b) respectively. These eigenvectors are all unit eigenvectors (i.e. their lengths are all 1) and they are perpendicular to each other [188]. More importantly, the eigenvalues provide us with the information about the patterns located behind the data: the eight data sets are related mostly along the eigenvector that with the highest eigenvalue. In fact, the eigenvectors, also known as the principal components, can be ordered by their corresponding eigenvalues from highest to the lowest. This order represents the component's order of significance. The components with less significance can be discarded without losing much information.

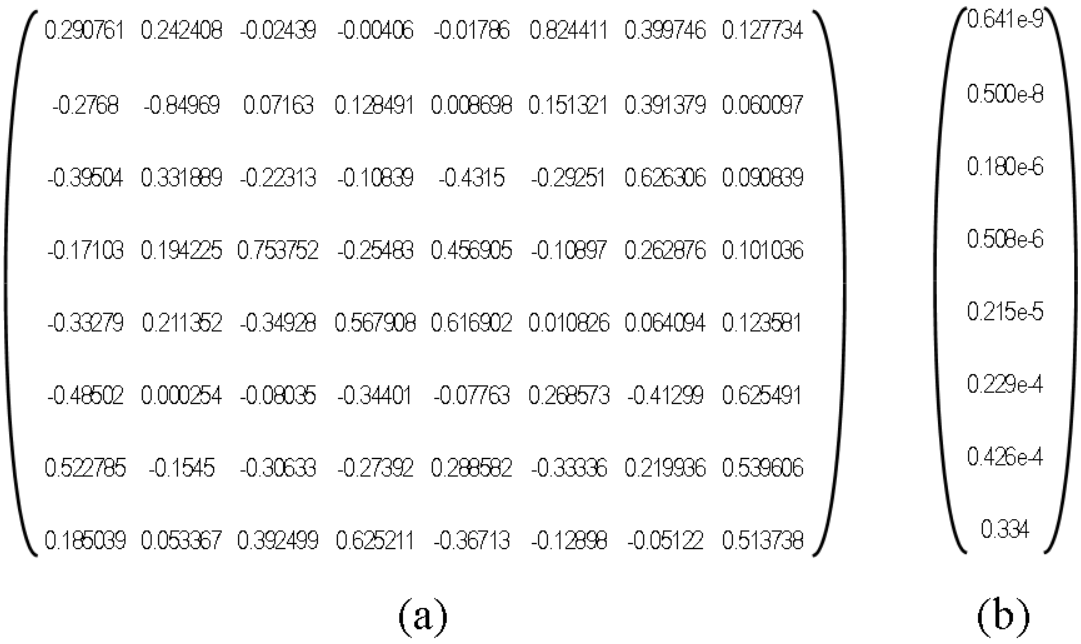


Figure 5.14. Calculated eigenvectors and eigenvalues. (a) The 8×8 matrix composed of all the eigenvectors. (b) The 8×1 matrix of eigenvalues

After the process of calculating the eigenvectors of the covariance matrix, the lines that characterize the data can be extracted. The rest of the steps involves transforming the raw data so that it is expressed in terms of the eigenvectors. More specifically, our eight-dimensional original feature vector generates eight eigenvectors, if only the first p eigenvectors are selected as the principal components, then the dimension of the final data set is p instead of eight. The method of obtaining the final data is to take the transpose of the matrix that composed of the compressed eigenvectors and multiply it on the left of the original data set, transposed, as expressed in Equation (5.1).

$$Final\ Data = \left(eig_1 \cdots eig_p\right)^T \times (C_1 \cdots C_8)^T \qquad (5.\ 1)$$

5.4.3. Data preparation

In this project, we tried two different classification techniques for recognizing the geometrical features of an approaching object. Moreover, we proposed two different capacitor scan types as described in detail in Section 3.2.3. That results in four different data sets with each data set representing the capacitive responses experimentally generated by one independent approach. All of these four datasets will be fed to the statistical learning tools for the purpose of training and testing. To be more specific, the

four approaches are the results of the combination between the classification techniques and the mutual capacitor scan types.

In the first scanning method, eight mutual capacitors are swept. The electrode connection modes for generating these eight mutual capacitors are depicted in Figure 3.13 (b). Six of the capacitors are formed by using two adjacent columns or two adjacent rows. The other two capacitors are generated between diagonal electrodes within the electrode matrix. This kind of scanning method results in the eight-dimensional feature vectors (i.e. $\{C_1, C_2, C_3, C_4, C_5, C_6, C_7, C_8\}$).

The second scanning method is inspired by the performance of the ring-shaped electrode connection structure. Four neighboring electrodes are connected to act as the transmitter and all the rest electrodes in the matrix work as the receiver. In this attempt, nine mutual capacitors as shown in Figure 3.13 (c) are scanned in turn. Thus, the obtained feature vectors are nine dimensional (i.e. $\{C_{I}, C_{II}, C_{III}, C_{IV}, C_{V}, C_{VI}, C_{VII}, C_{VIII}, C_{IX}\}$).

In terms of the classification techniques, the difference between the two techniques is whether or not it provides distance and shape information at the same time. The first technique divides all the acquired feature vectors into three classes depending on the shape information: the sphere, the plate, and the cylinder. In this technique, 900 independent measurements are performed for each class so that in total 2700 instances are obtained.

The second technique offers a finer classification that contains nine separate classes based on the distinct shape-position status. They are sphere with the distance of 5cm, 10cm, and 15cm (S@5, S@10, and S@15); plate with the distance of 5cm, 10cm, and 15cm (P@5, P@10, and P@15); and cylinder with the distance of 5cm, 10cm, and 15cm (Cy@5, Cy@10, and Cy@15). 300 measurements are carried out for each category, and thus, the total data points for the second technique is also 2700.

The first data set is the combination of the first classification technique and the first mutual capacitor scanning method. Each feature vector contains eight attributes and is labeled to one of the three classes. The second approach still follows the first classification technique but with the second scanning mode: The feature vectors are nine-dimensional. The third and fourth data sets are both generated according to the second classification technique. Both of them are divided into nine classes. The feature vectors in the third data

set are eight-dimensional whereas feature vectors in the fourth data set are nine-dimensional. The total numbers of observations for the four data sets are all the same that is 2700. Each dataset is split into two separate part with one contains 1800 samples for the purpose of training the learning machine, and the rest 900 different data points are used for testing.

The intermediate step between acquiring raw capacitive values and applying the statistical learning tools is scaling the data. This step is important and almost required for every learning algorithm is because scaling can avoid attributes in greater numeric ranges dominating those in smaller numeric ranges. In addition, it can also avoid numerical difficulties during the calculation. In this project, we linearly scale each attribute in all the four datasets to the range of [-1, +1]. In summary, the characteristics of the four data sets are given in Table 5.5.

Table 5.5. Description of the data sets

Data set	Classes	Class Description	Instances	Attributes	Attribute description	Training data	Testing data	Scale
1	3	Shapes	2700	8	6 adjacent column/row capacitors + 2 diagonal capacitors	1800	900	[-1,1]
2	3	Shapes	2700	9	9 capacitors between 4 neighboring electrodes and others	1800	900	[-1,1]
3	9	Distance & shape	2700	8	6 adjacent column/row capacitors + 2 diagonal capacitors	1800	900	[-1,1]
4	9	Distance & shape	2700	9	9 capacitors between 4 neighboring electrodes and others	1800	900	[-1,1]

5.4.4. K nearest neighbors (KNN) algorithm for classification

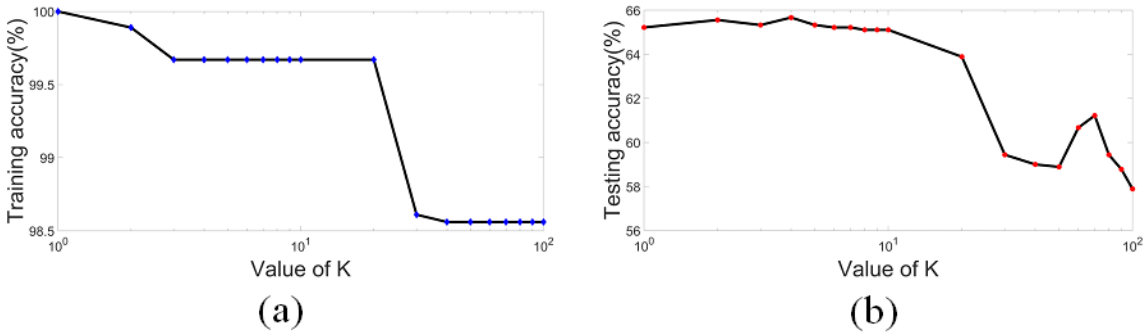


Figure 5.15. (a) Training accuracy vs. "K". (b) Testing accuracy vs. "K"

The k-nearest neighbors (KNN) model is a very simple but powerful tool that has been widely used in many applications and particularly in classification tasks [189]. In this project, MATLAB KNN function is applied to perform the classification on the abovementioned four data sets. Each sample is then classified according to the class memberships of its "k" nearest neighbors, as determined by the Euclidean distance in the proposed eight or nine-dimensional space.

The performance of KNN classification tool is highly dependent on the application specification "k" so we study the impact of varying "k" in terms of the classification accuracy. The third data set which provides both the distance and the shape information at the same time is used for this investigation. Every observation in this data set is eight-dimensional and is labeled as one of the nine classes. Figure 5.15 summarizes the training accuracy and the testing accuracy as a function of different values of k in (a) and (b) respectively. In this study, k varies from 1 to 100, with the step length of 1 when increasing k from 1 to 10 and 10 after that. As can be expected, when setting the number of k to be 1, there is no error exists in the training set, however, the testing error rate is not optimal. The training error rate drops slightly with the increase of the k, and the testing error experiences a small upward trend at the beginning but falls after the highest point. In this case, the change of k does not have too much effect on the improvement of error rate in terms of training and testing. It can also be observed that even though with a high training accuracy, the performance on the testing data is relatively poor.

The KNN tool categories query points based on their distances to points in a training data set. The most commonly used distance metric for KNN classifier is Euclidean distance, but there are several more distance metrics can also be applied for the purpose

of KNN classification. These distance metrics include standardized Euclidean distance, Mahalanobis distance, city block distance, and Chebyshev distance. Using different distance functions might affect the performance of the KNN classifier.

To measure the "distance" between points X and Y in a feature space, the aforementioned distance metrics are the results of different distance functions. Given A and B are represented by feature vectors $X = (x_1, x_2, \ldots, x_m)$ and $Y = (y_1, y_2, \ldots, y_m)$ where m is the dimension of the feature space. The Euclidean metric is the straight-line distance that can be expressed as:

$$d\ (Euclidean) = \sqrt{\sum_{i=1}^{m}(x_i - y_i)^2} \qquad (5.\ 2)$$

To balance out different contributions, some form of standardization is performed to the original data points. The conventional way to do this is to transform the variables so that they all have the same variance of 1 [190]. The transformation is commonly done as (original value – mean)/standard deviation, so the standardized Euclidean distance can be expressed as follows where s stands for the standard deviation.

$$d(Standardized\ Euclidean) = \sqrt{\sum_{i=1}^{m}(\frac{x_i - mean_x}{s_{xi}} - \frac{y_i - mean_y}{s_{yi}})^2} \qquad (5.\ 3)$$

Mahalanobis distance is the distance between two points in multivariate space, it measures the distance relative to the centroid. It solves the problem when there are more than three variables, and some of these variables are correlated. If C is the sample covariance matrix, the Mahalanobis distance can be calculated with:

$$d(\text{Mahalanobis}) = \sqrt{(X - Y)'C^{-1}(X - Y)} \qquad (5.\ 4)$$

City block distance is the shortest distance between the two points is along the hypotenuse. More specifically:

$$d(\text{city block}) = \sum_{i=1}^{m}|x_i - y_i| \qquad (5.\ 5)$$

Chebyshev distance, also known as maximum metric, is a metric defined on a vector space where the distance between two vectors is the largest of the differences along any coordinate dimension:

$$d(\text{Chebyshev}) = \max|x_i - y_i| \qquad (5.\ 6)$$

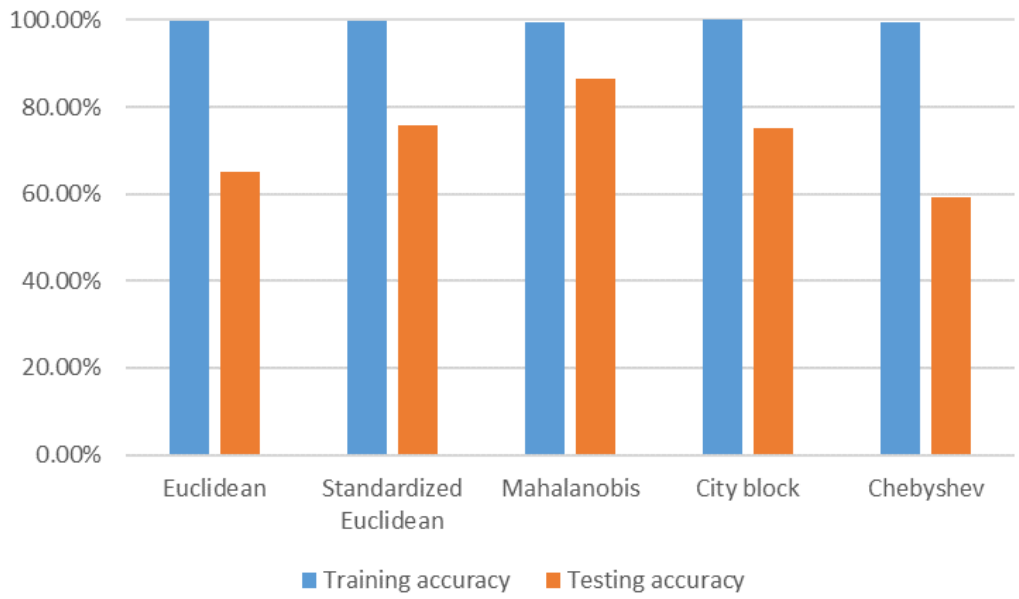


Figure 5.16. KNN performance as a function of different distance metrics (K = 10).

The performance of KNN classifiers using different distance functions is tested with data set 3 and with the value of "k" equals to 10. The average testing time it consumes is 0.12 s. Both training accuracy and testing accuracy result from all these distance metrics are plotted with the histogram as depicted in Figure 5.16. From which it can be observed that the training accuracies are all good (greater than 99%) regardless of the involved distance functions. In terms of testing accuracy, Mahalanobis function results in the best performance (86.56% testing accuracy) whereas the lowest testing accuracy (59.22%) comes from Chebyshev metric. This result indicates that some attributes inside this data set (data set 3) might be correlated to some other attributes.

The performance of KNN classifier is tested with all the four different datasets for comparison. Both the training accuracy and testing accuracy are provided in Table 5.6. From these numbers, it can be observed that the KNN classifier is able to achieve a high training accuracy (higher than 90%) and better performance for the tasks with fewer categories. When classifying the data into three classes, the second data set that each observation has nine attributes exhibits higher training and testing accuracies compared to the 8-dimensional datasets. That gives us a guidance on the choice of electrode connection patterns: measuring the mutual capacitors between four neighboring electrodes and all the rest electrodes as a whole is able to provide more information in distinguishing among the three shapes. However, when trying to perform a finer

classification with more categories, it is hard for KNN to give out a satisfied result. The testing accuracies with both the two data sets are 65.11% and 58.11%, respectively.

Table 5.6. KNN classification performance with different data sets

KNN	K = 10			
	Data set 1	Data set 2	Data set 3	Data set 4
Training accuracy	93.11%	95.54%	99.67%	91.17%
Testing accuracy	75.33%	95.89%	63.89%	58.11%

In addition to the low testing accuracy, another concern when applying KNN classifier is it only reveals the resulting classes, not provides the probability information. This disadvantage prevents it from being used for the more detailed analysis of the object's geometrical features. As a simple algorithm, its computational cost is very small, and the needed testing time is much less than 0.1 ms. In summary, KNN classifier can provide high testing accuracy and small computational cost classification solutions for simple tasks. However, for more complex situations with more categories, its performance is insufficient.

5.4.5. Neural networks (NN)

Neural networks (NN) gather their knowledge by looking for the underlying patterns and the relationships between the input attributes and the target classes in the data. NNs learn through experience, not from programming. This feature makes the NN a promising classification technique for our project where the relationship between the experimental capacitive data and its corresponding category is non-linear. An NN is formed from multiple artificial neurons, connected with coefficients (weights) that constitute the neural structure, and organized in layers [191]. During the training process, the inter-neuron connections are optimized until the error in predictions is minimized and the network reaches the specified level of accuracy.

In this project, the NN using supervised training and back-propagation rule is applied as it performs well at prediction and classification tasks. This type of network is a system of fully interconnected neurons organized in the input layer, the output layer, and the hidden layers between them. According to the theory presented in [192], most functions can be approximated using a single hidden layer. So in this project, the number of hidden layers is selected as one without further investigation. The hidden neurons are

part of the large internal pattern that determines a solution to a problem by communicating with other neurons. They are the building components of the neural networks, and the number of hidden neurons has a significant effect on the classification performance. In this section, the study of different numbers of hidden neurons is performed.

For classification problems, the NN goal is to model the posterior probabilities of class membership, conditioned on the input variables. In order to provide a better estimation of the performance of an NN, a more appropriate error function, the cross-entropy is considered. The concept of entropy was originally developed by physicists in the context of equilibrium thermodynamics and later introduced into information theory by Shannon [193]. The differences of information content between two probabilities distributions are measured by cross-entropy (CE), which is a lower convex function [194]. The loss function for classification problems can be constructed by cross-entropy [195] as expressed in Equation (5.7).

$$L_{CE} = -\frac{1}{N}\sum_{n=1}^{N} y_n^T \log P_n \qquad (5.7)$$

where the $\log(P_n)$ is an element-wise logarithmic function, and P_n is the output of the network. The cross-entropy holds several theoretical advantages over other error functions including squared error cost function. It is a log-linear error function so that it is impacted less by the outliners. It depends upon the relative errors of the network outputs, hence, cross-entropy could perform equally well on both large and small target values because they tend to result in similar relative errors. Cross-entropy minimization technique is frequently applied in the optimization of the classifier. Therefore, cross-entropy is used as the indicator for the training and testing performance in this study.

The same as the KNN classifier, the third data set in which the eight-dimensional feature vectors are targeted to nine classes is used for studying the NN algorithm parameters. MATLAB Neural Pattern Recognition tool is applied to perform the classification task. The neural networks being tested all contain a single hidden layer with the neurons varies from 1 to 10. Too few hidden neurons will hinder the learning process and too many will depress prediction abilities through overtraining. By increasing the number of the hidden neurons the NN could more closely follow the topology of the training dataset. However, exceeding an optimum number might result in overfitting and lower testing accuracy. The networks are trained in multiple epochs, in which all data from training corpus are sequentially presented. Neuron weights are initialized to small values,

and the standard back-propagation algorithm with stochastic gradient descent is used to train the networks [196]. After each epoch, the neural network is tested on the validation data. If log-likelihood of the validation data increases, the training process continues in new epoch until no significant improvement can be observed. The results are summarized in Table 5.7, in addition to the achieved cross-entropy and the number of epochs, the training time consumption is also included in this table.

Table 5.7. Neural network results with respect to the number of neurons

Neurons	1	2	3	4	5	6	7	8	9	10
Time	9s	15s	13s	16s	17s	56s	20s	67s	43s	49s
Epochs	69	102	92	116	122	393	146	462	311	351
Testing CE	0.114	0.0864	0.0882	0.0516	6.45e-3	1.42e-4	6.62e-3	6.41e-7	2.36e-5	1.46e-7
Testing accuracy (%)	89.23	91.72	91.56	94.97	99.36	99.99	99.34	100	100	100

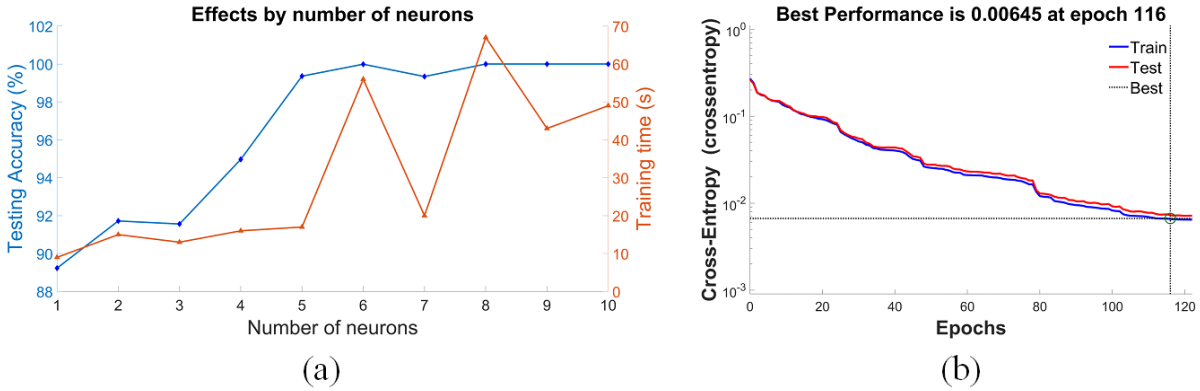


Figure 5.17. Neural network training performance. (a) Testing accuracy and training time consumption as functions of the number of neurons. (b) Testing cross-entropy as a function of epoches used.

From Table 5.7, it can be observed that the more neurons being used in the hidden layer, the better performance (the lower cross-entropy values and higher accuracies) the network can achieve. However, better performance is gained with the cost of larger computational cost: The larger number of neurons leads to taking more epochs to meet the minimum cross-entropy criterion. The testing accuracy and the training time consumption as functions of the number of neurons are shown in Figure 5.17 (a). When the number of neurons reaches 5, the testing accuracy is greater than 99%, and the

training time consumption which is 17s, is still within a moderate range. The required testing time consumption is 0.13 s. So the number of neurons is hereby selected as 5.

The NN performance in terms of training CE and testing CE with respect to the epochs is plotted in Figure 5.17 (b). In this case, the best performance is achieved at epoch 116. The training accuracy and the testing accuracy are similar while the training CE is slightly lower. Moreover, the training confusion matrix and the testing confusion matrix are given in Figure 5.18. The network can provide almost perfect classification results with both training data and testing data.

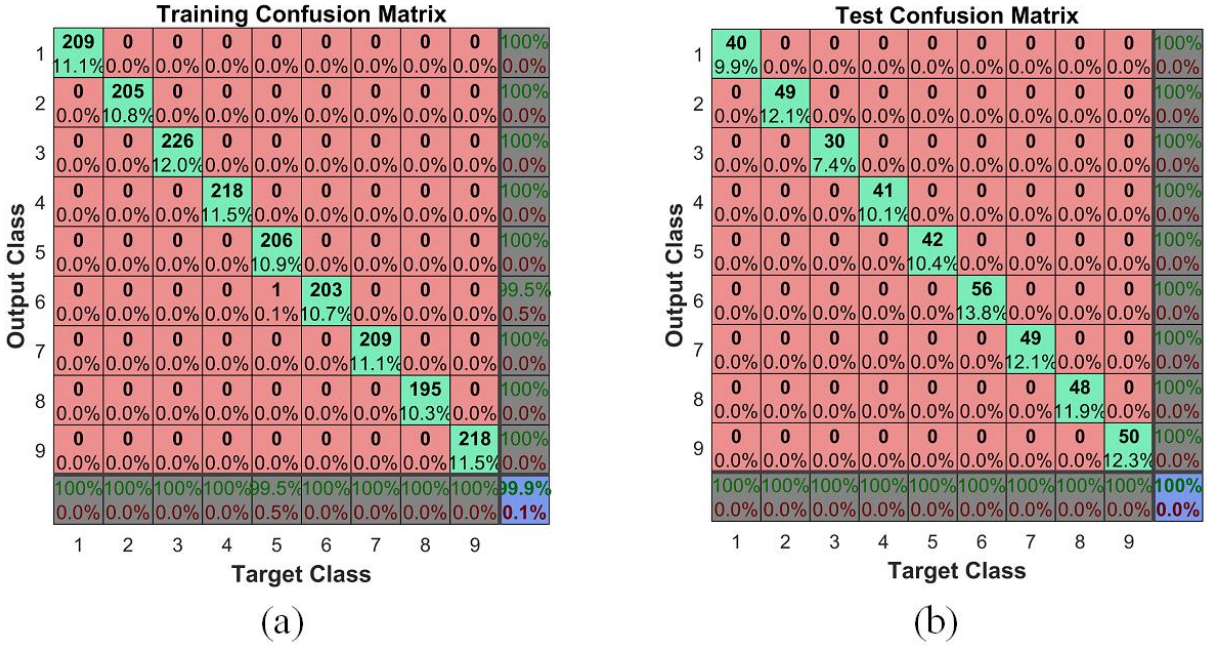


Figure 5.18. Neural network confusion matrices. (a) Training confusion matrix. (b) Testing confusion matrix

Again, which of the different capacitor sweeping modes can provide the best classification performance is one of the issues of interest. In this investigation, the number of hidden layers and the number of the neurons are set as one and five respectively. The neural network training results with all the four data sets are summarized in Table 5.8. For classification tasks involve more categories, the performance of NN is similar with either eight-dimensional or nine-dimensional feature vectors. The fourth data set is able to provide a slightly higher accuracy with a little less time consumption. However, the different scanning patterns lead to quite different results for classifying the data into three different surface profile classes. Training with data set 2 results a much lower cross-entropy with less training time.

Table 5.8. NN classification performance with different data sets

NN	Neurons = 5			
	Data set 1	Data set 2	Data set 3	Data set 4
Time	2s	1s	17s	16s
Epochs	27	29	122	110
Cross-entropy	0.161	6.54e-7	0.00645	0.00219
Testing Accuracy	85.13%	100%	99.36%	99.38%

In summary, the neural network is a powerful classifier to solve the recognition problems in this project. It provides high testing accuracy with every data set for each classification task. The only concern when applying NN is the high computational cost, which might restrict its usage in high-frequency systems.

5.4.6. Support vector machine (SVM)

As the sensing capacitance values are grouped into eight or nine-dimensional vectors for the purpose of profile recognition, the support vector machine (SVM) with a non-linear kernel function is employed. SVM is a classification tool which maps the features into a high-dimensional feature space and constructs a hyperplane to separate binary classes. It has been for instances proven successful for robotic arm control [197] and for patient locomotion-mode identification [198]. In addition to the high-dimensional feature, SVM classifier is selected as a promising candidate because it can classify accurately using nonlinear boundaries when linear boundaries are difficult to define. In this study, LIBSVM [199] which supports “one-against-one” (OAO) multiclass classification and employs 5-fold cross-validation is used.

The third data set that contains eight-dimensional feature vectors is fed to LIBSVM package for model training and new samples testing. LIBSVM supports for multiclass classification as well as estimating class-conditional probabilities for any given feature vector. SVM classification accuracy, defined as the percentage of correctly classified testing samples out of the total testing data, is dependent on the algorithm parameters β and γ. Cost parameter (β) determines various different trade-offs between computational cost and accuracy, and gamma (γ) defines how far the influence of a single training sample reaches. It is not known beforehand what combination of β and γ pair could lead to the best testing result for any given problem, so a series of different parameters are tested

with the experimental data set in this study. The investigated values for gamma varies from 0 to 8, and the range of studied cost parameter is 0 to 600. The impact of cost parameter and gamma on overall recognition accuracy is demonstrated in the 3D plot Figure 5.19. It can be observed that the overall testing accuracy is increasing from 74.67% to 90.33%, and it tends to be stable and at a relatively high level with moderate β and γ. As the definition of the cost parameter β, a larger value of β tends to consume more computational resources, so the parameter pair of β = 200 and γ = 4 is selected for this specific task.

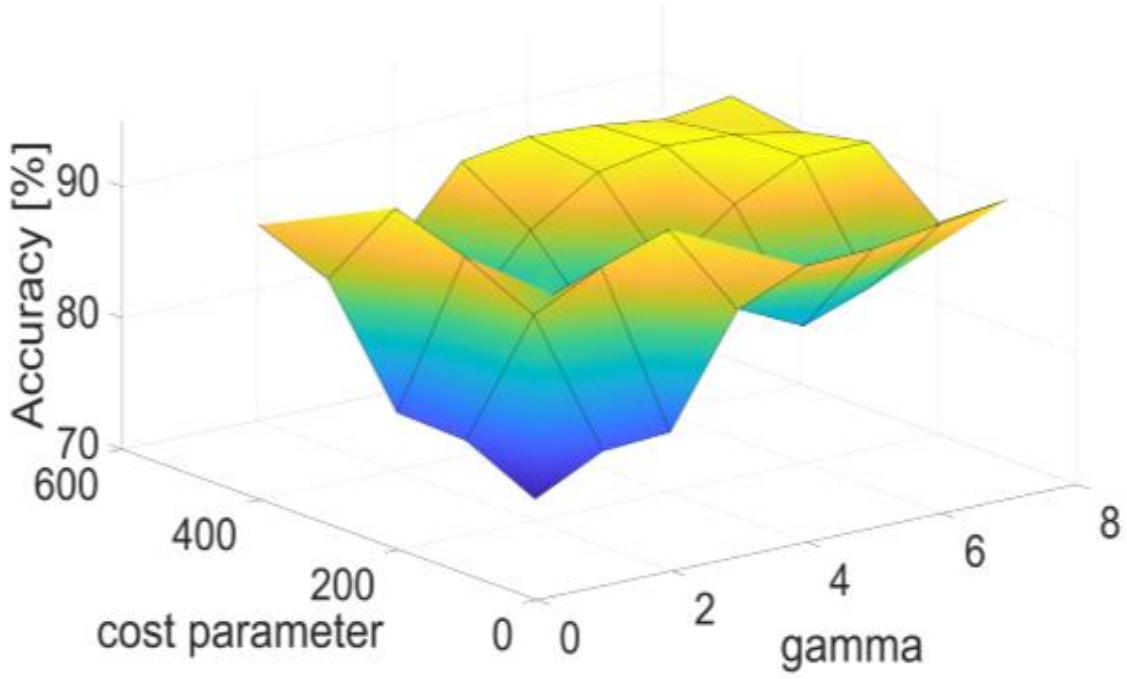


Figure 5.19. Impact on testing accuracy of SVM algorithm parameters: cost parameter (β) and gamma (γ)

For a more comprehensive investigation of the performance of the SVM classifier, the classify probabilities are tested. Nine new experimental data points, each belonging to one of the 9 classes, are provided for classification. The estimated probabilities are summarized in Table 5.9. In most cases, surface profile together with distance information can be recognized with high confidence (more than 95% probability). However, though classifying to the correct category, the boundaries among plate at 5 cm, cylinder at 5 cm and sphere at 10 cm are ambiguous to some extent.

Table 5.9. Confusion matrix with SVM classifier (%)

Actual class	Estimation class								
	S@5	*P@5*	*Cy@5*	*S@10*	*P@10*	*Cy@10*	*S@15*	*P@15*	*Cy@15*
S@5	95.158	0.914	1.031	0.257	0.0658	0.0547	0.217	0.0591	0.0233
P@5	0.227	46.055	38.052	5.875	0.0386	0.0328	0.0631	0.0374	0.197
Cy@5	0.165	38.820	42.923	45.900	0.0330	0.0280	0.0479	0.0322	0.0898
S@10	0.148	10.663	14.003	47.050	0.0300	0.0255	0.0451	0.0292	0.0759
P@10	0.0261	0.524	0.588	0.35	99.677	0.0026	0.0766	2.565	0.0276
Cy@10	0.0229	0.458	0.514	0.119	0.0037	99.689	0.0647	0.449	0.0240
S@15	0.159	0.733	0.824	0.190	0.0954	0.0787	99.253	0.0649	0.0082
P@15	0.0302	0.611	0.686	0.158	0.006	0.0476	0.0957	96.726	0.0317
Cy@15	4.064	1.220	1.377	0.314	0.0497	0.0416	0.134	0.0365	99.523

The performance of the SVM classifier might be influenced by using different electrode connection patterns. In order to figure out the most suitable pattern, all the four data sets are fed to the SVM algorithm for the purpose of training and testing. The selected values for the cost parameter (β) and gamma (γ) are 200 and 4, respectively. The obtained training accuracy and testing accuracy are listed in the Table 5.10, and the average testing time consumption is 0.043 s. From this table, it can be observed that SVM performance is highly dependent on the choice of sweeping mode. For the classification task with fewer categories, dataset 2 that scanning the nine mutual capacitors between the four neighboring electrodes and all the other electrodes shows superior to the other sweeping mode. Using dataset 2 provides a perfect SVM classifying performance: both the training accuracy and the testing accuracy is 100%. On the contrary, for the task of classifying the data points to finer distance/shape categories, data set 3 that contains 8-dimensional feature vectors is more suitable. Though with the similar training accuracy, the testing accuracy with data set 3 is much higher than with data set 4 (90.33% vs. 66.57%).

Table 5.10. SVM classification performance with different data sets

SVM	B = 200; γ = 4			
	Data set 1	Data set 2	Data set 3	Data set 4
Training accuracy	80.94%	100%	99.94%	91.44%
Testing accuracy	63.33%	100%	90.33%	66.57%

Overall, the multi-class SVM method can work as a powerful tool for both classification tasks with suitable choice of the data sets. The computational time consumption for one single classification is less than 0.1ms that is much faster than the required sensing frequency (tens of Hz). Moreover, the SVM method can be cascaded to complete more complex classification tasks for broader applications.

After performing the surface recognition task with the three different classifiers, their performance is summarized and compared with Figure 5.20. In this figure, each bar stands for the testing accuracy acquired from the combination of a certain classifier and one specific dataset. It can be verified that the three surface profiles that are most frequently encountered in the manufacturing environment can be recognized with all of the three classifiers. Moreover, the surface profile information together with three discrete vertical distances can be identified at the same time with high confidence by using the more advanced machine learning tools (Neural Network and Support Vector Machine). Generally, the NN classifier provides the best classifying results, but its computational cost is also the highest. By adding the surface profile recognition procedure, the ambiguity brought by the different shapes can be eliminated. Therefore, the most suitable regression models presented in section 5.3 can be selected, and the accuracy of distance estimation will be improved.

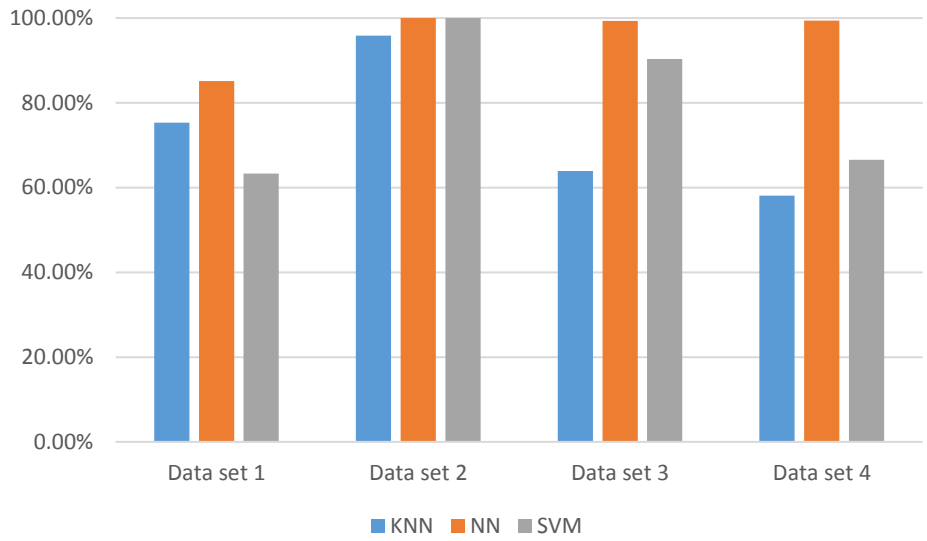


Figure 5.20. Comparison of different classifiers in terms of testing accuracy

5.5. Parallel motion tracking

The last functionality that can be achieved by the same sensing system is tracking an object which moves in parallel with the sensor. In this task, the aluminum plate is used to simulate the parallel movement. Initially, the lateral distance from the center of the moveable aluminum plate to the left edge of the sensor is 10cm. Then it moves along the reference axis, which in this case is x-axis, towards until exceeds the right edge of the

sensor for 10cm. For this task, more attention is paid to the movement direction, whereas the absolute vertical distance is of secondary interest. So the vertical distance is always kept as 5cm. For every 1 cm of displacement, the electrode matrix was programmed to scan the three mutual capacitors (C1, C2, and C3) described in Figure 3.12 (b). Thus, the trajectory history of the target object has to be analyzed continuously.

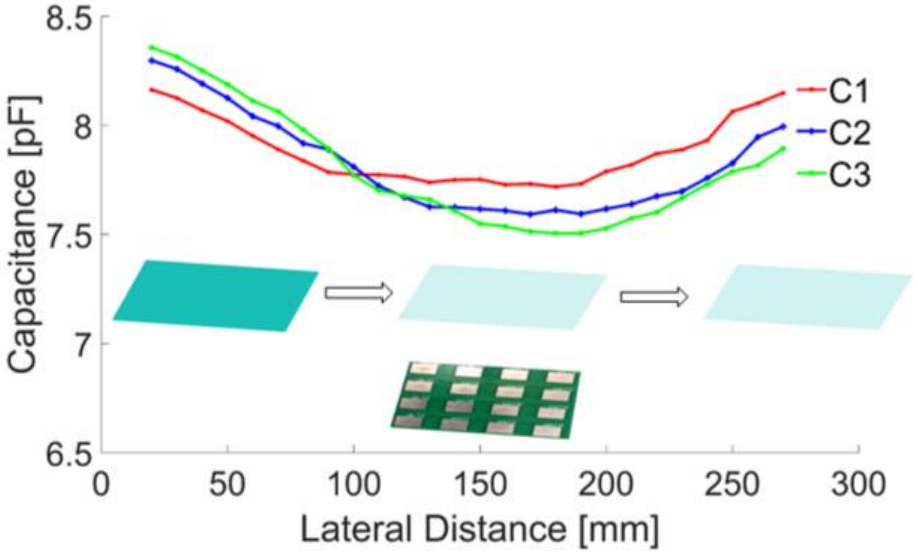


Figure 5.21. Parallel motion tracking.

The acquired capacitive readings along with the movement sketch are plotted in Figure 5.21 with red, blue, and green dots. As can be seen, the presence of an object is detected reliably by employing any one of the three capacitors, and the moving direction can be tracked by combining all the configurations together.

5.6. Distance evaluation capability on a robot

In the previous sections, the performance of the designed capacitive sensor is verified in the domain of lab equipment. In this section, its behavior properties in robotic applications will be investigated with an industrial robot KUKA LBR iiwa. The robot is a lightweight robot with a 7-axes jointed-arm. All drive units and current-carrying cables are installed inside the robot so that it can autonomously move and transport objects. It has the capability of orienting itself independently in its surroundings and moving into positions for automation tasks with millimeter precision [200].

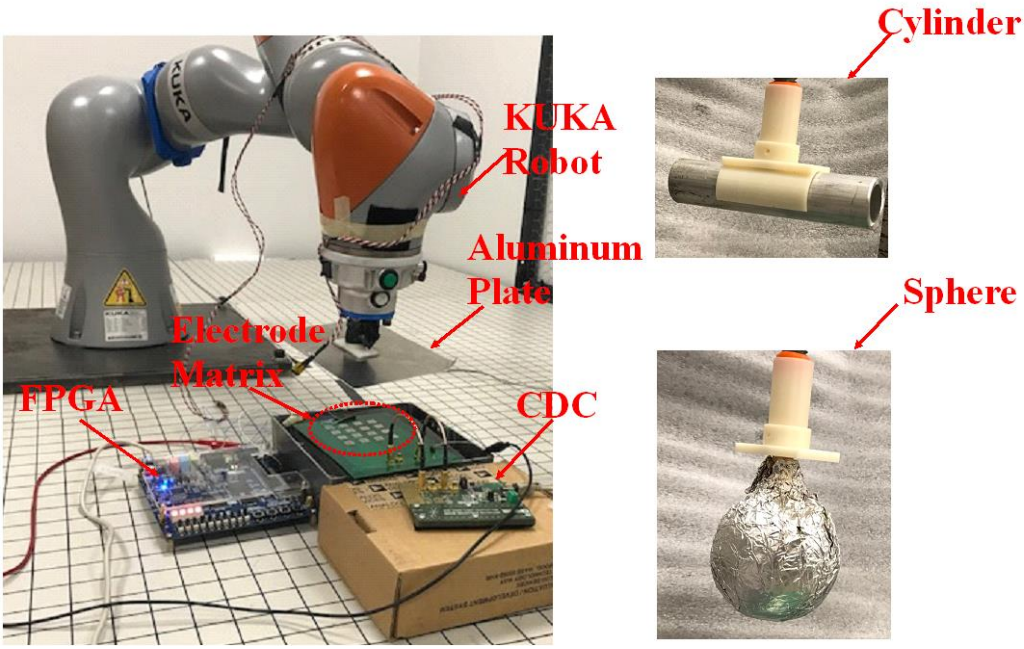


Figure 5.22. Experimental setup with KUKA robot.

To verify the shape classification would improve the accuracy of distance measurement, a series of experiments are conducted with the KUKA robot. The experimental setup is illuminated with Figure 5.22, and the KUKA robot is involved to replace the height-adjustable frame. The object is held by the robot, and the robot is programmed to perform pre-defined movements with high accuracy. The sensing electrode matrix is put underneath the robot arm which grips the desired object; the inner connection of this matrix is controlled through the cross-point switch by the FPGA; the sensed capacitive signals are acquired and quantized by the CDC chipset. Again, the three objects include a plate, a cylinder, and a ball are used exchangeably for the purpose of shape recognition.

The first set of experiments are carried out for investigating the distance measurement capability. The capacitive responses are measured with respect to the four electrode connection types (i.e. Type I, Type II, Type III, and Type IV with detailed being explained in Chapter 3) and the three objects with different shapes (i.e. plate, sphere, and cylinder). The vertical distance range of interest is from 1 cm to 20 cm.

The distance estimation functionality can be practically achieved with three steps: The sensor's inner electrode connection type is pre-selected by properly configuring the FPGA. Then program the robot to move the object from 1cm to 20cm vertically away from

the surface of the sensor with a constant speed, at the same time, the capacitive values are measured and recorded. The final step is creating the statistical regression model, that is done by sending the acquired capacitive responses and their corresponding distance values to MATLAB curve fitting tool. This experimental procedure is repeated for each electrode connection type for all the three objects. So that the performance differences can be compared.

For the plate, the raw experimental capacitive samples acquired by all the four electrode connection types are plotted with black dots in Figure 5.23. Fitted curves that stand for the mathematical formulas describe the relationship between a capacitance value and its corresponding vertical distance are denoted by the red lines. In this case, the two-term exponential regression models are able to provide high-degree of fittings. More specifically, the sum of squared errors of prediction (SSE) is close to 0 indicating the model has a small random error and the coefficient of determination R-square is almost 1 meaning that a great proportion of variance is accounted for by the model. In real-world applications, the vertical distance can be predicted with high confidence by feeding the measured capacitance to the desired regression function.

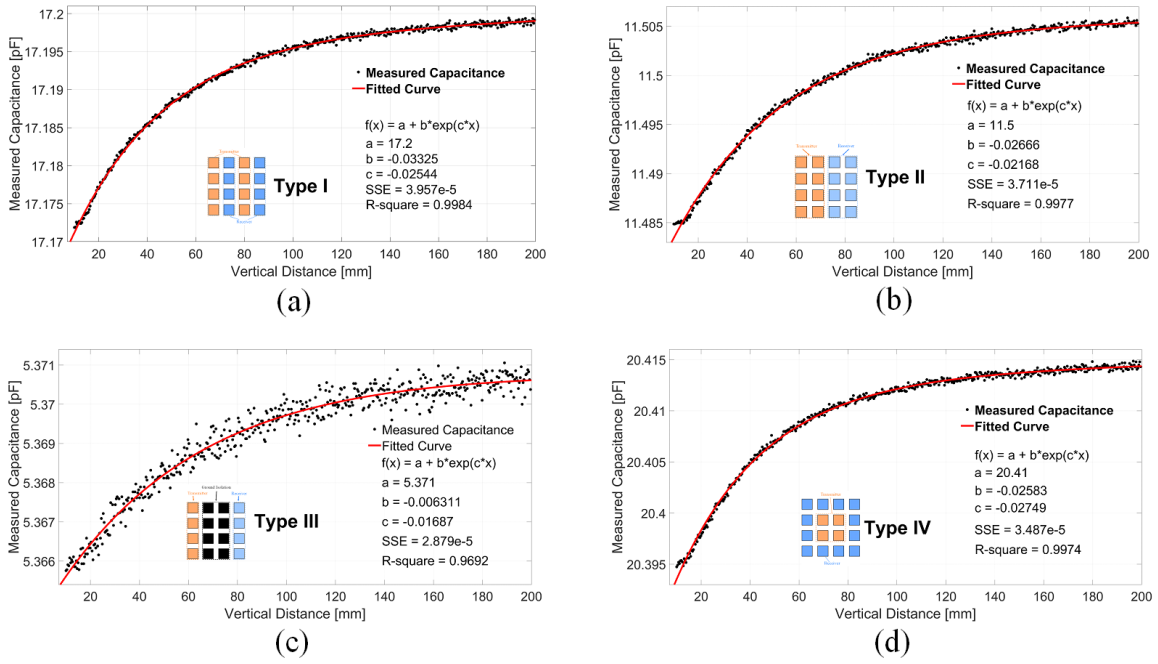


Figure 5.23. Distance measurement for the plate with the KUKA robot. Measured capacitive raw values (shown with the black dots) and the fitting curve (denoted by the red lines) with respect to four electrode connection types. (a) Type I (b) Type II. (c) Type III. (d) Type IV.

From Figure 5.23, it can be observed that the electrode connection Type IV provides the largest absolute capacitance values (starts from 20.395 pF), whereas the absolute values from Type III are the smallest (starts from 5.366 pF). Despite the great difference in terms of capacitive values, it makes more sense to perform a performance comparison by normalizing all the acquired capacitances to a common scale. More specifically, the linear scaling method is applied to rescale the data acquired from each connection type to the range of [0, 1]. Two-term exponential regression models are created based on the normalized data and then be linearly stretched to the same range of [0, 1]. The normalized fitting curves as functions of vertical distance for all the four types are summarized in Figure 5.24. Type I, II and IV exhibit similar performance in terms of linear detection range. Type III is able to achieve a slightly longer linear region, however, its capacitive responses are highly affected by the noises, as the signal strength is the weakest among the four types.

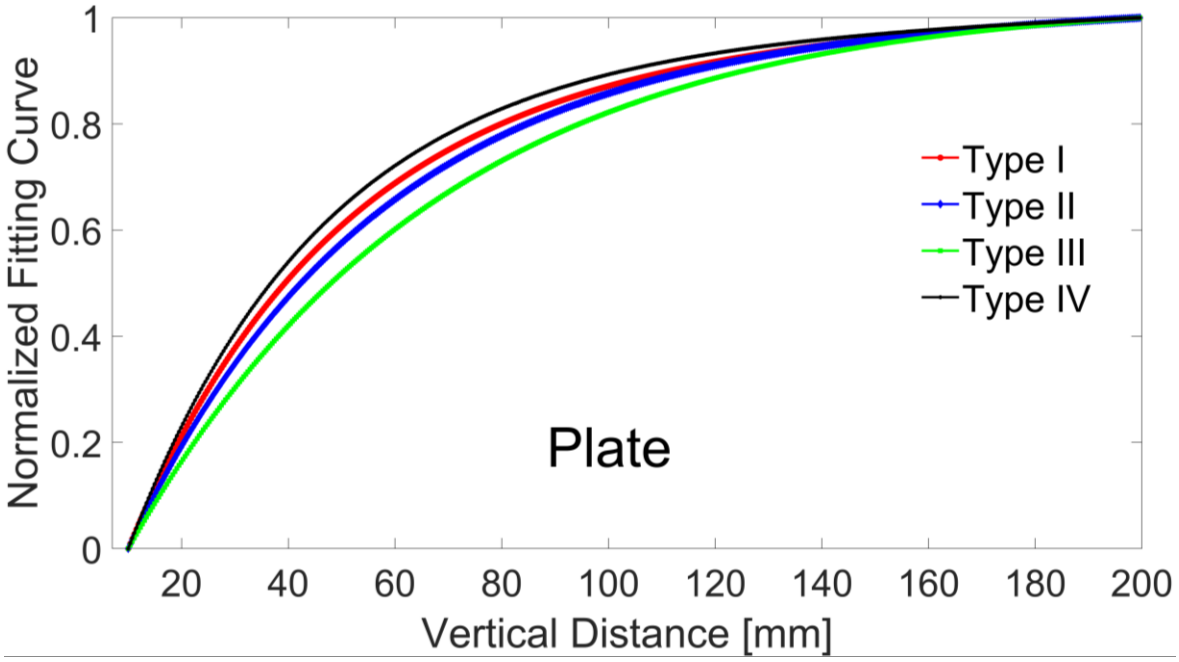


Figure 5.24. Normalized fitting curve as functions of vertical distance for the plate.

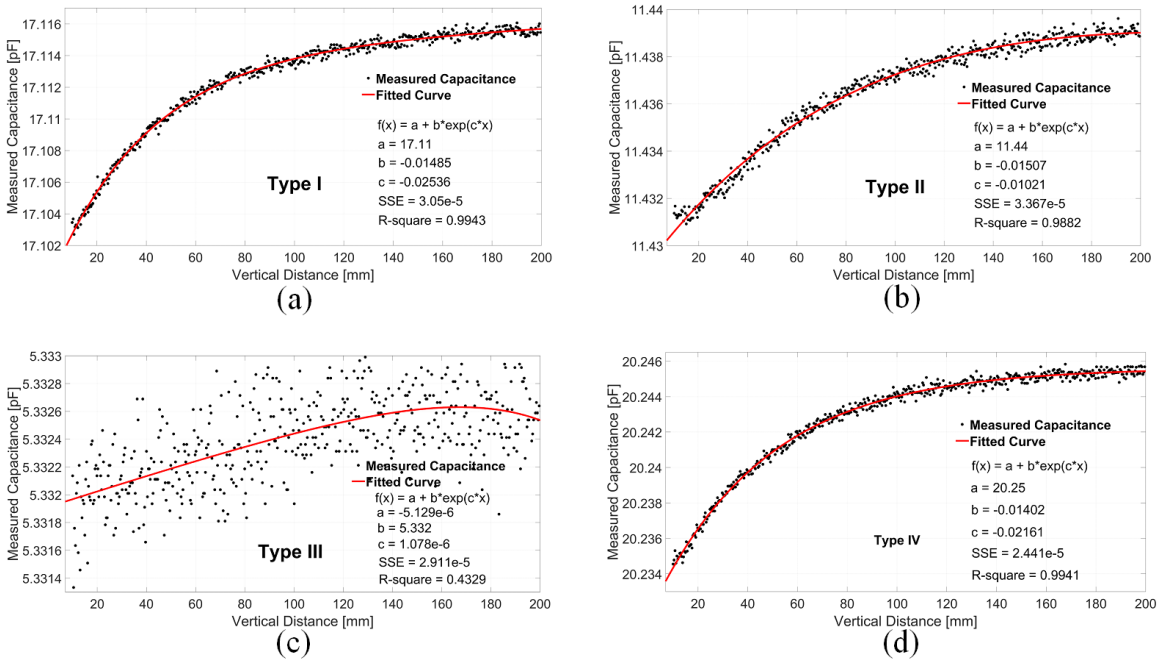


Figure 5.25. Distance measurement for the sphere with the KUKA robot. Measured capacitive raw values (shown with the black dots) and the fitting curve (denoted by the red lines) with respect to four electrode connection types. (a) Type I (b) Type II. (c) Type III. (d) Type IV.

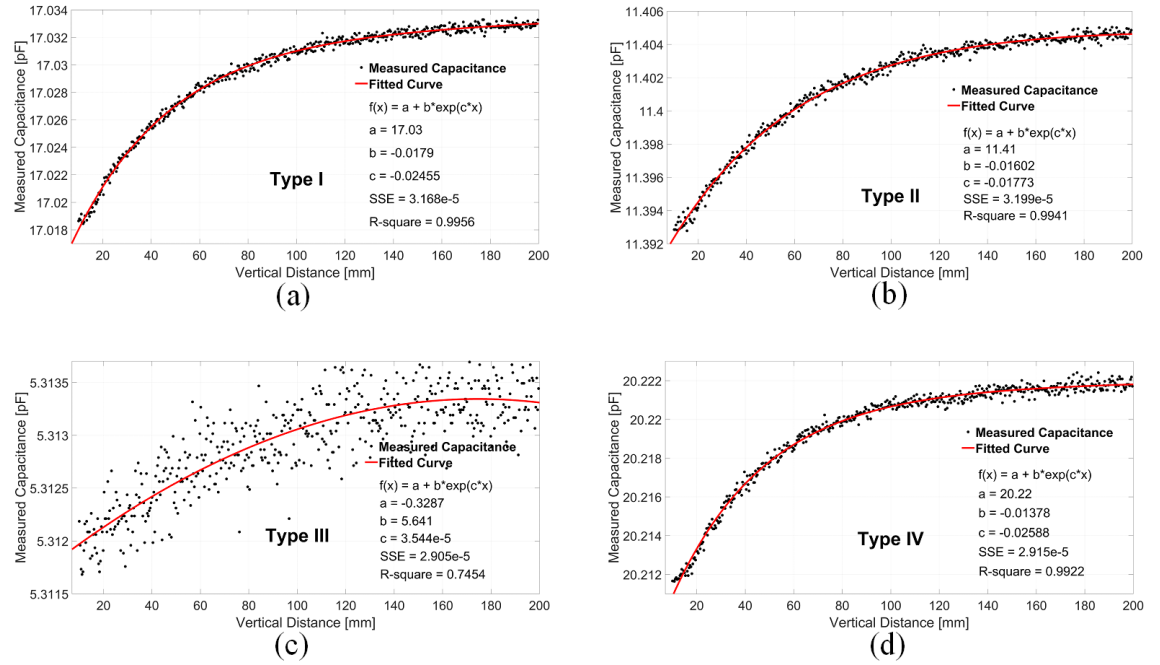


Figure 5.26. Distance measurement for the cylinder with the KUKA robot. Measured capacitive raw values (shown with the black dots) and the fitting curve (denoted by the red lines) with respect to four electrode connection types. (a) Type I (b) Type II. (c) Type III. (d) Type IV.

To investigate the sensing capabilities with respect to different shapes, the three experimental steps are repeated with the other two objects, the ball and the cylinder. The measured capacitive responses together with the corresponding fitted curves are demonstrated in Figure 5.25 and Figure 5.26 for the sphere and the cylinder respectively. Again, two-term exponential regression models provide high-degree of fittings and can be used for calculating distances from newly acquired capacitive signals.

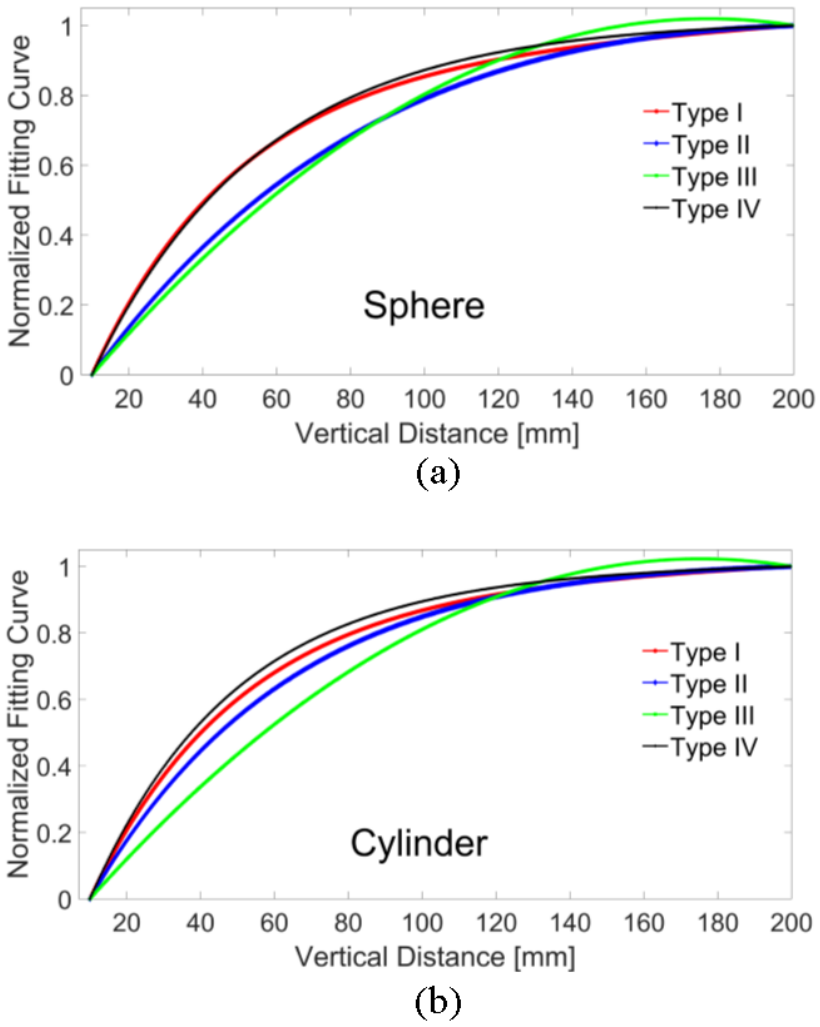


Figure 5.27. Normalized fitting curve as a function of vertical distance. (a) For the sphere. (b) For the cylinder.

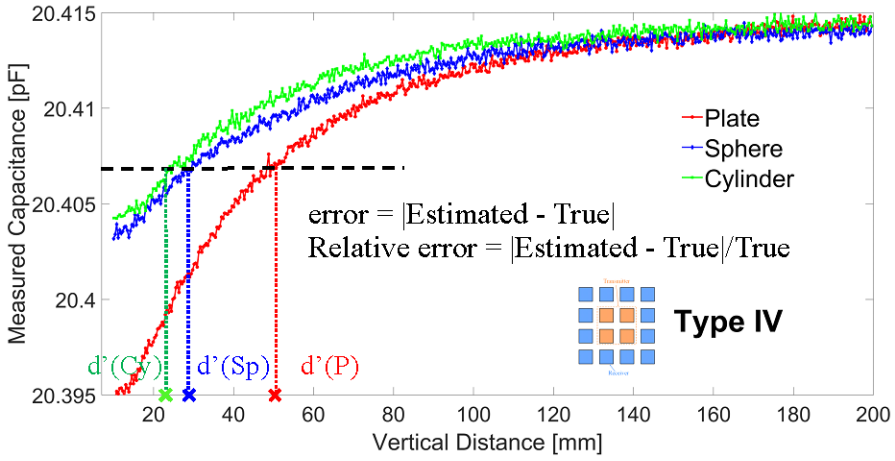
In order to get a better understanding of the distance measurement performance achieved by the four different types, the normalization is required for a comparison. Normalized fitting curves as functions of vertical distances for the ball are denoted by Figure 5.27 (a), and the same plots for the cylinder are shown in Figure 5.27 (b). For all

the three objects, there are several common facts: Electrode connection Type III that generates the largest spatial wavelength can achieve the longest linear detection range, however, it suffers from the environmental noises the most. Type I and Type IV depict similar performance in terms of linear detection range, Type I gives out larger capacitance changes whereas Type IV provides the largest absolute capacitive values. Type II offers moderate results in the aspects of linear range, signal strength, and capacitance changes. One of these electrode connection types can be selected based on the requirements and emphases of a specific application scenario.

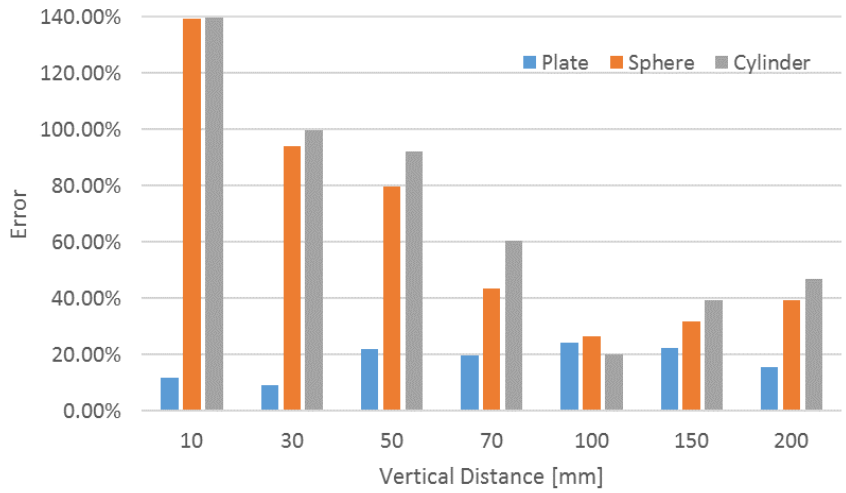
From the analysis of the experimental results, it can be verified that the vertical distances can be calculated by the exponential regression function with high accuracy under the condition of the shape of the object is pre-known. However, when comparing the data acquired by the same electrode connection type but from different objects, it can be figured out that both the absolute capacitive values and the changing rates are different. Type IV is used as an example to study the how the different shapes affect the distance evaluation accuracy as it demonstrates the least measurement difference. To eliminate the error brought by drifting, an offset canceling procedure is performed. Take the plate's asymptotic capacitance value that is measured when the plate is put infinitely far (in our measurement is 50 cm when the capacitances stop changing) from the sensor as a reference. Then shift the capacitance vs. distance curves of the other two objects to the same ending point by adding or subtracting a constant. The experimental capacitance values after offset canceling with respect to distances for all the three objects are summarized in Figure 5.28 (a). From which it can be observed that the same capacitance value can result in quite different distance values if confused about the object's shape. This fact will lead to a decreased distance evaluation accuracy or even a wrong detection.

An intuitional observation from Figure 5.28 (a) is that the shape feature would have significant impact on the estimation of distance, especially in close distance region. For a more precise analysis, error $e(d)$ and relative error $e(d)/d$ are investigated. The distance error is defined as the difference between the estimated distance and the true value, and the relative error can be expressed as the ratio of error and the true distance. For a group of discrete distances (i.e., 1cm, 3cm, 5cm, 7cm, 10cm, 15cm, 20cm), use the plate to preform the capacitive measurement. The acquired capacitance values are fed to all the three regression models to predict the distances $d'(P)$, $d'(Sp)$, and $d'(Cy)$. Consequently, the estimation error and the relative error can be calculated. The tolerance for the errors

varies with different distances: the same error value would have a larger impact on the measurement accuracy when the true distance is small.



(a)



(b)

Figure 5.28. Distance measurement error caused by unknown object shape. (a) A comparison of measured capacitance values for the three objects. (b) Calculated errors vs. vertical distances for the three objects

The regression model used here is generated by the measurement data with the plate. The calculated measurement errors as functions of discrete distances are plotted in Figure 5.28 (b), with blue, orange, and gray bars stand for the plate, sphere, and cylinder respectively. From the distance error values calculated the plate, the estimation error is less than 1 cm when the vertical distance is under 10 cm, and the maximum fitting error is 3.4 cm when the distance is 15 cm. Though the distance error increases with the vertical distance as the signal strength becomes weaker as the object moves farther from the

sensor, the measurement errors are limited to 25% all the time. That indicates the created regression model is with high accuracy and can be used for prediction. However, the same model would have a large error when applied to the other two objects, especially in short distances. The calculated error distances are competitive to the actual distances (i.e. the measurement errors are around 100%) when below 5 cm. Within the range of under 10 cm, a high detection accuracy is required in order to allow the robot to perform a proper reaction, so the regression model is not applicable.

According to the error analysis, the quality of the regression model is highly dependent on the shape of the object. Therefore, acquiring the shape information of an approaching object plays an essential role to improve the measurement accuracy. The shape recognition is achieved by performing a series of dynamic experiments and applying machine learning tools to the collected data.

The dynamic experiment is about automatically sweeping the inner electrode connection configurations so that the detailed circumstances can be monitored. For the purpose of shape classification, there are two different sweeping patterns as explained in Chapter 3. The first pattern successively measures eight independent capacitors that are composed of three capacitors generated by two adjacent columns (C_1 – C_3), three capacitors between two adjacent rows (C_4 – C_6), and two capacitors between diagonal electrodes (C_7 and C_8). In the second sweeping approach, four neighbor electrodes are grouped as the transmitter and all the rest electrodes are connected as the receiver. Moving the position of the transmitter all over the whole sensing matrix so that nine independent capacitors (C_I – C_{IX}) are generated. A graphical illustration of both the sweeping patterns are provided in Figure 3.13 (b) and (c).

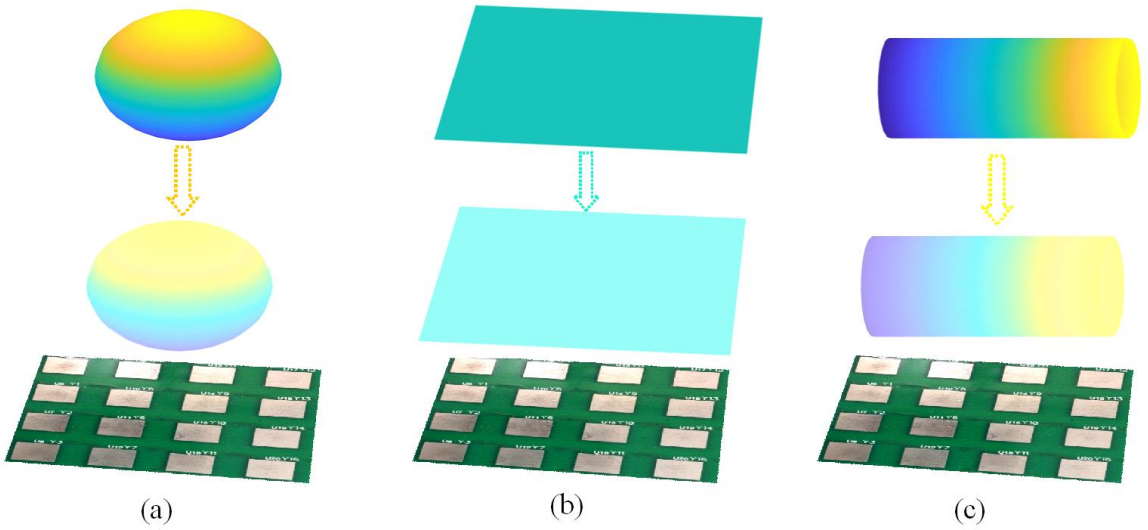


Figure 5.29. A conceptual demonstration of the experimental procedure. The pre-selected capacitors are measured when object is positioned at nine discrete distance values. (a) Sphere. (b) Plate. (c) Cylinder.

When designing the experimental procedure, the primary consideration is to recognize the shape while the object's vertical location is treated as one of the features. A conceptual measurement procedure is demonstrated in Figure 5.29. The KUKA robot is programmed to hold the object at nine different discrete distances (i.e. 3cm, 5cm, 6cm, 9cm, 10cm, 12cm, 15cm, 18cm, and 20cm) for two minutes. The first sweeping pattern is pre-selected by configuring the switch array through the FPGA, and at each distance, the eight generated mutual capacitors are measured in cycles during the 2-minute. This results in 100 times of repeating measurement for every capacitor. Every complete set of capacitive values can be fused to one eight-dimensional feature vector (i.e. {C1, C2, C3, C4, C5, C6, C7, C8}). Therefore, one-hundred feature vectors represent one independent data point are collected, and nine independent 8-dimensional data points with each stand for one separate position are obtained for the object. In this way, the data set (Dataset I) for classification is formed. It contains 27 independent data points with 9 for each class (i.e. plate, sphere, and cylinder) and each data point is made up of 100 eight-dimensional feature vectors.

The same procedure is repeated with the second sweeping pattern for all the objects so that a second data set (Dataset II) is acquired for the same three-shape classification purpose. The only difference between Dataset I and II is the dimension of

the feature vector, as the second sweeping pattern would lead to a measurement of nine mutual capacitors per cycle.

Before feeding the obtained data sets to the learning tools, all the raw capacitive values inside the data sets are linearly scaled to the range of [-1, +1]. This step would balance the effects between attributes in greater numeric range and smaller numeric range. In addition, it is able to reduce the computational burden of the learning tools. For both the data sets, 1800 feature vectors that represent 18 data points are used as inputs for training the statistical learning tools, and the rest 900 feature vectors that stand for the other 9 complete different data points are used as testing data for verifying the quality of the prediction model. A brief description of the characteristics of the two data sets is summarized in Table 5.11.

Table 5.11. Characteristics of the data sets

Dataset	Classes	Data points	Instances	Dimension	Training	Testing	Scale
I	3 shapes	27	2700	8 (6 capacitors between adjacent columns/rows and 2 between diagonal electrodes)	1800	900	[-1, +1]
II	3 shapes	27	2700	9 (9 capacitors between four neighbor electrodes and all the rest electrodes)	1800	900	[-1, +1]

Both datasets are fed to the three different machine learning tools (i.e. KNN, NN, and SVM) separately. Thus the learning performance can be compared in terms of the character of the dataset as well as the feature of the classifier.

The k-nearest neighbors methodology is the most straightforward tool in classifying data into multiple categories. Its working performance is highly dependent on the selection of the application parameter “k”. The impact brought by “k” is investigated by varying the value of “k” from 1 to 100 with the step length of 10. The training accuracy and the testing accuracy obtained from Dataset I and Dataset II are summarized in Table 5.12. In this scenario, KNN is providing perfect training quality, as with any value of “k” the training accuracy is 100%. Its performance in terms of testing is also perfect when using the eight-dimensional Dataset I. However, Dataset II that is experimentally obtained from the second electrode-connection sweeping pattern results in the testing accuracy up to 84.44%. The test accuracy then drops slightly to 84% when the value of “k” reaches 40.

The reason for this high performance is the size of the data sets are not large enough, and the boundary between different classes is relatively clear.

Table 5.12. KNN performance with respect to the value of "k"

Dataset		KNN									
		1	10	20	30	40	50	60	70	80	100
Training accuracy (%)	I	100	100	100	100	100	100	100	100	100	100
	II	100	100	100	100	100	100	100	100	100	100
Testing accuracy (%)	I	100	100	100	100	100	100	100	100	100	100
	II	84.44	84.44	84.44	84.11	84	84	84	84	84	84

Neural Network (NN) is another machine learning tool that uses a mathematical model for pattern recognition based on a connectionistic approach to computation. It is an adaptive system that changes its structure according to the information that flows through the whole network. The neural network involved in this project is performed on Matlab platform, and it contains multiple neurons arranged in layers. Backward propagation of errors, one of the most widely used ways to train a neural network, is used in our project. It is a supervised training scheme, which means it learns from labeled training data. For every input in the training dataset, the NN is activated and the output is observed and compared to the desired output (the label) which is known in advance. The error is propagated back and the model coefficients are adjusted accordingly until the output error is below a predetermined threshold.

As most functions can be approximated with only a single hidden layer [192], the number of hidden layer in this project is directly selected as one. Neurons, that receive the inputs and computes the corresponding outputs, are the basic computational units in a neural network. They perform computations and transfer information from inputs to the outputs, and the number of hidden neurons would have a significant effect on the performance of a neural network. In this study, the performance of the neural network is examined when varying the number of neurons from 1 to 5, and the acquired performance indicators are listed in Table 5.13.

As discussed in the previous section, the cross-entropy (CE) error, which depends on the relative errors of network output and is affected less by the outliners, is used and listed in the table as an indicator of the performance of the neural network. From the table,

it can be observed that the neural network is able to compute a correct output with only one hidden layer and small numbers of neurons. For both data sets, classifying new data into one of the three classes with high accuracy and confidence would not cost many computational resources of an NN.

Table 5.13. NN performance with respect to the number of neurons

		Neurons				
		1	2	3	4	5
Dataset I	Epochs	133	76	49	45	63
	Testing CE	2.32e-6	1.53e-7	2.48e-7	2.66e-7	4.07e-7
	Training accuracy	100%	100%	100%	100%	100%
	Testing accuracy	100%	100%	100%	100%	100%
Dataset II	Epochs	62	98	54	53	151
	Testing CE	0.0144	5.90e-7	1.63e-8	1.93e-8	1.08e-7
	Training accuracy	99.8%	100%	100%	100%	100%
	Testing accuracy	100%	100%	100%	100%	100%

The last classifier that has been examined is Support Vector Machine (SVM). The SVM is a discriminative classifier formally defined by a separating hyperplane. The same as the neural network, SVM is also a supervised learning method. Given labeled training data, the machine computes an optimal hyperplane, which is able to maximize the margin of the training data, that categorizes examples. LIBSVM that supports multiclass classification by using “one-against-one” (OAO) method is used on Matlab platform for the task of profile recognition.

The testing accuracy from LIBSVM is determined by two kernel function parameters: cost parameter (β) and gamma (γ). It is not known beforehand which combination of the two parameters will lead to the best result for a specific problem. Consequently, a parameter search procedure is done separately with both the data sets. Growing sequences of β and γ are tried for training the model, and the identified most efficient parameter pair is the one leads to the highest testing accuracy. The involved values of β and γ for the grid-search are the same: β/γ = 0.5, 1, 2, 3, 4, 5. From the previous learning results, classifying the experimental data into three shape categories

does not require much computational cost. So these relatively small values are selected. The classification accuracy with testing data as a function of the cost parameter and gamma summarized in Table 5.14.

Table 5.14. Testing Accuracy after grid-search for the parameters β and γ

β \ γ	0.5	1	2	3	4	5
	Dataset I					
0.5	66.67%	66.67%	66.67%	66.67%	82.33%	96.22%
1	66.67%	66.67%	82.33%	100%	100%	100%
2	66.67%	82.33%	100%	100%	100%	100%
3	66.67%	100%	100%	100%	100%	100%
4	82.33%	100%	100%	100%	100%	100%
5	82.33%	100%	100%	100%	100%	100%
	Dataset II					
0.5	66.67%	66.67%	66.67%	66.67%	66.67%	69.67%
1	66.67%	66.67%	66.67%	75.56%	82.67%	82.67%
2	66.67%	66.67%	82.67%	82.67%	82.67%	82.67%
3	66.67%	75.56%	82.67%	82.67%	82.67%	82.89%
4	66.67%	82.67%	82.67%	82.67%	82.67%	82.67%
5	69.67%	82.67%	82.67%	82.89%	82.67%	82.67%

The upper part of Table 5,14 represents the grid-search result from Dataset I in which samples are eight-dimensional. The testing accuracy starts from 66.67% when β and γ are both small, and then increases to 100% when either β or γ extends 2. That indicates the SVM is able to provide high-quality classification results with low computational cost. Whereas Dataset II that is composed of nine-dimensional examples would lead to lower testing accuracies. The SVM is getting a testing accuracy of 66.67% at low β and γ values, and the testing accuracy is growing gradually to 82.89% when β is 5 and γ equals to 3. After looking into the resulting classify probabilities, it can be found that the majority of testing errors are due to the confusions between the sphere and cylinder. As the definition of the cost parameter β, a larger value of β tends to consume more computational resources. For this specific task, Dataset I combined with a parameter pair value of (2, 2) could make the SVM a powerful tool for efficient classification results.

The classification performance of the three classifiers with the two different data sets is summarized and compared in Figure 5.30. In this figure, each bar represents the testing accuracy results from a combination of a certain classifier and one specific data set. It can be observed that the three-shape recognition purpose can be achieved with high accuracy with any one of the combination patterns. Though consuming the most computational resources, the neural network performs the best with both data sets. Whereas for the other two classifiers, eight-dimensional Dataset I tends to result in higher testing accuracies in this specific learning task. In addition, NN and SVM are able to give out probability results that will be useful for more complex classification purposes.

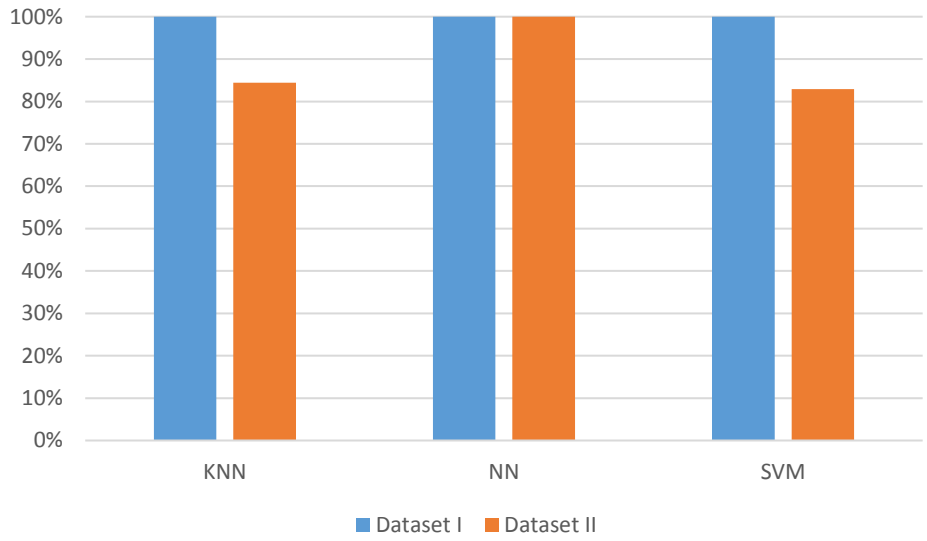


Figure 5.30. Comparison of different classifiers in terms of testing accuracy.

In summary, performing the machine learning procedure is able to classify the approaching object into the correct shape category with a high confidence. The acquired surface profile information is in turn used in selecting the proper regression model for the specific object for distance calculation. By adding the classification step, the error caused by an improper regression model can be eliminated. Based on the calculated error from the previous section, the overall distance measurement error below the distance of 5cm is reduced from 4.6cm to 1.1cm. This improvement makes the sensing system has the capability to work reliably in practice.

Chapter 6.

Conclusion and future work

Capacitive sensing systems are experiencing a strong market growth due to their contactless characters, high sensitivity, robust, and low cost. They have a significant impact on multiple industry segments including automotive, utilities, manufacturing, robotics, and safety in the process industry. Despite the considerable amount of research on capacitive collision alarming systems, the study on collision avoidance under nondeterministic conditions where robots have to sense and react to safety violations are quite limited. This technique challenge comes from that a capacitive sensor inherently suffers from it is sensitive to environmental interference and susceptible to errors due to variations in shape and size the approaching objects. In order to acquire a reliable human-robot interaction in manufacturing processes, it is desirable to design a multi-functional capacitive proximity sensing system that is able to provide a comprehensive and accurate monitor of the collaborative industrial workspace. The task is the subject of the research presented in this work.

6.1. Summary of the work

The objective of this study is to design a capacitive proximity sensing system with emphasis on optimizing its distance measurement accuracy. This goal is achieved by integrating three functionalities include distance measurement, surface profile recognition, and parallel motion tracking into the same sensing platform. Instead of just detecting the presence of an object, the quantitative distance is derived by the sensor for regulating the operation of a robot. Surface profile recognition in the proposed scenarios can be either distinguishing different objects according to their surface profiles regardless of their relative locations to the sensor or classifying an object into a few pre-set categories where each category stands for an object with a certain kind of shape located at a specific distance. In this way, the ambiguity issues in distance estimation can be alleviated. Moreover, an accessional functionality can be realized is the estimation of the parallel motion trajectory. The purpose of this feature is to reduce the possibility of shutting down the machine due to false alarms.

The basis of the whole design is the 4 × 4 matrix-shaped architecture that yielding 16 independent electrodes, which can be configured to generate multiple electric fields in order to mesh and monitor the close surroundings. Shunt detecting mode is applied in this design as it is more informative and flexible way of measurement and neither the transmitter nor the receiver is in contact with the object. A grounded backplane and an active shielding electrode matrix are added to shield out the interference, avoid the undesired detection from the back-side of the sensor, and reinforce the strength of the fringing electric field.

The sensing system is built up with discrete components: a four-layer printed circuit board (PCB), a field programmable gate array (FPGA), and a capacitance-to-digital (CDC) chipset. The top three layers of the PCB serve as carriers for sensing matrix, active shielding matrix, and backplane shielding. Two analog switch matrix chips and required supplementary electronic components are fabricated on the bottom side. The switch matrices are used to create desired connections within sensing and active shielding electrode matrix, respectively. The switch matrices are controlled by the FPGA which generates timing and switching signals. Ultimately, measured data is stored and processed on an external computer.

The raw measured capacitive data has to be fused and processed for desired information such as distance values and shape types. Data processing is further split into raw data processing (pre-processing) stage and high-level processing stage. Raw data processing is primarily intended to compensate for the non-ideal effects including high-frequency environmental noises and random walks. Statistical learning tools for understanding data is involved in high-level data processing. Quantitative regression models are built to seek out distances while different classifiers (i.e KNN, NN, and SVM) are employed to recognize the surface profiles.

The performance of the sensing modalities has been experimentally assessed. Experimental results obtained from lab equipment as well as with a real robot KUKA LBR iiwa are provided to demonstrate that the system is able to detect a metallic target whose shape is pre-known at distances of up to 18 cm with high resolutions. The three most frequently encountered object’s surface profiles can be recognized with high accuracy by all the three classifiers; the shape information together with some discrete distances can be classified with high confidence by the more advanced classification algorithm (i.e NN

and SVM). The purpose of this recognition task is to improve the regression model created for distance evaluation and guarantee the accuracy. The capability of tracking the parallel motion is verified by combining the capacitive responses from different electrode connection configurations.

6.2. Contributions

The presented project began in 2014 when the detection range of a commercially available capacitive proximity sensor was limited to a few centimeters [56]-[58]. Despite the considerable amount of research on capacitive proximity sensing and detection systems, the study on collision avoidance under nondeterministic conditions where robots had to sense and react to safety violations was quite limited. Most capacitance-based systems were only able to track small, predetermined objects only, such as a finger [201]. This work, in part, was a continuation and improvement over the previous research, which focused on the explorations on the capacitive responses from different materials and the object tracking [202], [203].

This project focused on the development of a multi-functional capacitive proximity sensor to improve worker safety during industrial human-robot interactions. The main contributions are extending the detection range to 20 cm and improving the distance estimation accuracy by recognizing the shape of the approaching object. The system relied on an array of electrodes and advanced signal processing procedures. Three sensing capabilities, namely distance measurement, surface profile recognition, and parallel motion tracking, have been implemented in a single platform. To the best of our knowledge, no similar solution has been proposed by other researchers.

The performance of the sensing modalities was experimentally assessed with lab equipment as well as on an industrial robot. The system can detect objects and classify their geometries at distances up to about 20 cm with high resolution. Recognizing the shape of the object, improved the regression models and reduced the close-distance measurement error by a factor of five compared to existing methods. The capability of tracking the parallel motion is demonstrated by combining the capacitive responses from different electrode connection configurations. The breakthroughs made through this work will make capacitive sensing a viable low-cost alternative to existing technologies for proximity sensing. Moreover, the application of the proposed solution is not restricted to

the field of robotics and the results can be applied to a wide range of other applications. As one example, there has been recent developments on the creating intelligent and responsive garments that use electrode arrays for human-computer interfaces [204], [205].

The presented research has resulted in the following publications. These papers focus on different aspects of the proposed sensing system and demonstrate the experimental results from different perspectives.

- F. Xia, F. Campi, and B. Bahreyni, “Tri-mode capacitive proximity detection towards improved safety in industrial robotics,” *IEEE Sensors Journal*, 18(12), 5058-5066.

- F. Xia, B. Bahreyni, and F. Campi, “Design of digital modules for capacitive proximity sensing system applications”, *Proceedings of the IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, 2016.

- F. Xia, B. Bahreyni, and F. Campi, “Multi-functional capacitive proximity sensing system for industrial safety applications,” *Proceedings of the IEEE Sensors Conference*, Oct. 2016.

Another journal paper based on the most recent experimental results, “Improving the accuracy of a proximity sensor based on target profile estimation”, is under preparation and will soon be submitted for publication.

6.3. Future work

The future work would be making the system more portable and more informative. More specifically, there are two aspects worth working on in the future: the design of the interface circuit and the optimization of the classification tools.

Currently, the sensing system is composed of several discrete components that are bulky. Moreover, as the desired capacitive responses come from the fringing electric field are generally small and weak. According to our observations during the experiments, the parasitic capacitances of the switch matrices worsen this situation as they might add

offsets to the sensing capacitive values and sometimes would even exceed the detected range of the capacitance-to-digital chip.

In order to achieve primary function of the sensing system, it is critical that the functional interaction and spatial integration among electronic, control, and signal processing technologies be accomplished in a synergistic way. The essential link between the transducer and the controller is provided by the interface circuit. Future study is needed in terms of designing low parasitic MOSFET switch arrays and a capacitance to digital converter with a wide dynamic range. In this way, the performance of the sensor in terms of detection range and accuracy can be enhanced and the whole system could be more integrated and portable. The ultimate goal is integrating the sensing system to a piece of working garment and design of the small-sized and low power-consumption interface circuit would benefit the integration.

Optimizing the classification tools is another aspect that can be considered as a future work. In this project, the high accuracy distance measurement is obtained with two separate steps: classification and regression. It is desirable to compress this procedure into one single step. Development of a one-step solution to estimate the quantified distance information can be followed in the future to simplify the sensing system. One potential way to achieve this goal is collecting more data by moving the objects in a more random way with the help of the KUKA robot and generating more complex electrode connection patterns by programming the FPGA.

References

[1] IFR, “International Federation of Robotics,” IFR International Federation of Robotics. [Online]. Available: https://ifr.org. [Accessed: 07-Jun-2018].

[2] G. Michalos, S. Makris, N. Papakostas, D. Mourtzis, and G. Chryssolouris, “Automotive assembly technologies review: challenges and outlook for a flexible and adaptive approach,” CIRP J. Manuf. Sci. Technol., vol. 2, no. 2, pp. 81–91, 2010.

[3] H. Bley, G. Reinhart, G. Seliger, M. Bernardi, and T. Korne, “Appropriate Human Involvement in Assembly and Disassembly,” CIRP Ann., vol. 53, no. 2, pp. 487–509, 2004.

[4] J. Krüger, T. K. Lien, and A. Verl, “Cooperation of human and machines in assembly lines,” CIRP Ann., vol. 58, no. 2, pp. 628–646, 2009.

[5] M. A. Goodrich and A. C. Schultz, “Human-Robot Interaction: A Survey,” Found. Trends® Hum.-Comput. Interact., vol. 1, no. 3, pp. 203–275, 2007.

[6] T. B. Sheridan, “Human–Robot Interaction: Status and Challenges,” Hum. Factors J. Hum. Factors Ergon. Soc., vol. 58, no. 4, pp. 525–532, Jun. 2016.

[7] P. Akella et al., “Cobots for the automobile assembly line,” in Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C), 1999, vol. 1, pp. 728–733 vol.1.

[8] Bicchi, Antonio, Michael A. Peshkin, and J. Edward Colgate. "Safety for physical human–robot interaction." Springer handbook of robotics. Springer, Berlin, Heidelberg, 2008, 1335-1348.

[9] R. D. Schraft, E. Helms, M. Hans, and S. Thiemermann, "Man-Machine-Interaction and co-operation for mobile and assisting robots." Proceedings of Fourth International ICSC Symposium on Engineering of Intelligent Systems (EIS 2004), Madeira, Portugal, 2004, Vol. 2.

[10] “OSHA Technical Manual (OTM) | Section IV: Chapter 4 - Industrial Robots and Robot System Safety | Occupational Safety and Health Administration.” [Online]. Available: https://www.osha.gov/dts/osta/otm/otm_iv/otm_iv_4.html#3. [Accessed: 08-Jun-2018].

[11] I. C. Manager, “Moving Parts of Machines.” [Online]. Available: http://www.iloencyclopaedia.org/part-viii-12633/safety-applications/94-58-safety-applications/moving-parts-of-machines. [Accessed: 23-Mar-2018].

[12] B. C. Jiang and C. A. Gainer, “A cause-and-effect analysis of robot accidents,” J. Occup. Accid., vol. 9, no. 1, pp. 27–45, Jun. 1987.

[13] J. H. Graham, J. F. Meagher, and S. J. Derby, “A Safety and Collision Avoidance System for Industrial Robots,” IEEE Trans. Ind. Appl., vol. IA-22, no. 1, pp. 195–203, Jan. 1986.

[14] “ISO 10218-1:2011 - Robots and robotic devices -- Safety requirements for industrial robots -- Part 1: Robots.” [Online]. Available: https://www.iso.org/standard/51330.html. [Accessed: 08-Jun-2018].

[15] “ISO/TS 15066:2016 - Robots and robotic devices -- Collaborative robots.” [Online]. Available: https://www.iso.org/standard/62996.html. [Accessed: 08-Jun-2018].

[16] L. M. Ssanderson, J. W. Collins, and J. D. McGlothlin, “Robot-related fatality involving a U.S. manufacturing plant employee: Case report and recommendations,” J. Occup. Accid., vol. 8, no. 1–2, pp. 13–23, Jun. 1986.

[17] S. Haddadin, A. Albu-Schäffer, and G. Hirzinger, “Requirements for Safe Robots: Measurements, Analysis and New Insights,” Int. J. Robot. Res., vol. 28, no. 11–12, pp. 1507–1527, Nov. 2009.

[18] P. A. Lasota, T. Fong, and J. A. Shah, “A Survey of Methods for Safe Human-Robot Interaction,” Found. Trends Robot., vol. 5, no. 3, pp. 261–349, 2017.

[19] S. Haddadin et al., “On making robots understand safety: Embedding injury knowledge into control,” Int. J. Robot. Res., vol. 31, no. 13, pp. 1578–1602, Nov. 2012.

[20] X. Broquère, D. Sidobre, and I. Herrera-Aguilar, “Soft Motion Trajectory Planner for Service Manipulator Robot,” in International Conference on Intelligent Robots and Systems, IROS 2008. IEEE/RSJ, Nice, France, 2008, pp. 2808–2813.

[21] M. Laffranchi, N. G. Tsagarakis, and D. G. Caldwell, “Safe human robot interaction via energy regulation control,” in 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009, pp. 35–41.

[22] C. Vogel, C. Walter, and N. Elkmann, “A projection-based sensor system for safe physical human-robot collaboration,” in 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2013, pp. 5359–5364.

[23] D. Fox, W. Burgard, and S. Thrun, “The dynamic window approach to collision avoidance,” IEEE Robot. Autom. Mag., vol. 4, no. 1, pp. 23–33, Mar. 1997.

[24] P. Saranrittichai, N. Niparnan, and A. Sudsang, “Robust local obstacle avoidance for mobile robot based on Dynamic Window approach,” in 2013 10th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, 2013, pp. 1–4.

[25] N. Najmaei, S. Lele, M. R. Kermani, and R. Sobot, “Human factors for robot safety assessment,” in 2010 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, 2010, pp. 539–544.

[26] G. B. Avanzini, N. M. Ceriani, A. M. Zanchettin, P. Rocco, and L. Bascetta, “Safety Control of Industrial Robots Based on a Distributed Distance Sensor,” IEEE Trans. Control Syst. Technol., vol. 22, no. 6, pp. 2127–2140, Nov. 2014.

[27] O. Khatib, “Real-time obstacle avoidance for manipulators and mobile robots,” in 1985 IEEE International Conference on Robotics and Automation Proceedings, 1985, vol. 2, pp. 500–505.

[28] P. A. Lasota, G. F. Rossano, and J. A. Shah, “Toward safe close-proximity human-robot interaction with standard industrial robots,” in 2014 IEEE

International Conference on Automation Science and Engineering (CASE), 2014, pp. 339–344.

[29] A. D. Luca and F. Flacco, “Integrated control for pHRI: Collision avoidance, detection, reaction and collaboration,” in 2012 4th IEEE RAS EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob), 2012, pp. 288–295.

[30] D. Kulić and E. Croft, “Pre-collision safety strategies for human-robot interaction,” Auton. Robots, vol. 22, no. 2, pp. 149–164, Jan. 2007.

[31] S. Golz, C. Osendorfer, and S. Haddadin, “Using tactile sensation for learning contact knowledge: Discriminate collision from physical interaction,” in 2015 IEEE International Conference on Robotics and Automation (ICRA), 2015, pp. 3788–3794.

[32] T. Y. Choi, B. S. Choi, and K. H. Seo, “Position and Compliance Control of a Pneumatic Muscle Actuated Manipulator for Enhanced Safety,” IEEE Trans. Control Syst. Technol., vol. 19, no. 4, pp. 832–842, Jul. 2011.

[33] A. D. Luca, F. Flacco, A. Bicchi, and R. Schiavi, “Nonlinear decoupled motion-stiffness control and collision detection/reaction for the VSA-II variable stiffness device,” in 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009, pp. 5487–5494.

[34] M. Geravand, F. Flacco, and A. D. Luca, “Human-robot physical interaction and collaboration using an industrial robot with a closed control architecture,” in 2013 IEEE International Conference on Robotics and Automation, 2013, pp. 4000–4007.

[35] S. Haddadin, A. Albu-Schaffer, A. D. Luca, and G. Hirzinger, “Collision detection and reaction: A contribution to safe physical Human-Robot Interaction,” in 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2008, pp. 3356–3363.

[36] A. D. Luca, A. Albu-Schaffer, S. Haddadin, and G. Hirzinger, “Collision Detection and Safe Reaction with the DLR-III Lightweight Manipulator Arm,” in 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2006, pp. 1623–1630.

[37] M. S. Erden and T. Tomiyama, “Human-Intent Detection and Physically Interactive Control of a Robot Without Force Sensors,” IEEE Trans. Robot., vol. 26, no. 2, pp. 370–382, Apr. 2010.

[38] E. M. P. Low, I. R. Manchester, and A. V. Savkin, “A biologically inspired method for vision-based docking of wheeled mobile robots,” Robot. Auton. Syst., vol. 55, no. 10, pp. 769–784, Oct. 2007.

[39] D. Henrich and T. Gecks, “Multi-camera collision detection between known and unknown objects,” in 2008 Second ACM/IEEE International Conference on Distributed Smart Cameras, 2008, pp. 1–10.

[40] R. Volpe and R. Ivlev, “A survey and experimental evaluation of proximity sensors for space robotics,” in Proceedings of the 1994 IEEE International Conference on Robotics and Automation, 1994, pp. 3466–3473 vol.4.

[41] “Banner Q4X Series Laser Distance Sensors Distributors.” [Online]. Available: http://www.clrwtr.com/Banner-Q4X-Series.htm. [Accessed: 26-Oct-2016].

[42] “MX1A and MX1B: Laser Displacement Detection Sensors.” [Online]. Available: https://www.idec.com/language/english/catalog/Sensors/MX1A_MX1BSeries.pdf. [Accessed: 26-Oct-2016].

[43] “Panasonic HG-C1030 Micro Laser Distance Sensor - 30mm - NPN.” [Online]. Available: https://www.sunxsensors.com/products/product/3218-panasonic-hg-c1030-cmos-laser-measurement-analog-distance-compact.html. [Accessed: 26-Oct-2016].

[44] “Digital Fiber Sensor: FX-301P Series.” [Online]. Available: https://www.sunxsensors.com/products/product/922-fx-301p.html. [Accessed: 26-Oct-2016].

[45] “SunX Digital Fiber Sensor: FX-411 Series.” [Online]. Available: https://www.sunxsensors.com/products/product/download/get.html?download_id=489. [Accessed: 14-Nov-2016].

[46] “DF-G3 Long Range Expert Dual DisplayFiber Amplifier.” [Online]. Available: http://info.bannerengineering.com/cs/groups/public/documents/literature/187435.pdf. [Accessed: 14-Nov-2016].

[47] “Sharp-GP2Y0A02YK0F: Distance Measuring Sensor Unit.” [Online]. Available: http://www.sharpsma.com/webfm_send/1487. [Accessed: 14-Nov-2016].

[48] “Sharp-GP2Y0A21YK: Optoelectronic Device.” [Online]. Available: http://www.sharpsma.com/webfm_send/1208. [Accessed: 14-Nov-2016].

[49] “Seed Technologu Inc Grove-80cm Infrared Proximity Sensor Model: SEN39046P.” [Online]. Available: https://www.terraelectronica.ru/%2Fds%2Fpdf%2FS%2FSEN39046P.pdf. [Accessed: 14-Nov-2016].

[50] “Ping Ultrasonic Distance Sensor: 28015.” [Online]. Available: https://www.parallax.com/sites/default/files/downloads/28015-PING-Sensor-Product-Guide-v2.0.pdf. [Accessed: 14-Nov-2016].

[51] “Sick Ultrasonic Sensors: UM30-212111.” [Online]. Available: https://www.sick.com/media/pdf/9/49/349/dataSheet_UM30-212111_6037661_en.pdf. [Accessed: 14-Nov-2016].

[52] “Compact Ultrasonic Sensor-E4E2.” [Online]. Available: https://www.ia.omron.com/data_pdf/cat/e4e2_ds_e_2_1_csm1726.pdf?id=504. [Accessed: 14-Nov-2016].

[53] “Sick Inductive Proximity Sensor-IME12-04BPSZCOK.” [Online]. Available: https://www.sick.com/media/pdf/2/02/502/dataSheet_IME12-04BPSZC0K_1040763_en.pdf. [Accessed: 14-Nov-2016].

[54] “Panasonic Cylindrical Inductive Proximity Sensor-GX-8MU.” [Online]. Available:

http://www3.panasonic.biz/ac/e/search_num/index.jsp?c=detail&part_no=GX-8MU. [Accessed: 27-Oct-2016].

[55] “Schneider Inductive Sensor-XS630B1MAL2.” [Online]. Available: http://www.alliedelec.com/m/d/1a6ae83ea4584c6ccb43656b64245c1d.pdf. [Accessed: 14-Nov-2016].

[56] “Capacitive Proximity Sensors-Smooth Body Models.” [Online]. Available: http://pdf.datasheet.directory/datasheets-1/eaton_corporation/E53KBL34T111E.pdf. [Accessed: 14-Nov-2016].

[57] “Autonics Electric Capacitive type proximity Sensor CR Series.” [Online]. Available: http://www.farnell.com/datasheets/1533393.pdf?_ga=1.227663261.1053298186.1449704776. [Accessed: 14-Nov-2016].

[58] “Sick Capacitive Proximity Sensor-CQ35-25NPP-KC1.” [Online]. Available: https://www.sick.com/media/pdf/7/67/267/dataSheet_CQ35-25NPP-KC1_6020479_en.pdf. [Accessed: 14-Nov-2016].

[59] S. Walker, K. Loewke, M. Fischer, C. Liu, and J. K. Salisbury, “An Optical Fiber Proximity Sensor for Haptic Exploration,” in Proceedings 2007 IEEE International Conference on Robotics and Automation, 2007, pp. 473–478.

[60] S. C. Huang, W. H. Hsu, P. C. P. Chao, and C. H. Tsai, “A New Active 3D Optical Proximity Sensor Array and its Readout Circuit,” IEEE Sens. J., vol. 14, no. 7, pp. 2185–2192, Jul. 2014.

[61] G. Plantier, C. Bes, and T. Bosch, “Behavioral Model of a Self-Mixing Laser Diode Sensor,” IEEE J. Quantum Electron., vol. 41, no. 9, pp. 1157–1167, Sep. 2005.

[62] J.-Y. Fang, C.-H. Tien, H.-P. D. Shieh, P. Herget, J. A. Bain, and T. E. Schlesinger, “Optical Feedback Height Control System Using Laser Diode Sensor for Near-Field Data Storage Applications,” J. Light. Technol., vol. 25, no. 12, pp. 3704–3709, Dec. 2007.

[63] J. H. Gao and L.-S. Peh, “A smartphone-based laser distance sensor for outdoor environments,” in 2016 IEEE International Conference on Robotics and Automation (ICRA), 2016, pp. 2922–2929.

[64] A. Rogalski, “History of infrared detectors,” Opto-Electron. Rev., vol. 20, no. 3, Jan. 2012.

[65] E. C. Chen et al., “Polymer Infrared Proximity Sensor Array,” IEEE Trans. Electron Devices, vol. 58, no. 4, pp. 1215–1220, Apr. 2011.

[66] L. Gitelman, S. Stolyarova, S. Bar-Lev, Z. Gutman, Y. Ochana, and Y. Nemirovsky, “CMOS-SOI-MEMS Transistor for Uncooled IR Imaging,” IEEE Trans. Electron Devices, vol. 56, no. 9, pp. 1935–1942, Sep. 2009.

[67] A. Rogalski, Infrared detectors, 2nd ed. Boca Raton: Taylor & Francis, 2011.

[68] C. Canali, G. D. Cicco, B. Morten, M. Prudenziati, and A. Taroni, “A Temperature Compensated Ultrasonic Sensor Operating in Air for Distance and Proximity Measurements,” IEEE Trans. Ind. Electron., vol. IE-29, no. 4, pp. 336–341, Nov. 1982.

[69] S. D. Min, D. J. Yoon, S. W. Yoon, Y. H. Yun, and M. Lee, “A study on a non-contacting respiration signal monitoring system using Doppler ultrasound,” Med. Biol. Eng. Comput., vol. 45, no. 11, pp. 1113–1119, Nov. 2007.

[70] T. Mohammad, Using Ultrasonic and Infrared Sensors for Distance Measurement, World Academy of Science, Engineering and Technology 51, 2009, pp. 293-299.

[71] P. A. Passeraub, G. Rey-Mennet, P. A. Besse, H. Lorenz, and R. S. Popovic, “Inductive proximity sensor with a flat coil and a new differential relaxation oscillator,” Sens. Actuators Phys., vol. 60, no. 1, pp. 122–126, May 1997.

[72] Y. Hamasaki and T. Ide, “A Multi-layer Eddy Current Micro Sensor For Non-destructive Inspection Of Small Diameter Pipes,” in The 8th International Conference on Solid-State Sensors and Actuators, 1995 and Eurosensors IX. Transducers ’95, 1995, vol. 2, pp. 136–139.

[73] H. T. Cheng, A. M. Chen, A. Razdan, and E. Buller, “Contactless gesture recognition system using proximity sensors,” in 2011 IEEE International Conference on Consumer Electronics (ICCE), 2011, pp. 149–150.

[74] Z. Chen and R. C. Luo, “Design and implementation of capacitive proximity sensor using microelectromechanical systems technology,” IEEE Trans. Ind. Electron., vol. 45, no. 6, pp. 886–894, Dec. 1998.

[75] B. E. Noltingk, “A novel proximity gauge,” J. Phys. [E], vol. 2, no. 4, p. 356, 1969.

[76] J. R. Smith, E. Garcia, R. Wistort, and G. Krishnamoorthy, “Electric field imaging pretouch for robotic graspers,” in 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2007, pp. 676–683.

[77] B. Mayton, L. LeGrand, and J. R. Smith, “An Electric Field Pretouch system for grasping and co-manipulation,” in 2010 IEEE International Conference on Robotics and Automation, 2010, pp. 831–838.

[78] A. Braun, H. Heggen, and R. Wichert, “CapFloor – A Flexible Capacitive Indoor Localization System,” in Evaluating AAL Systems Through Competitive Benchmarking. Indoor Localization and Tracking, Springer, Berlin, Heidelberg, 2011, pp. 26–35.

[79] M. Sousa, A. Techmer, A. Steinhage, C. Lauterbach, and P. Lukowicz, “Human tracking and identification using a sensitive floor and wearable accelerometers,” in 2013 IEEE International Conference on Pervasive Computing and Communications (PerCom), 2013, pp. 166–171.

[80] T. Grosse-Puppendahl et al., “Platypus: Indoor Localization and Identification through Sensing of Electric Potential Changes in Human Bodies,” 2016, pp. 17–30.

[81] A. Mujibiya and J. Rekimoto, “Mirage: exploring interaction modalities using off-body static electric field sensing,” 2013, pp. 211–220.

[82] V. Miika, K. Lasse, M. Jaakko, and V. Jukka, “Unobtrusive human height and posture recognition with a capacitive sensor,” J. Ambient Intell. Smart Environ., no. 4, pp. 305–332, 2011.

[83] R. Wimmer, “Capacitive Sensors for Whole Body Interaction,” in Whole Body Interaction, Springer, London, 2011, pp. 121–133.

[84] J. Cheng, O. Amft, G. Bahle, and P. Lukowicz, “Designing Sensitive Wearable Capacitive Sensors for Activity Recognition,” IEEE Sens. J., vol. 13, no. 10, pp. 3935–3947, Oct. 2013.

[85] K. Nomura et al., “A flexible proximity sensor formed by duplex screen/screen-offset printing and its application to non-contact detection of human breathing,” Sci. Rep., vol. 6, no. 1, Apr. 2016.

[86] T. G. Zimmerman, J. R. Smith, J. A. Paradiso, D. Allport, and N. Gershenfeld, “Applying electric field sensing to human-computer interfaces,” 1995, pp. 280–287.

[87] J. R. Smith, “Field mice: Extracting hand geometry from electric field measurements,” IBM Syst. J., vol. 35, no. 3.4, pp. 587–608, 1996.

[88] J. Smith, T. White, C. Dodge, J. Paradiso, N. Gershenfeld, and D. Allport, “Electric field sensing for graphical interfaces,” IEEE Comput. Graph. Appl., vol. 18, no. 3, pp. 54–60, May 1998.

[89] Smith, Joshua Reynolds. "Electric field imaging." PhD diss., Massachusetts Institute of Technology, 1999.

[90] Braun, Andreas, and Pascal Hamisu. "Designing a multi-purpose capacitive proximity sensing input device." Proceedings of the 4th International Conference on PErvasive Technologies Related to Assistive Environments. ACM, 2011.

[91] Braun, Andreas, Stephan Neumann, Sönke Schmidt, Reiner Wichert, and Arjan Kuijper. "Towards interactive car interiors: the active armrest." In Proceedings of the 8th Nordic Conference on Human-Computer Interaction: Fun, Fast, Foundational, 2014, pp. 911-914.

[92] Grosse-Puppendahl, Tobias, Sebastian Beck, and Daniel Wilbers. "Rainbowfish: visual feedback on gesture-recognizing surfaces." In CHI'14 Extended Abstracts on Human Factors in Computing Systems, 2014, pp. 427-430.

[93] Rogers, Simon, John Williamson, Craig Stewart, and Roderick Murray-Smith. "FingerCloud: uncertainty and autonomy handover incapacitive sensing." In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 2010, pp. 577-580.

[94] S. Rogers, J. Williamson, C. Stewart, and R. Murray-Smith, “AnglePose: robust, precise capacitive touch tracking via 3d orientation estimation,” 2011, p. 2575.

[95] T. Grosse-Puppendahl, A. Braun, F. Kamieth, and A. Kuijper, “Swiss-cheese extended: an object recognition method for ubiquitous interfaces based on capacitive proximity sensing,” in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 2013, pp. 1401–1410.

[96] M. Le Goc, S. Taylor, S. Izadi, and C. Keskin, “A low-cost transparent electric field sensor for 3d interaction on mobile devices,” in Proceedings of

the 32nd annual ACM conference on Human factors in computing systems, 2014, pp. 3167–3170.

[97] Kim, Yuhwan, Seungchul Lee, Inseok Hwang, Hyunho Ro, Youngki Lee, Miri Moon, and Junehwa Song. "High5: promoting interpersonal hand-to-hand touch for vibrant workplace with electrodermal sensor watches." In Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing, pp. 15-19. ACM, 2014.

[98] G. Singh, A. Nelson, R. Robucci, C. Patel, and N. Banerjee, “Inviz: Low-power personalized gesture recognition using wearable textile capacitive sensor arrays,” in 2015 IEEE International Conference on Pervasive Computing and Communications (PerCom), 2015, pp. 198–206.

[99] T. A. Große-Puppendahl, A. Marinc, and A. Braun, “Classification of User Postures with Capacitive Proximity Sensors in AAL-Environments,” in Ambient Intelligence, 2011, pp. 314–323.

[100] G. Laput, C. Yang, R. Xiao, A. Sample, and C. Harrison, “EM-Sense: Touch Recognition of Uninstrumented, Electrical and Electromechanical Objects,” Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology, 2015, pp. 157–166.

[101] C. Harrison, M. Sato, and I. Poupyrev, “Capacitive fingerprinting: exploring user differentiation by sensing electrical properties of the human body,” Proceedings of the 25th annual ACM symposium on User interface software and technology, 2012, pp. 537-544.

[102] S. Han and J. Park, “Grip-Ball: A spherical multi-touch interface for interacting with virtual worlds,” in 2013 IEEE International Conference on Consumer Electronics (ICCE), 2013, pp. 600–601.

[103] Kim, Kee-Eung, et al. "Hand grip pattern recognition for mobile user interfaces." Proceedings of the National Conference on Artificial Intelligence, 2006, p. 1789.

[104] J. R. Smith, “Field mice: Extracting hand geometry from electric field measurements,” IBM Syst. J., vol. 35, no. 3.4, pp. 587–608, 1996.

[105] L. K. Baxter, Capacitive sensors: design and applications. New York: IEEE Press, 1997.

[106] J. R. Smith, “Electric field imaging,” PhD Thesis, Massachusetts Institute of Technology, 1999.

[107] T. G. Zimmerman, J. R. Smith, J. A. Paradiso, D. Allport, and N. Gershenfeld, “Applying electric field sensing to human-computer interfaces,” in Proceedings of the SIGCHI conference on Human factors in computing systems, 1995, pp. 280–287.

[108] R. Schinzinger and P. A. Laura, Conformal mapping: methods and applications. Courier Corporation, 2003.

[109] O. G. Vendik, S. P. Zubko, and M. A. Nikol’skii, “Modeling and calculation of the capacitance of a planar capacitor containing a ferroelectric thin film,” Tech. Phys., vol. 44, no. 4, pp. 349–355, 1999.

[110] Z. Chen and R. C. Luo, “Design and implementation of capacitive proximity sensor using microelectromechanical systems technology,” IEEE Trans. Ind. Electron., vol. 45, no. 6, pp. 886–894, Dec. 1998.

[111] W. J. Getsinger, “A coupled strip-line configuration using printed-circuit construction that allows very close coupling,” IRE Trans. Microw. Theory Tech., vol. 9, no. 6, pp. 535–544, 1961.

[112] J. L. Novak and J. J. Wiczer, “A high-resolution capacitative imaging sensor for manufacturing applications,” in Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on, 1991, pp. 2071–2078.

[113] S. B. Cohn, “Shielded Coupled-Strip Transmission Line,” IRE Trans. Microw. Theory Tech., vol. 3, no. 5, pp. 29–38, Oct. 1955.

[114] Ø. Isaksen, “A review of reconstruction techniques for capacitance tomography,” Meas. Sci. Technol., vol. 7, no. 3, pp. 325–337, Mar. 1996.

[115] H. Wang, W. Yin, W. Q. Yang, and M. S. Beck, “Optimum design of segmented capacitance sensing array for multi-phase interface measurement,” Meas. Sci. Technol., vol. 7, no. 1, pp. 79–86, Jan. 1996.

[116] X. Li, G. de Jong, and G. C. M. Meijer, “The application of the capacitor’s physics to optimize capacitive angular-position sensors,” IEEE Trans. Instrum. Meas., vol. 46, no. 1, pp. 8–14, Feb. 1997.

[117] A. J. Jaworski and G. T. Bolton, “The design of an electrical capacitance tomography sensor for use with media of high dielectric permittivity,” Meas. Sci. Technol., vol. 11, no. 6, pp. 743–757, Jun. 2000.

[118] R. Igreja and C. J. Dias, “Analytical evaluation of the interdigital electrodes capacitance for a multi-layered structure,” Sens. Actuators Phys., vol. 112, no. 2–3, pp. 291–301, May 2004.

[119] S. S. Gevorgian, T. Martinsson, P. L. J. Linner, and E. L. Kollberg, “CAD models for multilayered substrate interdigital capacitors,” IEEE Trans. Microw. Theory Tech., vol. 44, no. 6, pp. 896–904, Jun. 1996.

[120] X. Li, G. de Jong, and G. C. M. Meijer, “The influence of electric-field bending on the nonlinearity of capacitive sensors,” IEEE Trans. Instrum. Meas., vol. 49, no. 2, pp. 256–259, Apr. 2000.

[121] “COMSOL Multiphysics® Modeling Software.” [Online]. Available: https://www.comsol.com/. [Accessed: 19-Mar-2018].

[122] X. B. Li, S. D. Larson, A. S. Zyuzin, and A. V. Mamishev, “Design of multichannel fringing electric field sensors for imaging. Part I. General design principles,” in Conference Record of the 2004 IEEE International Symposium on Electrical Insulation, 2004, pp. 406–409.

[123] X. B. Li, S. D. Larson, A. S. Zyuzin, and A. V. Mamishev, “Design principles for multichannel fringing electric field sensors,” IEEE Sens. J., vol. 6, no. 2, pp. 434–440, Apr. 2006.

[124] R. Pallás-Areny and J. G. Webster, Sensors and signal conditioning, 2nd ed. New York: Wiley, 2001.

[125] C. S. Helrich, The classical theory of fields: electromagnetism. Heidelberg ; New York: Springer, 2012.

[126] W. J. Watterson, R. D. Montgomery, and R. P. Taylor, “Fractal Electrodes as a Generic Interface for Stimulating Neurons,” Sci. Rep., vol. 7, no. 1, Dec. 2017.

[127] M. S. Abdul Rahman, S. C. Mukhopadhyay, and P.-L. Yu, Novel Sensors for Food Inspection: Modelling, Fabrication and Experimentation, vol. 10. Cham: Springer International Publishing, 2014.

[128] A. V. Mamishev, K. Sundara-Rajan, F. Yang, Y. Du, and M. Zahn, “Interdigital sensors and transducers,” Proc. IEEE, vol. 92, no. 5, pp. 808–845, May 2004.

[129] F. W. Smith, H. J. Neuhaus, S. D. Senturia, Z. Feit, D. R. Day, and T. J. Lewis, “Electrical conduction in polyimide between 20 and 350° C,” J. Electron. Mater., vol. 16, no. 1, pp. 93–106, Jan. 1987.

[130] Huang, Yunzhi, Zheng Zhan, and Nicola Bowler. "Optimization of the coplanar interdigital capacitive sensor." AIP Conference Proceedings. Vol. 1806. No. 1, 2017.

[131] H. K. Lee, S. I. Chang, and E. Yoon, “Dual-Mode Capacitive Proximity Sensor for Robot Application: Implementation of Tactile and Proximity Sensing Capability on a Single Polymer Platform Using Shared Electrodes,” IEEE Sens. J., vol. 9, no. 12, pp. 1748–1755, Dec. 2009.

[132] S. Aksoy and R. M. Haralick, “Feature normalization and likelihood-based similarity measures for image retrieval,” Pattern Recognit. Lett., vol. 22, no. 5, pp. 563–582, Apr. 2001.

[133] L. K. Baxter, Capacitive sensors: design and applications. New York: IEEE Press, 1997.

[134] Macleod, Peter. A review of flexible circuit technology and its applications. PRIME Faraday Partnership, 2002.

[135] Wong, William S., and Alberto Salleo, eds. Flexible electronics: materials and applications. Vol. 11. Springer Science & Business Media, 2009.

[136] J. Cheng, O. Amft, G. Bahle, and P. Lukowicz, “Designing Sensitive Wearable Capacitive Sensors for Activity Recognition,” IEEE Sens. J., vol. 13, no. 10, pp. 3935–3947, Oct. 2013.

[137] Pethig, R. "Dielectric properties of body tissues." Clinical Physics and Physiological Measurement 8.4A, 1987, p. 5.

[138] Witten, Ian H., et al. Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann, 2016.

[139] H. Mannila, “Data mining: machine learning, statistics, and databases,” in Proceedings of 8th International Conference on Scientific and Statistical Data Base Management, 1996, pp. 2–9.

[140] Rahm, Erhard, and Hong Hai Do. "Data cleaning: Problems and current approaches." IEEE Data Eng. Bull. 23.4, 2000, pp. 3-13.

[141] “Data Collection Methods and Data Preprocessing Techniques For Healthcare Data Using Data Mining,” vol. 8, no. 6, p. 6, 2017.

[142] “The-Morgan-Kaufmann-Series-in-Data-Management-Systems-Jiawei-Han-Micheline-Kamber-Jian-Pei-Data-Mining.-Concepts-and-Techniques-3rd-Edition-Morgan-Kaufmann-2011.pdf.” .

[143] Richardson, Mark. "Principal component analysis." URL: http://people. maths. ox. ac. uk/richardsonm/SignalProcPCA. pdf (last access: 3.5. 2013). Aleš Hladnik Dr., Ass. Prof., Chair of Information and Graphic Arts Technology, Faculty of Natural Sciences and Engineering, University of Ljubljana, Slovenia ales. hladnik@ ntf. uni-lj. si 6, 2009, p. 16.

[144] H. Abdi and L. J. Williams, “Principal component analysis,” Wiley Interdiscip. Rev. Comput. Stat., vol. 2, no. 4, pp. 433–459, Jul. 2010.

[145] Jolliffe, Ian. "Principal component analysis." International encyclopedia of statistical science. Springer, Berlin, Heidelberg, 2011, pp. 1094-1096.

[146] R. Bro and A. K. Smilde, “Principal component analysis,” Anal Methods, vol. 6, no. 9, pp. 2812–2831, 2014.

[147] G. James, D. Witten, T. Hastie, and R. Tibshirani, An Introduction to Statistical Learning, vol. 103. New York, NY: Springer New York, 2013.

[148] W. Mendenhall and T. Sincich, A second course in statistics: regression analysis, 7. ed., international ed. Boston, Mass.: Prentice Hall, 2012.

[149] J. O. Rawlings, S. G. Pantula, and D. A. Dickey, Applied regression analysis: a research tool, 2nd ed. New York: Springer, 1998.

[150] Montgomery, Douglas C., George C. Runger, and Norma F. Hubele. Engineering statistics. John Wiley & Sons, 2009.

[151] F. Pereira, T. Mitchell, and M. Botvinick, “Machine learning classifiers and fMRI: A tutorial overview,” NeuroImage, vol. 45, no. 1, pp. S199–S209, Mar. 2009.

[152] Ayodele, Taiwo Oladipupo. "Introduction to machine learning." New Advances in Machine Learning. InTech, 2010.

[153] Abu-Mostafa, Yaser S., Malik Magdon-Ismail, and Hsuan-Tien Lin. Learning from data. Vol. 4. New York, NY, USA:: AMLBook, 2012.

[154] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft, “When Is ‘Nearest Neighbor’ Meaningful?,” in Database Theory — ICDT’99, vol. 1540, C. Beeri and P. Buneman, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 217–235.

[155] G. James, D. Witten, T. Hastie, and R. Tibshirani, Eds., An introduction to statistical learning: with applications in R. New York: Springer, 2013.

[156] Robert, Christian. "Machine learning, a probabilistic perspective.", the MIT press, 2014.

[157] M. A. Alsheikh, S. Lin, D. Niyato, and H. P. Tan, “Machine Learning in Wireless Sensor Networks: Algorithms, Strategies, and Applications,” IEEE Commun. Surv. Tutor., vol. 16, no. 4, pp. 1996–2018, Fourthquarter 2014.

[158] T. Cover and P. Hart, “Nearest neighbor pattern classification,” IEEE Trans. Inf. Theory, vol. 13, no. 1, pp. 21–27, Jan. 1967.

[159] McCulloch, Warren S., and Walter Pitts. "A logical calculus of the ideas immanent in nervous activity." The bulletin of mathematical biophysics 5.4, 1943, pp. 115-133.

[160] M. Gholami, N. Cai, and R. W. Brennan, “An artificial neural network approach to the problem of wireless sensors network localization,” Robot. Comput.-Integr. Manuf., vol. 29, no. 1, pp. 96–109, Feb. 2013.

[161] S. S. Haykin and S. S. Haykin, Neural networks and learning machines, 3rd ed. New York: Prentice Hall, 2009.

[162] A. Jabbari, R. Jedermann, R. Muthuraman, and W. Lang, “Application of Neurocomputing for Data Approximation and Classification in Wireless Sensor Networks,” Sensors, vol. 9, no. 4, pp. 3056–3077, Apr. 2009.

[163] P. Coulibaly, F. Anctil, and B. Bobée, “Daily reservoir inflow forecasting using artificial neural networks with stopped training approach,” J. Hydrol., vol. 230, no. 3–4, pp. 244–257, May 2000.

[164] B. K. Bose, “Neural Network Applications in Power Electronics and Motor Drives mdash;An Introduction and Perspective,” IEEE Trans. Ind. Electron., vol. 54, no. 1, pp. 14–33, Feb. 2007.

[165] C. M. Bishop, Pattern recognition and machine learning. New York: Springer, 2006.

[166] J. L. McCLELLAND, D. E. Rumelhart, and G. E. Hinton, “The Appeal of Parallel Distributed Processing,” in Readings in Cognitive Science, Elsevier, 1988, pp. 52–72.

[167] K. Hornik, M. Stinchcombe, and H. White, “Multilayer feedforward networks are universal approximators,” Neural Netw., vol. 2, no. 5, pp. 359–366, Jan. 1989.

[168] C. Cortes and V. Vapnik, “Support-vector networks,” Mach. Learn., vol. 20, no. 3, pp. 273–297, Sep. 1995.

[169] K. P. Murphy, Machine learning: a probabilistic perspective. Cambridge, MA: MIT Press, 2012.

[170] C. Schuldt, I. Laptev, and B. Caputo, “Recognizing human actions: a local SVM approach,” in Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004., 2004, vol. 3, pp. 32-36 Vol.3.

[171] L. Zhou, L. Wang, L. Liu, P. Ogunbona, and D. Shen, “Support vector machines for neuroimage analysis: Interpretation from discrimination,” Support Vector Mach. Appl., pp. 191–220, Nov. 2014.

[172] N. Cristianini, An introduction to support vector machines: and other kernel-based learning methods / Nello Cristianini and John Shawe-Taylor. Cambridge, U.K. ; New York: Cambridge University Press, 2012.

[173] Duda, Richard O., Peter E. Hart, and David G. Stork. Pattern classification. John Wiley & Sons, 2012.

[174] Wang, Zhe, and Xiangyang Xue. "Multi-class support vector machine." Support Vector Machines Applications. Springer, Cham, 2014, pp. 23-48.

[175] C.-W. Hsu and C.-J. Lin, “A comparison of methods for multiclass support vector machines,” IEEE Trans. Neural Netw., vol. 13, no. 2, pp. 415–425, 2002.

[176] S. Arlot and A. Celisse, “A survey of cross-validation procedures for model selection,” Stat. Surv., vol. 4, no. 0, pp. 40–79, 2010.

[177] Kohavi, Ron. "A study of cross-validation and bootstrap for accuracy estimation and model selection." Ijcai. Vol. 14. No. 2, 1995.

[178] Analog Devices, "16 × 16 Crosspoint Switch Arrar," AD75019 datasheet, 2013.

[179] E. Christen and K. Bakalar, “VHDL-AMS-a hardware description language for analog and mixed-signal applications,” IEEE Trans. Circuits Syst. II Analog Digit. Signal Process., vol. 46, no. 10, pp. 1263–1272, Oct. 1999.

[180] R. Lipsett, C. F. Schaefer, and C. Ussery, VHDL: Hardware Description and Design. Springer Science & Business Media, 2012.

[181] S. Muhlbacher-Karrer, M. Brandstotter, D. Schett, and H. Zangl, “Contactless Control of a Kinematically Redundant Serial Manipulator Using Tomographic Sensors,” IEEE Robot. Autom. Lett., vol. 2, no. 2, pp. 562–569, Apr. 2017.

[182] Analog Devices, "24-Bit Capacitance-to-Digital Converter with Temperature Sensor," AD7745_7746 datasheet, 2005.

[183] Smith Jr, Philip T. "Analysis and application of capacitive displacement sensors to curved surfaces." 2003.

[184] A. Hoffmann, A. Poeppel, A. Schierl, and W. Reif, “Environment-aware proximity detection with capacitive sensors for human-robot-interaction,” 2016, pp. 145–150.

[185] A. Braun, R. Wichert, A. Kuijper, and D. W. Fellner, “Capacitive proximity sensing in smart environments,” J. Ambient Intell. Smart Environ., vol. 7, no. 4, pp. 483–510, Jan. 2015.

[186] N. L. Leech and A. J. Onwuegbuzie, “An array of qualitative data analysis tools: A call for data analysis triangulation.,” Sch. Psychol. Q., vol. 22, no. 4, p. 557, 2007.

[187] T. Grosse-Puppendahl, Y. Berghoefer, A. Braun, R. Wimmer, and A. Kuijper, “OpenCapSense: A rapid prototyping toolkit for pervasive interaction using capacitive sensing,” in 2013 IEEE International Conference on Pervasive Computing and Communications (PerCom), 2013, pp. 152–159.

[188] Smith, Lindsay I. A tutorial on principal components analysis. [Online], Available: https://ourarchive.otago.ac.nz/bitstream/handle/10523/7534/OUCS-2002-12.pdf?sequence=1. [Accessed: 2002].

[189] Q. Yu et al., “Optimal Pruned K-Nearest Neighbors: OP-KNN Application to Financial Modeling,” in 2008 Eighth International Conference on Hybrid Intelligent Systems, 2008, pp. 764–769.

[190] Greenacre, Michael. Correspondence analysis in practice. Chapman and Hall/CRC, 2017.

[191] Agatonovic-Kustrin, S., and R. Beresford. "Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research." Journal of pharmaceutical and biomedical analysis 22.5, 2000, pp. 717-727.

[192] B. D. Ripley, Pattern Recognition and Neural Networks. Cambridge University Press, 2007.

[193] Bishop, Chris, and Christopher M. Bishop. Neural networks for pattern recognition. Oxford university press, 1995.

[194] M. Yi-de, L. Qing, and Q. Zhi-bai, “Automated image segmentation using improved PCNN model based on cross-entropy,” in Proceedings of 2004 International Symposium on Intelligent Multimedia, Video and Speech Processing, 2004., 2004, pp. 743–746.

[195] Feng, Ziyong, Zenghui Sun, and Lianwen Jin. "Learning deep neural network using max-margin minimum classification error." Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on. IEEE, 2016, pp.2677-2681

[196] Mikolov, Tomáš, et al. "Recurrent neural network based language model." Eleventh Annual Conference of the International Speech Communication Association. 2010.

[197] B. Crawford, K. Miller, P. Shenoy, and R. Rao, “Real-time classification of electromyographic signals for robotic control,” in AAAI, 2005, vol. 5, pp. 523–528.

[198] H. Huang, F. Zhang, L. J. Hargrove, Z. Dou, D. R. Rogers, and K. B. Englehart, “Continuous Locomotion-Mode Identification for Prosthetic Legs Based on Neuromuscular; Mechanical Fusion,” IEEE Trans. Biomed. Eng., vol. 58, no. 10, pp. 2867–2875, Oct. 2011.

[199] C.-C. Chang and C.-J. Lin, “LIBSVM: a library for support vector machines,” ACM Trans. Intell. Syst. Technol. TIST, vol. 2, no. 3, p. 27, 2011.

[200] KUKA Robotics, “Spez_LBR_iiwa”, datasheet, [Online]. Available: https://www.kuka.com//media/kukadownloads/imported/.../spez_lbr_iiwa_en.pdf. [Accessed: 23-May-2016].

[201] Rosenberg I, Perlin K. The UnMousePad - An interpolating multi-touch force-sensing input pad. ACM Transactions on Graphics. 2009;65: 1-9.

[202] F. Aezinia, B. Bahreyni, M. Alexiuk, M. Fallah-Rad, and Labros Petropoulos, “A capacitive proximity sensor for enhanced clinical workflow with intra-operative MRI”, The 4th Annual Intraoperative Imaging Society Meeting, St. Louis, USA, Feb 2013.

[203] F. Aezinia, Y. Wang, and B. Bahreyni, “Three dimensional touchless tracking of objects using integrated capacitive sensors,” IEEE Trans.Consum. Electron., vol. 58, no. 3, pp. 886–890, Aug. 2012.

[204] Google Jacquard, [Online]. Available: https://atap.google.com/jacquard. [Accessed: 25-Sep-2017].

[205] Roh J, Mann Y, Freed A,Wessel D. Robust and reliable fabric, piezoresistive multitouch sensing surfaces for musical controllers. Proceedings of the International Conference on New Interfaces for Musical Expression, pp. 393-398. 2011.