Lebanese Official University and University of Genoa in Collaboration with Instituto Italiano Di Tecnologia (IIT)

Doctoral Thesis

Event Driven Tactile Sensors for Artificial Devices

Author:
Ali Abou Khalil

Supervisors:
Prof. Hussein Chible
Prof. Ali Hamie
Dr. Chiara Bartolozzi
Prof. Maurizio Valle

A thesis submitted in fulfillment of the requirements for the degree of Doctor of Philosophy

Event-Driven Perception for Robotics- iCub Facility
Istituto Italiano di Tecnologia (iit), Genoa

Gratitude

Words in some cases are feeble to give a person his right. However, in some cases, they are the only tool to reflect one's feelings toward others. To Professor Hussein Chible, a great thank and tribute to him for his way of cooperation, guidance and humble personality. I really enjoyed being a student working with him in my PhD thesis. I felt very comfortable in almost everything that joined us in our way to finish this work.

To Dr. Bartolozzi, I would like to thank her for her help, care and guidance in finishing my PhD. I remember the comments in strengthening the publications before the submission. In addition, I do not forget her role in providing me with good types of equipment and setups. I will not forget her permission to me to join the training phase that paved the way to me for a stable future and lifestyle.

I would like to thank Prof. Valle for his support after my arrival to Italy and his comments and advices during my PhD.

To Professor Ali Hamie, I thank him for his guidance especially in the period I spent in Lebanon. I benefited from his comments during discussions and reviewing the publications. In addition, he gave me valuable advises that I can never miss and the funding that he provided me with for the sake of research.

Finally, to the source of joy and optimism in my life, to my beloved father, sisters, and brother whose encouragement, cooperation and emotional support paved my way to the fruitful ending.

Abstract

Present day robots are, to some extent, able to deal with high complexity and variability of real world environment. Their cognitive capabilities can be further enhanced, if they physically interact and explore the real world objects. For this, the need for efficient tactile sensors is growing day after day in such a way are becoming more and more part of daily life devices specially in robotic applications for manipulation and safe interaction with the environment.

In this thesis, we highlight the importance of touch sensing in humans and robots. Inspired by the biological systems, in the first part, we merge between neuromorphic engineering and CMOS technology where the former is a field of science that replicates what is biologically (neurons of the nervous system) inside humans into the circuit level. We explain the operation and then characterize different sensor circuits through simulation and experiment to propose finally new prototypes based on the achieved results.

In the second part, we present a machine learning technique for detecting the direction and orientation of a sliding tip over a complete skin patch of the iCub robot. Through learning and on line testing, the algorithm classifies different trajectories across the skin patch. Through this part, we show the results of the considered algorithm with future perspective to extend the work.

Contents

List of Figures

List of Tables

Abbreviations

SiN	**Si**licon **N**euron
ED	**E**vent **D**riven
POSFET	**P**iezoelectric **O**xide **S**emiconductor **F**ield **E**ffect **T**ransistor
WROTA	**W**ide **R**ange **O**perational **T**ransconductance **A**mplifier
AER	**A**ddress **E**vent **R**epresentation
LIF	**L**eaky **I**ntegrate and **F**ire
$\mathbf{V}_{POS}$	**P**OSFET **O**utput Voltage
ML	**M**achine **L**earning
TM	**T**ransition **M**atrix
DM	**D**istance **M**atrix
PGM	**P**robabilistic **G**raphical **M**odel
PF	**P**article **F**ilter
PDF	**P**robability **D**istribution and **F**unction

Chapter 1

Introduction

A sense is a physiological capacity of organisms that provides data for perception. It is paramount for agents (both biological and artificial) to gather information about their environment and act accordingly.

The integument or skin is the largest organ of the body, making up 12-16% of bodyweight, with a surface area of 1.5-2 m^2 [3]. Consequently, the skin is of extreme significance for humans. They use touch not only to explore but also to feel and cohabit with the surrounding, objects, people and also with themselves. It has also several functions and roles. It gives the brain a wealth of information about the environment around to analyze and interpret to take suitable actions. Moreover, it forms a physical barrier to the environment, allowing and limiting the inward and outward passage of water, electrolytes and various substances while protecting against micro-organisms, ultraviolet radiation, toxic agents and mechanical insults. It also provides useful information for stable grasping of objects, accurate and stable movements.

Mechanoreceptors are sensory receptors that respond to mechanical pressure or distortion. They encode the spatio-temporal tactile information by generating trains of action potentials (voltage pulses generated when the stimulus is greater than a certain threshold [4]).

Mechanoreceptors are classified into three different groups depending mainly on the temporal dynamics of their response (see **Fig. 1.1**):

1. **Slow adapting**: Ruffini Endings, Merkel's Cells, and Tactile Disks are examples of slow adapting mechanoreceptors. These receptors are generally

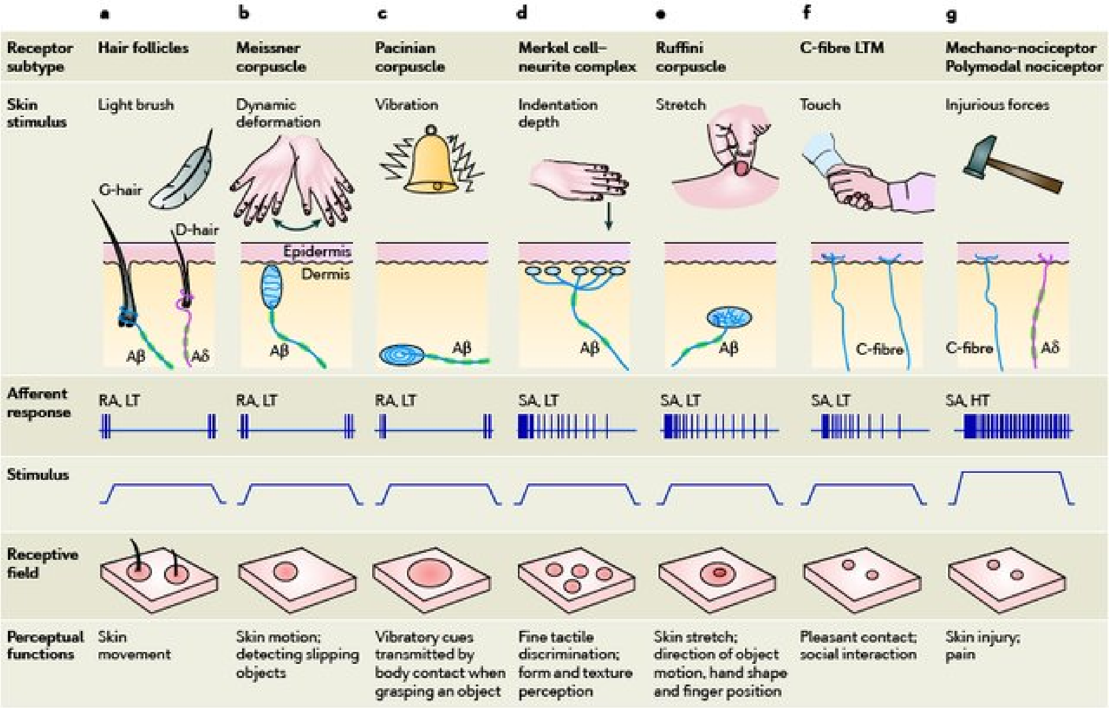


FIGURE 1.1: Mechanoreceptors are a diverse class of sensory endings.

located near the surface of the skin and are responsible for much of the static perceptive capabilities. The adaptation time scale for these cells can be from 10 to more than 100 seconds. They could be differentiated also for structural complexity.

2. **Moderate adapting**: Free nerve endings and hair follicle receptors are good examples of mechanoreceptors with moderate adaptation rates. These receptors can be located near the surface of the skin, and adapt to changes on periods of order 1 second.

3. **Rapid adapting**: Pacinian Corpuscles and Meissner corpuscle are rapidly adapting mechanoreceptors and are often the most sensitive cells to very small changes in the stimulus. These rapidly adapting cells return to a normal rate of pulses in less than 0.1 seconds.

Comparable to its importance for humans, tactile sensing is one of the faces and part of the emerging technology that is gaining importance in different fields of our daily life issues.

Even though autonomous robots mainly rely on some form of visual perception to interact with the surrounding environment, there are tasks that would be impossible or too complicated without the sense of touch. Inferring contact information from vision requires complex 3D scene reconstruction, which limits the effective deployment of robots in dynamical environments. Tactile feedback has the potential to improve the interaction skills of robots. For example, in the control of grasping and manipulating objects, touch provides important information related to the position of the object in the hand, thereby informing the controller about the local surface curvature of the object, friction, or the force exerted by the fingers. Overall, touch helps the robot to deal with uncertainties — about the object position or its shape — that make purely vision-based approaches difficult in unstructured environments. Recent research targets algorithms that enable the learning of control strategies to maintain a stable grip in the presence of uncertainties or perturbations [5]. In particular, slip detection and force control allow manipulation of fragile objects or those with slippery surfaces. These tasks require tactile sensors to provide accurate estimation of normal, tangential forces as well as incipient slip from tiny vibrations. Tactile sensing can also reveal objects' properties that are hidden (or difficult to extract) using vision. Combined with force/torque sensing technologies, tactile sensing allows robots to detect contacts, estimate interaction forces and regulate them for simultaneous whole-body postural and compliance control [6].

The wide range of application scenarios has boosted research in the different fields involved in the technological development of tactile sensors, ranging from material science, electronics, transducers up to communication, sensory encoding and processing, aiming at performance, sensitivity, robustness, reliability, compactness, and power consumption. The possible applications in which this kind of technology could be used are many such as prosthesis [7], in domestic houses, supervision and assistance of elderly people, industrial applications or security and military and space exploration systems. Otherwise, other fascinating and fundamental application fields could be found out in all the disciplines that concern in some sense health care and medicine in the coming future.

Inspired by biological systems, mainly the human skin, in this thesis, we merged between between CMOS technology utilizing POSFET based tactile sensors and neuromorphic circuits to mimic the biological behavior of neurons on circuit level in two modes: sustained and transient. Consequently, we implemented the principles of neurons in biology to achieve suitable tactile sensing circuits for artificial robots.

Besides, we built on the response of capacitive sensors to detect the orientation of sliding object over the skin patch of iCub robot using a machine learning technique used currently in the field of autonomous cars. The algorithm aims to classify the trajectory of the tactile stimuli. s

1.1 Neuromorphic Engineering

The approach of neuromorphic engineering applies the computational principles discovered in biological organisms to those tasks that biological systems perform easily. Neuromorphic engineering is based on the use of very-large-scale integration (VLSI) systems containing electronic analog circuits, that work in weak inversion domain, to mimic neuro-biological architectures present in the nervous system; it is a new interdisciplinary field that takes inspiration from biology, physics, mathematics, computer science and engineering to design artificial neural systems, such as, for example, vision systems, auditory processors, and autonomous robots, whose physical architecture and design principles are based on those of biological nervous systems. Biological sensory systems, responsible for processing the information of the outside world, have excellent performances in terms of compactness, power dissipation, sensitivity, the dynamic range of the response and resistance to noise and are specific for the function they have to perform. For these reasons, an efficient approach is to build artificial sensory systems that are inspired by biological ones.

1.2 Event Driven Approach

The integration of multiple tactile sensing devices into the body robotic platforms, with zones of increased spatial density like the fingertips, generates a huge amount of data to be collected, communicated and processed. Thus, a need for efficient encoding of the tactile signals arises. An elegant and smart approach for such a mission is the neuromorphic event-driven asynchronous encoding and transmission of sensory data[8]. In event-driven encoding, the sensor is only activated and data is collected and processed once the amplitude of the sampled signal changes. In this way, the system automatically adapts to the dynamics of the external environment and removes redundancy at the acquisition level. As a result, a decrease of the bandwidth, power and computational load of the whole system is achieved with guaranteed high temporal resolution and low latency[9].

A notable example of event-driven sensing, that is being adopted by the robotics and computer vision communities, is the Dynamic Vision Sensor (DVS) [10]. It comprises an array of independent pixels, each responding to local changes of brightness, once a change is detected, the sensor sends an output digital pulse that signals wherein the array and when the change has happened, using the standard Address-Event-Representation (AER) [11]. As each pixel sends a digital pulse only – and as soon as – a change is detected, without waiting for the whole array to be scanned (as in traditional imaging), the sensor removes redundancies and ensures minimum latency.

1.3 Tactile Sensors

The development of the sense of touch in robotics is an engineering challenge. The so-called electronic skin, which covers different parts of a robot with sensors that respond to mechanical and other environmental stimuli, requires system level development that spans from materials and electronics up to communication and processing.

The current state-of-the-art solutions for tactile sensors provide a wide variety of different types based on using the various mode of transduction. The major transduction methods that have been reported in the literature are: resistive, tunneling effect, capacitive, optical, piezoelectric, and others. Each method has specific characteristics, advantages/disadvantages and peculiarities.

In the coming sections, I will briefly go through some of the transduction methods and explain some of their properties and drawbacks [12].

1.3.1 Resistive Sensors

Tactile sensors based on the resistive mode of transduction are of two types: the resistance that depends on contact location and resistance that depends on the applied force or, in other words, piezo-resistive tactile sensors.

Piezoresistive touch sensors are made of materials whose resistance changes with force/pressure. Piezoresistive tactile sensing is also popular among the MEMS-based and silicon-based tactile sensors [13]. The sensors are appealing, because of the low cost, good sensitivity, low noise and simple electronics and can be found in many experimental tactile systems. One of their drawbacks is the relatively stiff backing, high power consumption, hysteresis and short lifetime of the materials. Although examples of advanced robotic hands equipped with such mode of transduction [14], these sensors generally require a serial or manual assembly, provide a highly non-linear response and suffer from hysteresis.

1.3.2 Tunnel Effect Tactile Sensors

Tactile sensors based on QTC have come up recently and are commercially available. QTC's have the unique capability of transformation from a virtually perfect insulator to metal like a conductor when deformed by compressing, twisting or stretching of the material. Robot hands with QTC based tactile sensors have also been reported in literature [15] and [16].

Integration of such method in tactile sensors makes it difficult to be implemented on fingertip because the size of a sensor (2.5 cm^2 size) which is larger than a typical human fingertip, but, has a spatial resolution better than that of the human fingertip (40 μm).

1.3.3 Optical Sensors

Tactile sensors with the optical mode of transduction use the properties of optical reflection between media of different refractive index. The transducer structure is composed of a clear plate, a light source and a compliant membrane stretched above, but not in close contact with the plate. The lower surface of the plate acts as the imaging area. Light is directed along an edge of the plate and it goes through total internal reflection (when no force is applied) or diffuse reflection (when force is applied). The light coming out of the plate due to diffuse reflection can be recorded by CCD or CMOS cameras placed in the imaging area. The intensity of the light (bright or dark patches on image) is proportional to the magnitude of the pressure between object and plate. Some cases of large area skin based on LEDs has also been reported [17]. Also, fusion between touch and vision has been reported in [18].

Optical based tactile sensors are immune to electromagnetic interference, are flexible, sensitive and fast, but in some situations they are bulky. Besides, they face a decrease in their performance under strong light conditions and high power consumption.

1.3.4 Capacitive Sensors

Capacitive sensors consist of a capacitor, in which the distance between plates or the effective area is changed by the applied force by shifting their relative position due to their separation by a deformable dielectric material. The measurement of capacity yields an estimate of pressure [19] and [20]. The compatibility of these sensors with flexible substrates and the availability of off-the-shelf components for the readout electronics make capacitive technology suitable for robotics, especially for large areas.

Capacitive sensors are compact, highly sensitive and have theoretically unlimited operational bandwidth (in practical cases the choice of the dielectric material often limits the bandwidth to relatively low-frequency ranges), availability of commercial A/D chips (technology borrowed from the touch screens), wide range of measurable forces and performs very well with regard to typical problems of capacitive tactile sensor such as cross-talk between the taxel and stray capacity [21]. The main drawbacks are the degradation of the elastomeric materials used for the deformable dielectric due to mechanical wear and tear, hysteresis, drift of sensitivity due to temperature, and — depending on the materials — relatively complex production processes.

In my second part of the thesis, I used the capacitive sensors [19],[22] and [23], i.e. a complete skin patch of the iCub robot, to detect the orientation of a sliding object on it.

1.3.5 Piezoelectric Sensors

The piezoelectric materials have the property of generating charge/voltage proportional to the applied force. Alternatively, they are capable of generating force due to electrical input. Thus, they can be used both as sensors and actuators and due to this property they fall under the category of ”Smart Materials”. Piezoelectric materials are suitable for use as tactile sensors. However, the temperature sensitivity of piezoelectric materials is a major cause that hinders use as tactile sensors.

The use of PVDF for tactile sensing was reported for the first time in [24] and thereafter several works based on PVDF or its copolymers have been reported in literature [25],[26] and [27].

While quartz and some ceramics PZT have better piezoelectric properties, the polymers such as polyvinylidene difluoride (PVDF) normally have been used in touch sensors because of some excellent features, such as flexibility, workability, and long-lasting chemical stability [28]. Moreover, their response is fast and linear over a large range of stimuli, making them also suitable for dynamic force sensing. They have been used for the implementation of tactile sensors based on an integrated device.

This thesis is divided into two parts.

1. **In part I**, we present the characterization of two neuromorphic circuits for readout of the POSFET:

   The first is based on the Leaky IF (LIF) neuron [29]. When the POSFET is touched, its output is transformed into a current and integrated by the LIF neuron that emits digital pulses at a rate proportional to the input force or its slope [30–33], reproducing forms of transient and sustained activity typical of skin mechanoreceptors (Discussed in Chapters 2 and 3).

   The second is more compact, with lower latency and area-saving version of the event-driven POSFET for the encoding of transient activity of the input signal: it is based on the change detection circuit of the DVS pixel, where the POSFET output substitutes the input photoreceptor with The design comprises the self-resetting switched capacitor amplifier and comparators of the DVS pixel and is driven by a source follower connected to the output of the POSFET. This circuit only encodes the transient component of the signal (Discussed in Chapter 2).

2. **In part II**, for its importance for robots on the control side and estimation of the structure and properties of objects, sliding, we present a machine learning technique suitable for detecting the orientation of a sliding object over the skin patch of the iCub robot. The algorithm is based on a probabilistic graphical model (**PGM**). The latter is divided into parts: offline learning and online testing. While the first part includes the learning phase of different paths followed by the sliding object, the second part includes the online testing where a new path is considered and the algorithm predicts to detect the orientation of such new path (Highlighted in Chapter 4).

Chapter 2

Neuromorphic Tactile Sensing Circuits: Properties, Operation and Simulation

2.1 Introduction to Silicon Neurons

Neuromorphic circuits are hybrid analog/digital very large scale integration (VLSI) circuits that emulate the electro-physiological behavior of real neurons and conductances.

This type of neuromorphic interface circuits is constituted mainly by MOS devices working in the sub-threshold regime. The characteristic features of this type of circuits is event-based asynchronous operation, which differs from the traditional mode of operation of clock-driven digital circuits.

The event-based approach, instead, responds asynchronously to the set event, as the relative temporal changes in input intensity, with a redundancy and response latency reduction and improvement of the bandwidth and dynamic range. To this just described is added the possibility of exploiting constructive working standard solutions at low power consumption which does not require an excessive complexity of the wiring as exploitable with the AER.

Numerous physiological studies have precisely delineated the operation of the principal type of cells of the central nervous system. Neurons are highly specialized for generating electrical signals in response to chemical and electrical inputs and transmitting them to other cells.. The soma is the "central process unit" of the neuron, and it performs an important non-linear processing step: if the total input exceeds a certain threshold, then an output signal, the action potential, is generated. The electrical relevant signal for the nervous system is the potential difference across the soma membrane. Under resting conditions, the potential inside the cell membrane is about -70 mV relative to that of the surrounding bath, conventionally defined to be 0 mV, and the cell is said to be depolarized. To maintain such a potential difference, the current has to flow. This is the activity of the ion pumps located in the cell membrane which transport ions to maintain ionic concentration gradients.

Predominantly, sodium, potassium, calcium, and chloride are the ionic species involved. Ions flow according to their concentration gradient through a variety of ions channels that open and close in response to voltage changes as well as to internal or external signals. Current flowing through open channels outside the cell makes the membrane passes a threshold, a positive feedback process is initiated, and the neuron generates an action potential. It is roughly a 100 mV fluctuation of the membrane potential lasting about 1ms. This signal propagates along the axon where it is actively regenerated to arrive at the synaptic bouton at the end of the axonal arborization. The generation of action potentials also depends on the recent firing history of the cell. For a few milliseconds, after an action potential has been fired, it may be virtually impossible to initiate another spike. This is called the absolute refractory period.

Hardware emulations of neural systems that use SiNs operate in real-time, and the speed of the network is independent of the number of neurons or their coupling. SiNs offer a medium in which neuronal networks can be emulated directly in hardware rather than simply simulated on a general-purpose computer. They are much more energy-efficient than simulations executed on general-purpose computers, so they are suitable for real-time large-scale neural emulations. Where SiN circuits provide a tangible advantage is in the investigation of questions concerning the strict real-time interaction of the system with its environment. Besides, the technology developed to build these real-time, low-power neuromorphic systems can be used to engineer brain-inspired computational solutions for practical applications.

The term neuromorphic includes a set of analog VLSI circuits that operate using the same physics of computation used by the nervous system (e.g., silicon neuron circuits that exploit the physics of the silicon medium to directly reproduce the bio-physics of nervous cells) and also analog/digital hardware implementations of neural processing systems, as well as spike-based sensory processing systems [34].

Within this context, many different types of SiNs have been proposed, that emulate real neurons at many different levels: from complex biophysical models that emulate ion channel dynamics and the detailed dendrite or axon morphologies to basic integrate- and fire (IF) circuits [29].

2.2 Neuromorphic Circuits

2.2.1 Integrate and Fire Neuron

The IF neuron models of spiking neurons have complex dynamics that require intensive computational resources and long simulation times. This is especially true for conductance-based models that describe in detail the electrical dynamics of biological neurons. These models include non-linear voltage-dependent membrane currents and are difficult to analyze analytically and to implement. For this reason, phenomenological spiking neuron models are more popular for studies of large network dynamics. In these models, the spikes are stereotyped events generated whenever the membrane voltage reaches a threshold. The IF model neuron, despite its simplicity, captures many of the broad features shared by biological neurons.

This model can be implemented using analog VLSI technology and can be used to build low power, massively parallel, large recurrent networks, providing a promising tool for the study of neural network dynamics [35]. Starting from the Hodgkin and Huxley model (refer to [36] for a more detailed discussion) simplifications are possible, and they lead to a one-dimensional integrate-and-fire model. The analytical tractability of this effective description together with its simple silicon implementation will make it the favorite candidate to realize and study, both theoretically and experimentally, controllable neural networks. The reduction of the model is based on simplifications on the output and the input side of the Hodgkin and Huxley model. On the output side, the major consideration is that a Hodgkin and Huxley neuron will typically emit a spike whenever its membrane potential reaches a threshold value of about -55 to -50mV. The action potential produced is roughly always the same independently from the evolution of the input currents that have triggered the spike.

The spike has a stereotyped shape that seems not to convey important information, which will arise from the time of spike occurrences. This suggests that the generation of the spike can be incorporated from the equations and reduced to a pure boundary condition so that when the membrane potential V(t) crosses a given threshold θ_V, then V(t) undergoes a pulse-like excursion, the spike, before returning to resting value Vr. The costly numerical integration is then stopped as soon as the spike is triggered and restarted after the downstroke of the spike about 1.5 to 2ms later. This interval of time corresponds to an absolute refractory period (τ_{abs}) of the neuron. This reduction simplifies the equations that have to describe only the sub-threshold behavior of the potential and no more the delicate equilibrium among conductances dynamics that account for the spike generation.

$$\frac{dV}{dt} = F(V) + \frac{I(t)}{C} \tag{2.1}$$

Equ.2.1 is the generic formulation of the integrate-and-fire model. The function F(V) could be further simplified and reduced for instance to a linear function: this is possible discarding all the active conductances and maintaining only the constant one gl of the leakage current (mainly due to chloride ions). This means that the membrane potential is stimulated only by the synapse contributions, and merely discharged through a fixed conductance. This version of the model is called the passive or leaky integrate-and-fire (LIF) neuron. For small fluctuations about the resting membrane potential, neuronal conductances are approximately constant; the LIF model assumes that this constancy holds over the entire subthreshold range.

2.3 Change Detector Neuron

Another possible compact neuromorphic circuit that processes information using energy- efficient, asynchronous, event-driven methods to detect temporal contrasts of the input, is outlined in this section. This type of circuit, named asynchronous change detector neuron is inspired by some work that has focused on visual sensors ([37];[38]; [39]). In particular, one of the distinguishable characteristics between the various features of this activity-driven event-based vision sensors is that they, above all, quickly output compressed digital data in the form of events: in fact, these sensors reduce redundancy, latency and increase dynamic range. Moreover, the output of these digital sensors is easily interfaced to the conventional digital post-processing, reducing costs and latency. The exploited key point in thinking about such an interface circuit to the transducer was trying to take advantage of the biologically inspired use of asynchronous, exceedingly sparse, data-driven digital signaling as core aspect of the computational architecture Address-Event Representation (AER) systems naturally providing a way to incorporate demand-based computation, where data originating in one place drives computation in another. The high speed of silicon electronics allows communication of sparse, low frequency asynchronous digital address-events by sharing high speed digital buses.

The basic feature of the circuit presented is the possibility to, independently and in continuous time, quantize local relative intensity changes of the input, to generate spike events. These events appear at the output of the sensor as an asynchronous stream of digital pulses that signify external force (applied on the transducer) change and have sub-millisecond timing precision. The output data rate depends on the dynamic content of the external solicitation and is typically orders of magnitude lower than those of conventional mode of transduction, which also involves high transmission power dissipation, increased channel bandwidth requirements, increased memory size and post-processing power. The objective of this pixel design was to achieve a low mismatch in the transistor elements and low latency. In general, as well as in this particular case, this type of neuromorphic circuits also has the important characteristic to easily remove the DC mismatch between pixels or taxels, caused by the variations of threshold voltage levels of every transistor. The removal of the mismatch DC is made possible thanks to a particular configuration of the differencing circuit. Other strengths of the configuration under consideration that deserve to be mentioned, are the reduced power consumption and the compactness due to the small circuit area which can be thought of as a good choice in the case of the humanoid robot iCub.

2.4 Neuromorphic Tactile Sensing Circuits

I will highlight the different sensor circuits under study by describing their functionalities, constituents and finally the simulation results.

The neuromorphic CMOS circuits included within the chip are two. The first circuit is named: proportional POSFET and the second is differential POSFET circuit. The two different circuits have been proposed to explore the solution for the realization of a novel sensing device. The purpose behind these two configurations is to find the best candidates to emulate the functioning of the basic corpuscles that give the sense of touch to humans and implement it employing electronic circuits in tactile sensors domain for artificial devices.

2.5 Methodology

Before going into the details of the circuits under consideration and their properties, I would like to describe the different steps followed to achieve the desired goals listed below. Having the circuits ready for characterization, the plan is to simulate them at first and then build the suitable experimental setup to fully characterize them. The schematics for the circuits are implemented using Cadence 6.1.6. Thus, the simulation part started using the latter by performing static and dynamic analysis. This step provided a good knowledge about the behavior of the circuits under different inputs and conditions as a result of applying different simulation trials and tests. Next to that, the work started on building the experimental setup. With the help of the latter, similar to the simulation tests are done and data is collected for analysis.

From one side, the first goal behind the analysis part is to compare the new collected data to that from [40] (that is the older version of the circuit prototype) to make sure that the modifications for the new circuit design is suitable for reducing the disadvantages of the proportional configuration in [40] for what concerns the offset, low sensitivity for low input signals, stability and robustness to the fabrication mismatch. Second, it aims to show that combing the POSFET and the change detector can replace the differential configuration based on the LIFs presented in [30] where the peculiarity of this circuit is all represented by the highlighted NAND port on the feedback loop. This is a classic digital port that takes the digital AER interface output and bring it back to the offset cancellation circuit. Therefore, by taking into account the advantages of the change detector which is the basis of the Dynamic Vision Sensor (DVS) silicon retina being the artificial synapse for visual applications that is able to encode the variation of the input force or voltage. From the other side, the plan is to build on the analysis to plan for future circuit prototypes making use of the current and the older ones.

2.6 Transducer Adopted in Our Sensor Circuits

2.7 POSFET Device

Unlike conventional sensing devices, the POSFET (piezoelectric oxide semiconductor field-effect transistor) [41], where the piezoelectric material, deposited over the gate of complementary metal– oxide–semiconductor transistor, senses the force-generated charges (refer to **Fig.2.1**). This device is innovative for the fact that it presents an integration of the readout circuitry with the sensing material (i.e. piezoelectric polymer film) being therefore consistent with the ”Sense and Process at same place” concept. The fusion of the sensing material and the electronics improves the signal to noise ratio and thus enhance its force sensitivity. The concept of the POSFET is based on the knowledge and potentiality of piezoelectric materials. In particular, it is known that a piezoelectric film working in the sensing mode generates a charge/voltage as a matter of applying a force/stress on it. This charge/voltage is proportional to the applied force/stress.

As the first part of the sensor, the POSFET is maintained in common drain configuration. The underlying MOS transistor of the POSFET is smaller in dimensions (see **Table 2.1**). This change in return saves the occupied area by this big transistor on the fabricated chip. The bias current of the POSFET is set by a cascoded current mirror, that can be changed by tuning the corresponding input gate voltages of the current source transistors, sets its biasing point in the saturation region

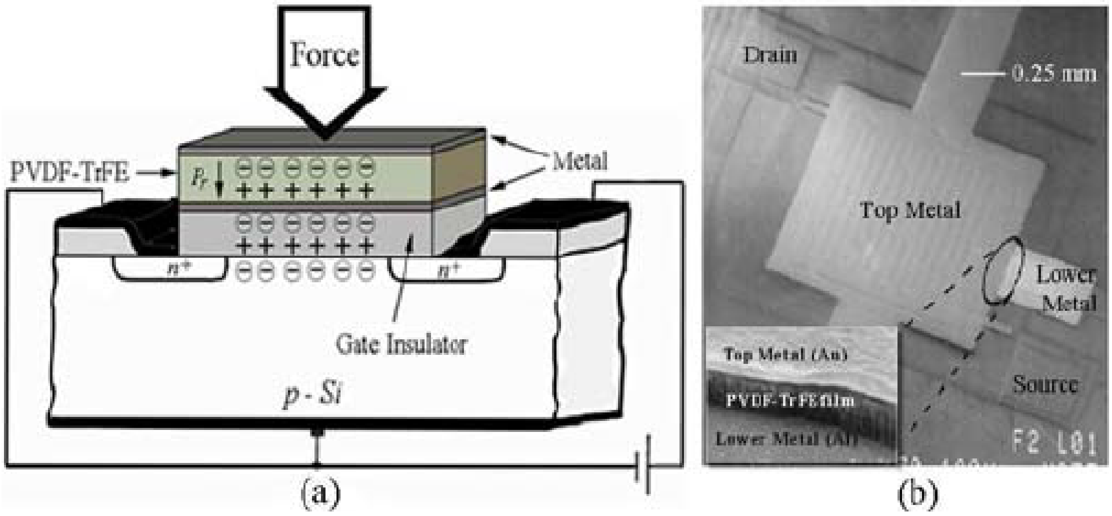


FIGURE 2.1: (a) POSFET device structure. (b) Depositing the piezo-electric material on the gate of a MOS transistor [1].

providing a precise supply current to the sensor stage. According to the dimensions and configuration, the POSFET acts as a voltage buffer stage copying at its source terminal the voltage signal applied to its gate with a gain ≃1 on one side, and decouples the piezo-electric material (PVDF-TrFE) from the electronic circuitry from the other side.

2.8 The Proportional POSFET Sensor Circuit: Structure and Operation

Figs. 2.2 and **2.3** shows the block diagram and circuit schematics of the fabricated ED taxel. The POSFET is composed of the PVDF-TrFE, deposited on a gate of an underlying MOS transistor. Usually, the PVDF-TrFE is deposited on the silicon oxide utilizing spin coating, resulting in a compact sensotronic device with no wiring requirements and less sensitive to noise. In our test prototype, however, the characterization is performed with a PVDF-TrFE layer remotely connected to the gate of the POSFET transistor.

In the proposed implementation, the POSFET response to the mechanical stimulus is linearly transformed into trains of digital pulses with frequency proportional to the applied input force/pressure.

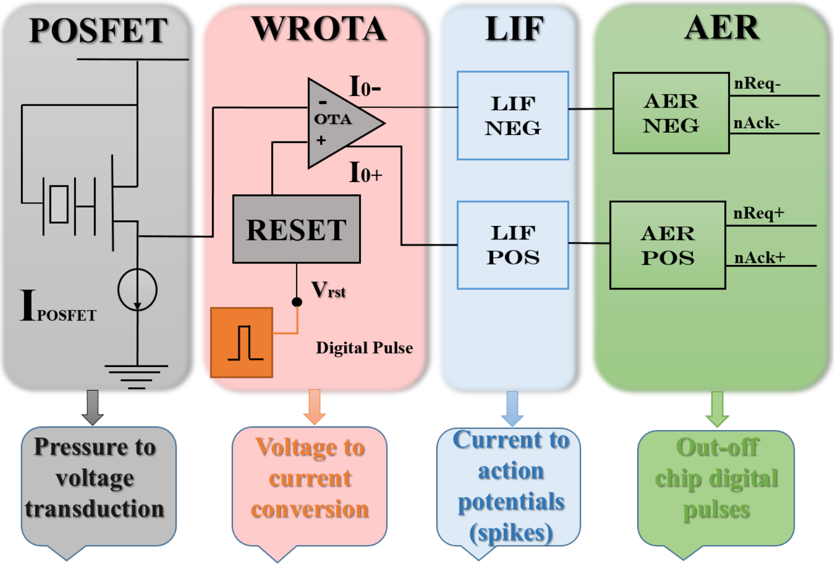

POSFET
WROTA
LIF
AER
Io-
OTA
-
+
Io+
RESET
Vrst
Digital Pulse
IPOSFET
LIF NEG
LIF POS
AER NEG
AER POS
nReq-
nAck-
nReq+
nAck+
Pressure to voltage transduction
Voltage to current conversion
Current to action potentials (spikes)
Out-off chip digital pulses

FIGURE 2.2: Block diagram: The POSFET in source-follower configuration converts pressure into a voltage signal. A Wide Range OTA (WROTA) compares the output of the POSFET to the reference voltage stored across a capacitor. The latter voltage of the WROTA is regularly restored by the RESET circuit, to remove slow drifts. The output current – Io− (for decreasing pressure) and Io+ (for increasing pressure) – feeds the corresponding LIF circuit. The AER block manages the protocol handshake by sending digital pulses out off chip.

The POSFET is in common drain configuration with tunable biasing current I_{POSFET} provided by a cascoded current sink to provide a more reliable and precise supply current to the sensor stage. This biasing configuration of the MOS ensure that the POSFET operates in strong inversion and is always in saturation. The small voltage excursion due to the pressure applied to the PVDF does not change the operating point of this structure. The POSFET output voltage is amplified by a symmetric Wide Range Operational Transconductance Amplifier (WROTA) to obtain a current signal proportional to the applied contact force/pressure.

2.8.1 The Differential Stage: WROTA and Offset Cancellation Circuits

The second part of the sensor circuit, i.e. the Differential stage, is composed of the WROTA and the Reset in pink and blue respectively as shown in **Fig. 2.3**. The second stage shares with the POSFET device the bias circuit.

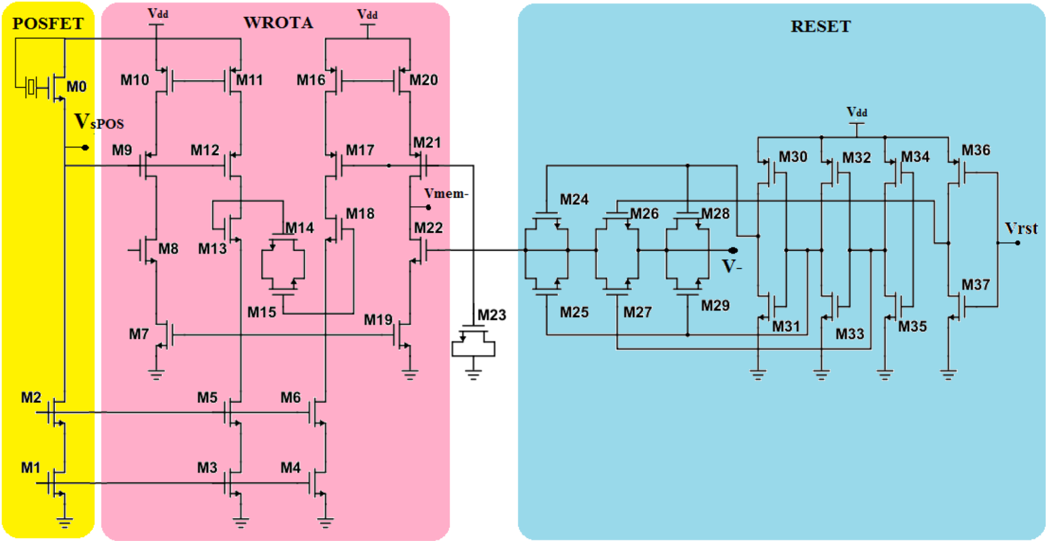


FIGURE 2.3: Circuit schematics: the POSFET, WROTA and Reset circuits. For clarity, only the OTA branch with the output for the negative LIF V^{-}_{mem} is shown.

In general, for all applications which need a precise translation from a voltage signal to a current one, the transconductance must be as constant as possible to prevent the introduction of non-linearities across the generated current at its outputs. Thus, the implementation of the Krummenacher design [42] is chosen in this WROTA design. This allows enlarging significantly the input range in which the transconductance is almost constant improving the linearity of the transfer function of the WROTA compared to the classic OTA where the transconductance is considered to be constant only in a small neighborhood of the differential input voltage where it is zero. Besides, the branches of the differential pairs are cascoded using the low-voltage cascode mirrors (see **Fig. 2.3**). The two current sinks of the differential pair and the auxiliary branches are equipped with this configuration to ensure the largest current matching possible to the output to achieve the possible symmetric currents for the positive and negative LIFS in case of pressure and release. The fully cascoded output provides also a balanced output resistance reducing considerably the output voltage lowering. The transmission gate is designed with balanced transistor dimensions and double dummy switches on both terminals for the reductions of disturbances caused by the charge injection on the source terminal of the POSFET device. A greater capacitance in length and width is used to reduce the noise effects and the drift disturbances on the second differential input of the WROTA. Finally, implementing a series of inverters to produce a better reset signal from a generic source. By this way, the switches are synchronized to reduce the residual charge injection.

A transfer-gate switch (in the lower part of the Diff block) shorts the POSFET output terminal with the gate of a MOS capacitor connected to the negative input of the WROTA in order to track the bias voltage of the first stage. When the transfer-gate opens, the input voltage is sampled across the MOS capacitor and this allows to polarize the WROTA negative terminal and to obtain an output current proportional to the variations. This configuration implements a sample and hold circuit aimed at implementing an offset cancellation device that can be periodically controlled by a digital pulse to remove low-frequency steady-state fluctuations. The output of the WROTA drives two identical LIF circuits that translate their input current into spike trains. As the LIF circuits are rectifying, the current output signals are I$o+$, which corresponds to a positive input stimulus (pressure on the sensor) and I$o-$ for a negative one (release). The two current signals are in turns integrated by the membrane capacitors of two LIFs (one encoding for pressure, named LIF POS, and one encoding release, named LIF NEG, respectively).

The membrane potential of the LIF neurons (nodes corresponding to I$o-$ and I$o+$) increases at the increase of the input current, and then it generates an action potential when it crosses a tunable threshold. By changing V_{sf} (i.e. the gate voltage of the biasing MOS transistor of a common drain included in the neuron's circuit), it is possible to change the neuron's threshold voltage, for more information refer to [29]. The membrane potential is transformed into digital pulses by the "AER" block that handles the digital pulses (nReq+ and nReq-) and the handshake for the management of the Address Event Representation protocol (nAck+ and nAck-) [11].

To reduce low-frequency disturbances and drifts, the "Diff" block comprises an offset cancellation circuitry activated by the external reset signal Vrst.

The digital pulses are sent off-chip through of a small handshake circuit that allows the transmission of the pulse only after the receiver has acknowledged the request signal. This structure will be necessary when the multiple ED taxels will be integrated on a single die to form a matrix of sensing elements, whereby the digital pulses from each taxel will be multiplexed over time on a single bus, using the standard Address-Event Representation (AER) [11].

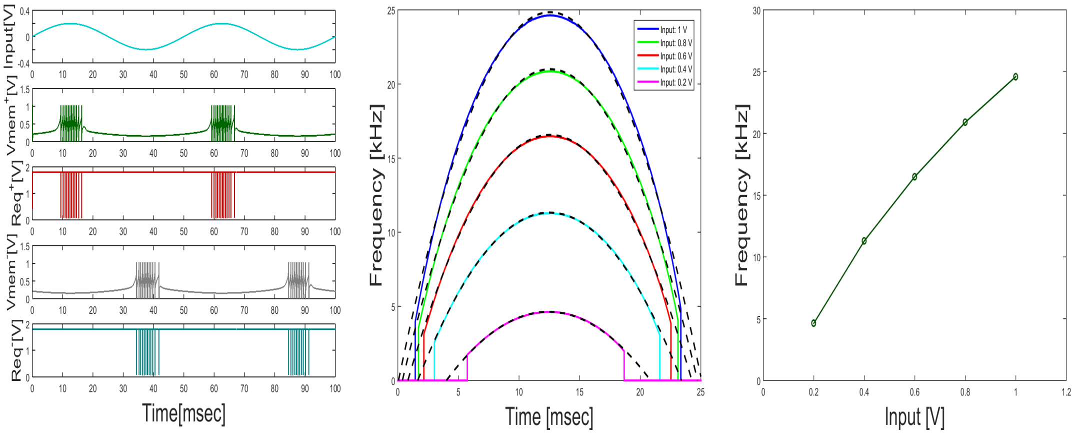
The LIF and AER circuits are the same used in [40], with updated size to adapt to the 180-nm process. In **Table 2.1**, we show the dimensions of the main transistors for the first two stages of the sensor circuit where the four output transistors of the WROTA have a double size to increase the transconductance.

Table 2.1: **Dimensions of Most Critical Transistors**

Transistors	**W** [μm]	**L** [μm]
M_0	651	1
M_1- M_8, M_{19},M_{22}	5	1.3
M_{13}, M_{18}	5.4	1.3
M_{14}, M_{15}	1.35	1.3
M_9-M_{12}, M_{16},M_{17}	7.6	1.3
M_{20}, M_{21}	15.2	1.3
M_{24}, M_{28}	0.22	0.18
M_{26}	0.44	0.18
M_{30},M_{32},M_{34},M_{36}	0.26	0.18
M_{31},M_{33},M_{35},M_{37}	0.22	0.18
M_{23}	41.5	41.5

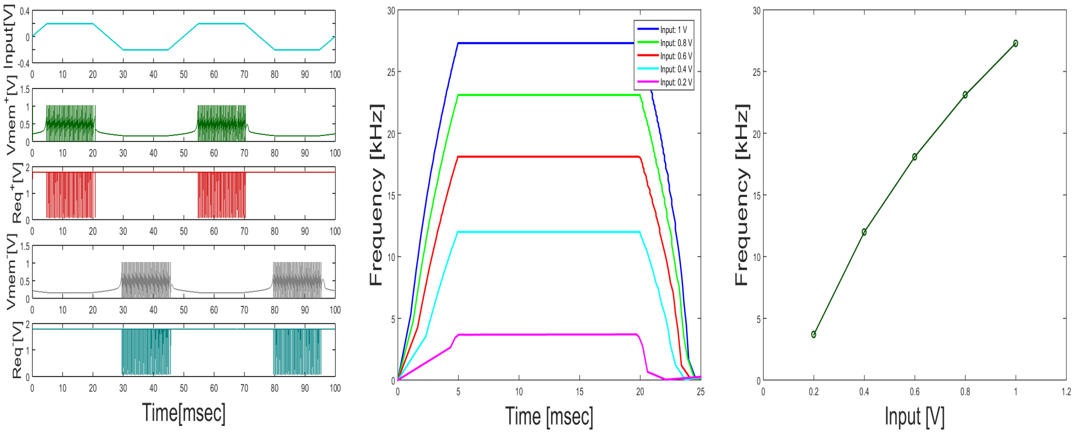
2.9 Sensor Circuit Simulation Results

Through simulations on SPECTRE Cadence 6.1.6, we show the behavior of the overall sensor and highlight the improvements concerning the sensitivity and neuron firing rate. To examine the overall behavior of the sensor and to prove it quantitatively, we applied trapezoidal and sinusoidal input waveforms along with a sweep on the input amplitudes.



(A) (B) (C)

Figure 2.4: (A) The response of the positive and negative LIFs to an input voltage of 100 mV amplitude with frequency 20 Hz. (B) Results of five simulations with an amplitude sweep on the input voltage: Instantaneous frequency for a positive half period of the sinusoidal input for the positive LIF fit with sine waves. (C) Peak firing rate of the positive LIF for different amplitudes of the sinusoidal input voltage.



(A) (B) (C)

Figure 2.5: (A) The response of the positive and negative LIFs to an input trapezoidal voltage of 100 mV amplitude with frequency 20 Hz. (B) Results of five simulations with an amplitude sweep on the input voltage: Instantaneous frequency for a positive half period of the sinusoidal input for the positive LIF. (C) Peak firing rate of the positive LIF for different amplitudes of the trapezoidal input voltage.

Figs. 2.4a and **2.5a** show the response of the sensor circuit to sinusoidal and trapezoidal inputs of 100 mN amplitude at 20-Hz respectively. The cyan curves show the input voltage, the green and gray show the positive and negative neuron membranes and the corresponding digital outputs, obtained by the combination (logic OR) of the two request signals of the ”AER” block over time in red and cyan. Both neurons respond to the input voltage once its amplitude is constant and stay silent once the latter changes (a sign of proportional behavior), unlike the two comparators that respond only to the variation in the input amplitude [43]. **Figs. 2.4b** and **2.5b** show the instantaneous firing rate of the positive neuron for different values of the input voltage amplitude over a positive half period of the input waveform. In **Fig. 2.4b**, the output follows a sine function as shown through the fit with the dashed sine functions. This proves that the output firing rate of the circuit responds proportionally to the input voltage, rather than to its rate of change over time. The peak instantaneous firing rate of the curves in **Figs. 2.4b** and **2.5b** are displayed in **Figs. 2.4c**, and **2.5c** versus the input voltage amplitude respectively. The circuit transfer function illustrates the linearity between the output firing rate with the input voltage to the POSFET.

2.10 The Differential POSFET Sensor Circuit: Structure and Operation

Fig. 2.6 shows the schematic of the tactile sensor circuit – or taxel – equipped with the change detection and pulse generator circuits from the Dynamic Vision Sensor pixel. The photo-receptor, that responds to brightness variations, is replaced by the POSFET in common drain configuration biased by the current I_{POSFET}. The POSFET copies the input force signal to its output with approximately unity gain. Thus, it drives the capacitive input of the differencing circuit. The following capacitive feedback inverting amplifier is balanced with a reset switch that shorts its input and output together, resulting in a reset voltage level. The self-resetting switched capacitors C_1 and C_2 differential circuit [10] removes the DC signal from the input and generates the derivative of the output voltage signal coming from the POSFET. Concerning the DVS photoreceptor stage, the logarithmic amplifier is removed and the buffer stage is converted into a common source configuration based on M_{sfb} and M_{sf}, where M_{sf} is no more a PMOS but rather an NMOS. The input stage of the differential circuit is hence an inverting amplifier rather than a source follower. The comparators (MONn,MONp,MOFFn,MOFFp) compare the output of the inverting amplifier against global thresholds arbitrary tuned that are offset from the reset voltage to detect increasing and decreasing changes. If the input of a comparator overcomes its threshold, an ON or OFF event is generated. The "ON" and "OFF" comparators produce digital pulses when the differential circuit output exceeds (or goes lower than) an externally tuned threshold (V_{don} and V_{doff}), respectively. As one of the two comparators switches, the AER block sends a digital pulse to the output arbitration and implements taxel level handshake that resets the differential amplifier.

The ON and OFF events are communicated to the periphery by the circuits that implement the 4-phase AE handshaking with the peripheral AE circuits. For further information concerning the peripheral AE circuits, please refer to [37].

Table 2.2 shows the dimensions of transistors and capacitors capacitance used in the simulations and for a prototype test circuit sent to production, in AMS 180 nm technology. In particular, the POSFET is much smaller than in the previous implementation [32] reducing the overall area occupied. Further, with respect to the previous design (in AMS 350 nm), capacitors C_1, C_2 and C_3 are implemented

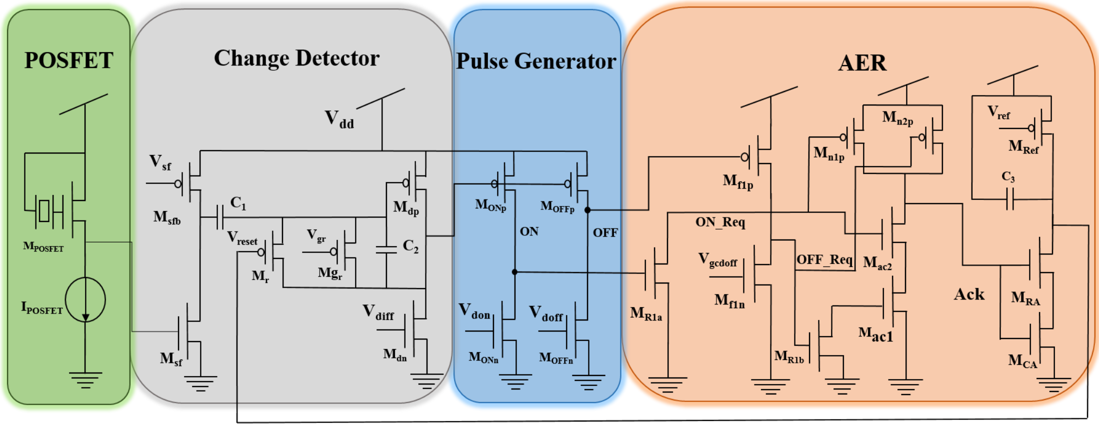


FIGURE 2.6: Circuit Schematic of: the POSFET, change detector, pulse generator and the AER

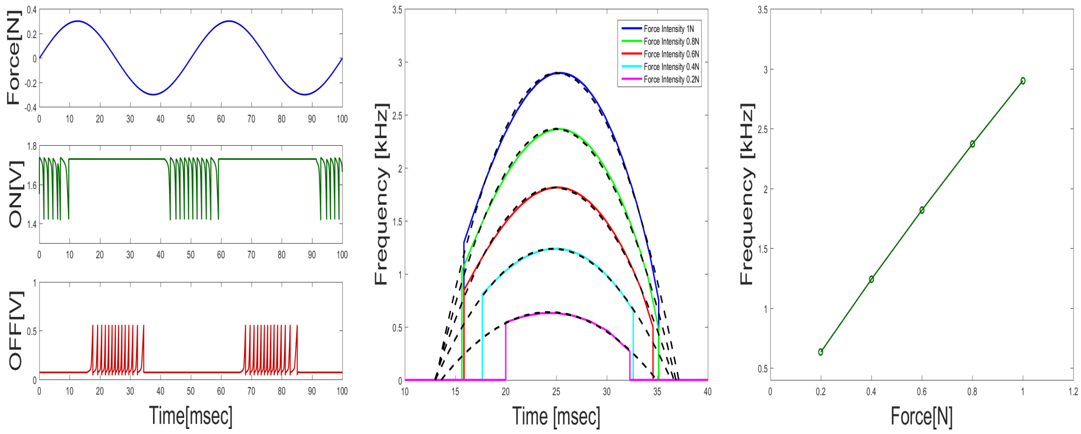
as Metal Insulator Metal (MIM) capacitors, featuring more linear capacitance, more accurate proportions and a much smaller structure that could be overlapped to the the circuit itself, saving additional area.

Transistors	**W** [μm]	**L** [μm]
M_{POSFET}	651	1
M_{sfb}	0.6	0.6
M_{sf}	9	0.22
M_r,M_{gr}	0.5	0.180
M_{dp},M_{ONp},M_{OFFp}	0.8	1.6
M_{dp},M_{ONn},M_{OFFn}	0.8	1.6
M_{R1a},M_{f1p}	0.22	0.3
M_{f1n}	0.6	1.2
M_{R1b},M_{n1p},M_{n2p},M_{ac2},M_{ac1}	0.22	0.3
M_{Ref}	0.6	1.1
M_{RA},M_{CA}	0.5	0.3
Capacitors	**Capacitance** [fF]	
C_1	602.13	
C_2	30.95	
C_3	32.54	

TABLE 2.2: **Transistors Dimensions and Capacitors Capacitance.**

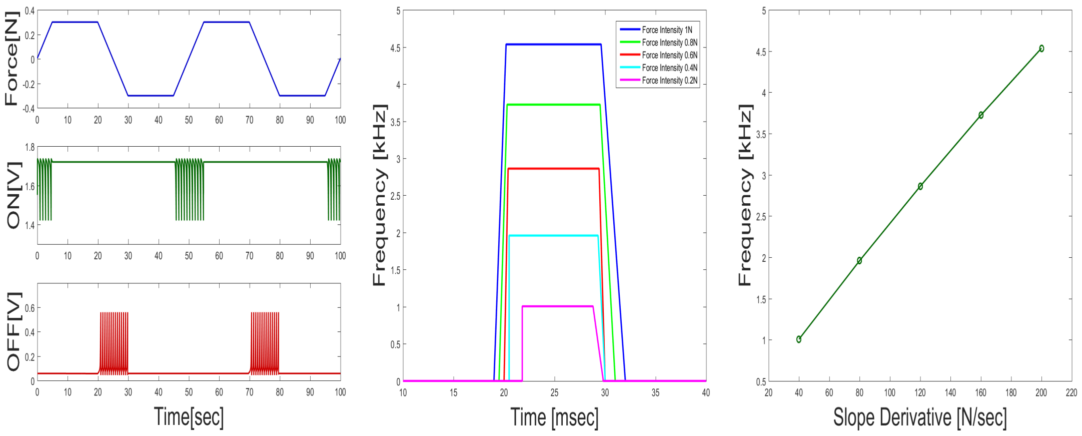
2.11 Circuit Characterization Through Simulation

To validate the behavior of the overall sensor, we performed simulations on SPECTRE Cadence 6.1.6. The simulation comprises mechanical, electro-mechanical and electrical models of the PVDF-TrFE [31]. We used sinusoidal and trapezoidal input waveforms to characterize the circuit behavior for changing input force amplitude and slope. This quantitative analysis provides an evaluation of the overall system functionality and proves the feasibility of the approach used to realize a differential – or *transient* – encoding system.



(A) (B) (C)

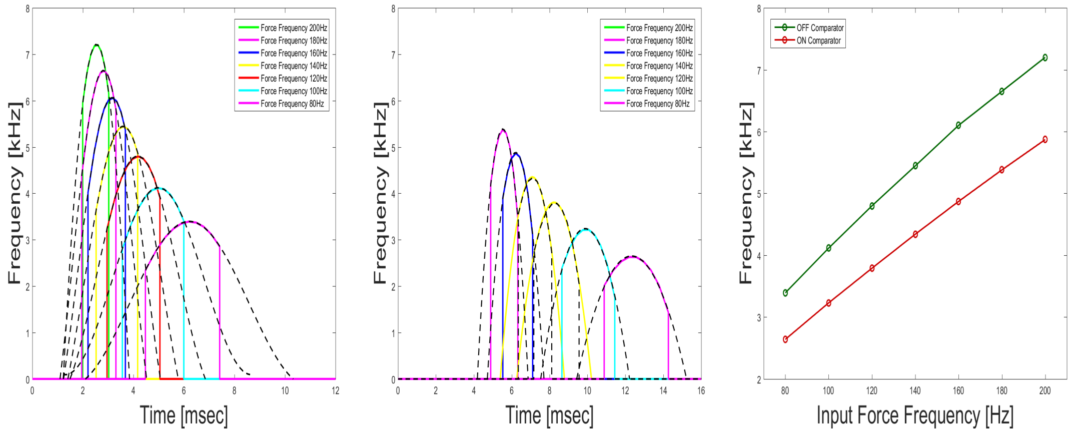
Figure 2.7: (A) The response of the ON and OFF comparators to an input sinusoidal force of 300mN amplitude and frequency 20 Hz. (B) Results of five simulations with an amplitude sweep on the input wave force: Instantaneous frequency for a positive half period of the sinusoidal input for the OFF comparator fit with cosine waves. (C) Peak firing rate of the OFF comparator for different amplitudes of the sinusoidal input force.



(A) (B) (C)

Figure 2.8: The response of the ON and OFF comparators to an input trapezoidal force of 300 mN amplitude and frequency 20Hz. (B) Results of five simulations with an amplitude sweep on the input wave force: Instantaneous frequency for a positive half period of the trapezoidal input for the OFF comparator. (C) Peak firing rate of the OFF comparator for different slopes of the trapezoidal input force.

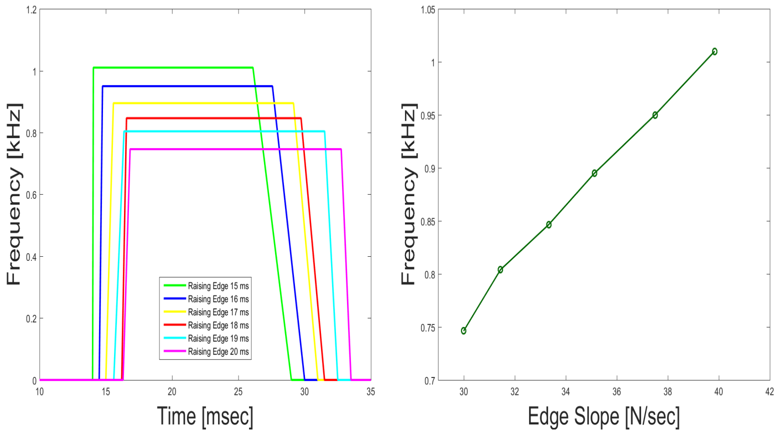
Figs. 2.7a and **2.8a** show the response of the circuit to sinusoidal and trapezoidal forces applied to the piezoelectric material of 20Hz frequency with a 300 mN amplitude. The top curves show the input force, the middle and bottom curves show that positive and negative charges of the input signal are transformed into output digital pulses by the ON and OFF comparator, respectively. The comparators do not respond to the constant input amplitude. **Figs. 2.7b** and **2.8b** show the instantaneous firing rate of the OFF comparator for different values of the input force amplitude over a positive half period of the input waveform. The output in **Fig. 2.7b** follows a cosine function as shown through the fit with the dashed cosine functions, confirming that the output firing rate of the circuit responds to the derivative (i.e. rate of change) of the input force, rather than to its absolute value. **Figs. 2.7c** and **2.8c** show the peak instantaneous firing rate of the curves shown in **Figs. 2.7b** and **2.8b** versus the input force amplitude, in both experiments the amplitude of the applied force was used to effectively change the slope of the applied force, but the results cannot rule out an effect of the force absolute value in effecting the output firing rate, for this reason, we performed another set of experiments where we changed only the input frequency of the sinusoidal wave and the rise time of the trapezoidal wave, while keeping the amplitude constant at 300 mN.



(A) (B) (C)

Figure 2.9: Instantaneous frequency for a positive half period with different input force frequencies for: (A) OFF comparator. (B) ON comparator. (C) Peak firing rate of the OFF and ON comparators for different input force frequencies.

Figs. 2.9a and **2.9b** show the instantaneous firing rate of the OFF and ON comparators while changing the input force-frequency over a half period respectively. The dashed lines show a good fit with a cosine function, as an evidence of the dependence of the output instantaneous firing rate on the derivative of the input signal. Furthermore, **Fig. 2.8c** shows the peak of the instantaneous firing rate with the mismatch between the two comparators. It shows the linearity with the input force-frequency (i.e. rate of change) The same applies for a trapezoidal waveform, as shown in **Figs. 2.10a** and **2.10b**.



(A) (B)

Figure 2.10: (A) Instantaneous frequency of the trapezoidal input for the OFF comparator over a positive half period with different raising edges duration. (B) Peak firing rate of the OFF comparator for different raising edges slopes.

2.12 Discussion

Our simulations show that, as expected, the instantaneous firing rate of the proposed event-driven POSFET readout encodes for the rate of change of the input force applied to the piezoelectric material. Concerning the circuit based on the LIF neuron [30], the proposed implementation based on the change detector circuit from the DVS sensor improves the sensitivity to lower input forces, ranging from 0.2 N (instead of 0.3 N).

Moreover,the absence of an integration stage – needed in the LIF based implementation – limits the latency of the communication (i.e 0.7s compared to approximately 6 seconds for the first spike in [32]) because the design avoids the current integration into the LIF membrane capacitance over time and the circuit response time depends also on the sensitivity threshold. This is accompanied by a decrease of the instantaneous firing rate of the circuit in response to the same input [32], leading to lower power and bandwidth consumption, coupled with high temporal resolution and low latency.

Besides, simplifying the design of the comparators improves the compactness of the circuit. Concerning the area saving, the area of the new version is $62 \times 84\ \mu m^2$ compared to $68 \times 178\ \mu m^2$ in [33].

Chapter 3

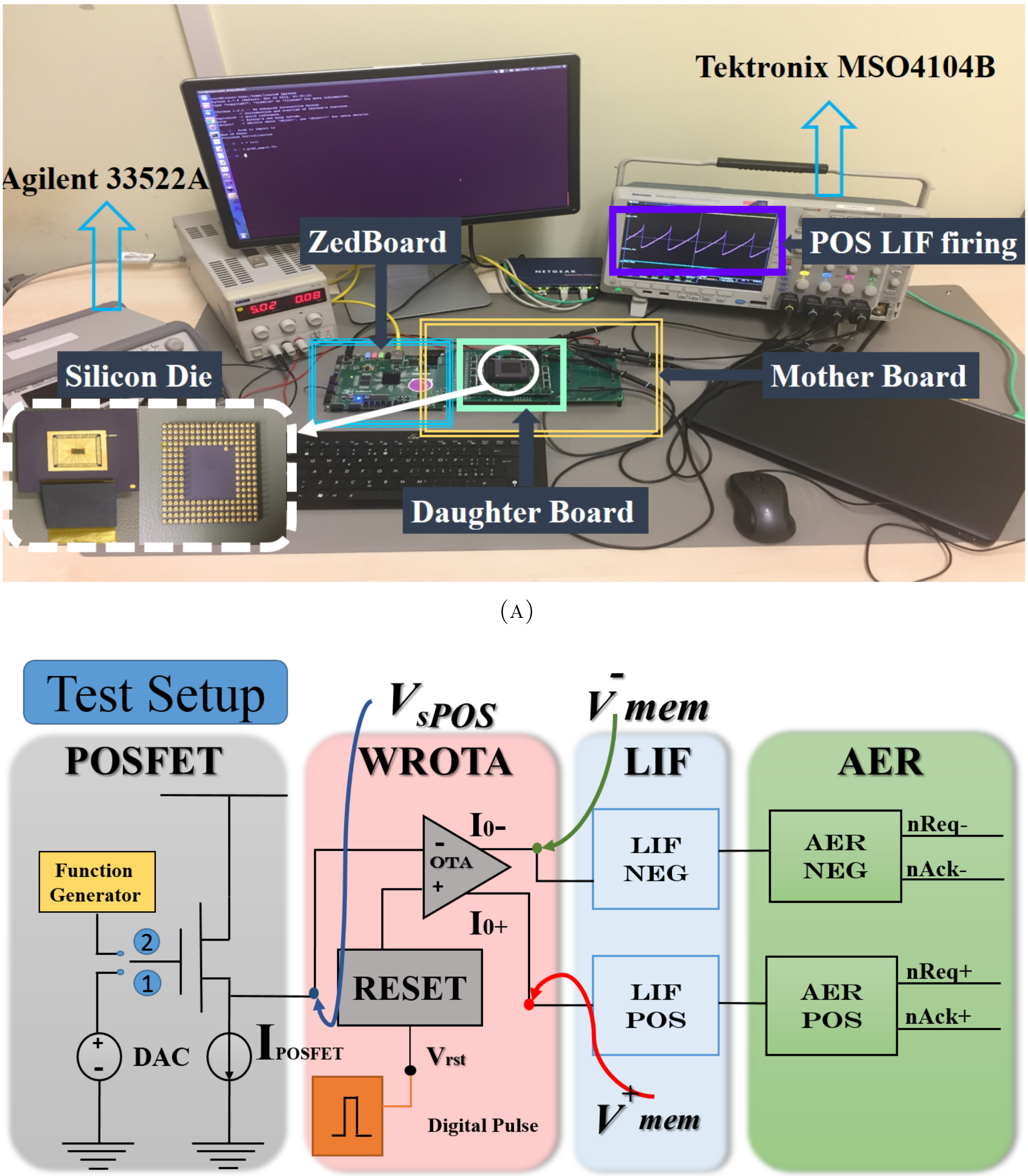
Neuromorphic Tactile Sensing Circuits: Experimental Characterization

Done with the simulation part, this chapter comes to complete the characterization of the sensor circuits. The experimental characterization part is divided into two parts: electrical and electromechanical. While in the first part, the stimulus is considered to be either a fixed potential difference input using a DAC or an arbitrary waveform through a function generator, the second part considers the real PVDF-TrFE material as a stimulus for the sensor circuit.

3.1 Electrical Characterization

3.1.1 Experimental Setup

Fig.4.5a shows the experimental setup used for the electrical characterization of the sensor circuit along with the block diagram of the circuit. The test chip is mounted on a daughter-board as an interface between the chip and the mother-board. The latter is equipped by a 54 digital-to-analog converters (DACs) that set the biases needed for the correct functioning of the circuits included inside the test chip. Besides, it provides the daughter-board with the required power supply (1.8 V in this case). Using a Linux operating system, the computer communicates with the motherboard through the Linux based Zed-Board in which the commands are sent using Python scripts utilizing an Ethernet connection.



(A)

(B)

Figure 3.1: (A) Electrical characterization experimental setup with the device under test (in white) mounted on the daughter-board (in yellow) connected to the Zed-Board (in cyan), the Tektronix MSO4104B as an acquisition tool and the Agilent 33522A arbitrary waveform generator. (B) Electrical characterization experimental setup: the device under test is characterized by a ground truth stimulus generated by the programmable function generator (that substitutes the PVDF-TrFE input). The output of the source follower (V_{sPOS}) and LIF (V^{+}_{mem}, V^{-}_{mem}) are measured by means of the oscilloscope.

The VXI-11 protocol is an RPC-based communication protocol primarily designed for connecting instruments (such as oscilloscopes) to controllers (e.g., PCs). Using this protocol as a means of communication with the oscilloscope, the Python scripts ensure the synchronization of the signal generation and the acquisition of the output with an Ethernet-controlled oscilloscope (Tektronix MSO4104B) (see **Fig. 4.5a**). Using the scripts and the communication protocol, it is possible to control the oscilloscope during the experiment and acquisition phase such as setting the acquisition length, the sampling frequency, the triggering modality, and many other available options the experimental procedure needs and the oscilloscope is equipped with. In this characterization part, the piezoelectric material is excluded and replaced by either a constant voltage source through a DAC and an arbitrary waveform generator (Agilent 33522A 2-Ch, 250 MSa/s shown in **Fig. 4.5a** as stimuli for the static and dynamic analysis respectively. **Fig. 3.1b** shows the test setup with the block diagrams of the sensor circuit together with the probed signals.

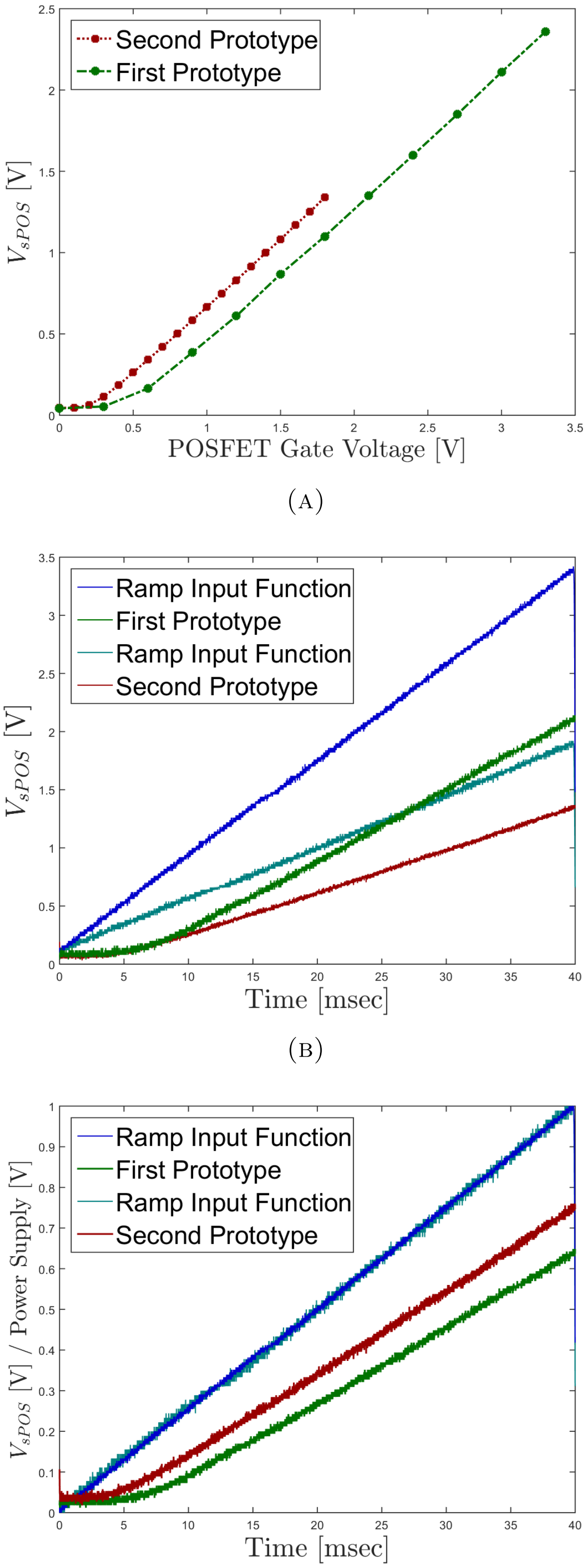
3.1.2 Experimental Results

This characterization is done by applying a constant controlled voltage to the gate of the POSFET excluding the piezoelectric film. A large NMOS transistor, in common drain configuration with its gate, is pinned out of the chip. Its gate voltage is set by providing a DAC from the ZedBoard through the motherboard to the daughter-board with specific values spreading over the entire range from 0 V to 1.8 V. The parts characterized within the coming subsections follow the main parts of the sensor(POSFET and WROTA) at first and then the whole sensor.

3.1.2.1 POSFET Characterization

With the availability of both fabricated chips produced in AMS 180-nm and 350-nm processes, the POSFET is characterized separately and deeply to understand its behavior concerning constant voltages and ramp signal inputs across its input gate from one side and to show the improvements achieved in the new designed one from the other side.

Due to the new technology (i.e. AMS 180-nm) where the input gate voltages are half compared with the previous one (i.e. AMS 350-nm), the bias current set by the enhanced cascoded current sink is 100 nA while it was 1 μA in the previously designed prototype. This biasing and the diode-connected configuration of the MOS transistor of the POSFET ensure that the POSFET operates in strong inversion and is always in the saturation region. Being a common biasing current with the WROTA, both of them need one order of magnitude less to be biased and drive the whole sensor to function properly, which serves in low power consumption as they constitute almost two-thirds of the whole sensor circuit.



(A)

(B)

(C)

Figure 3.2: Electrical Characterization: (A) Source follower output voltage (V_{sPOS}) for a constant electric signal applied to the gate of the POSFET. (B) The output voltage of the POSFET (V_{sPOS}) for a ramp input signal applied to its gate (C) The gain of both POSFET implementations where this figure extends from both traces in **Fig. 3.2b**. The latter are normalized by dividing them with the power supply (i.e. Vmax). Experimental results are shown for the device under test (green) and for a previous implementation (red) for comparison [2].

Fig. 3.2a shows the transfer functions. The DAC provides the input gate voltage over the entire available range and the output is recorded accordingly. The usual response of a common drain configuration is shown where the non-linearity is present at small input voltages due to the biasing current and the linearity between the input and the output is clear afterward where the gain factor multiplied by the input voltage dominates compared to the biasing current effect.

Fig. 3.2b shows the instantaneous response of both POSFETs to ramp signals of 25-Hz, 1.8 Vp-p and 3.3 Vp-p with an offset of 900 mV and 1.75 V from a function generator to the input of the POSFET. The experiment is controlled by a Python script setting the acquisition length to 40 ms and triggering when the output of the POSFET drops below a specified value. **Fig. 3.2c** extends from **Fig. 3.2b** where each acquired value is divided by the maximum voltage supply for both prototypes. The new POSFET with the new technology shows a better gain factor and thus better performing its role in sensing input signals and copying them to its output.

3.1.2.2 WROTA Drift with Time

By considering the previous sensor version that suffered from the drift across its analogue memory capacitance connected to the positive input terminal of the WROTA (i.e OTA+) [5], the same procedure considered is replicated to detect if this problem is solved with the new modifications applied to the offset cancellation circuit in the modified prototype.

Four different input signals are applied to the gate of the POSFET and the evolution of the memory capacitance voltage in blue along with the output of the POSFET (i.e OTA-) in red over time are shown in **Fig.3.3**. The acquisition was performed for 1000s imposing the output bias voltage of the POSFET stage at the beginning and then observing the evolution of the memory voltage after the reset pulse (V_{rst}). For each input bias point, the voltage drifts with a pseudo-exponential slope to a value of about 0.6 V with a time constant of about 300s. Scaling down of the technology to half leads to the saturation of the WROTA+ to half the value of that in the previous technology (i.e. 0.6 V). As the problem is still present, an extra effort has to be exerted to find the possible causes and take the appropriate steps accordingly.

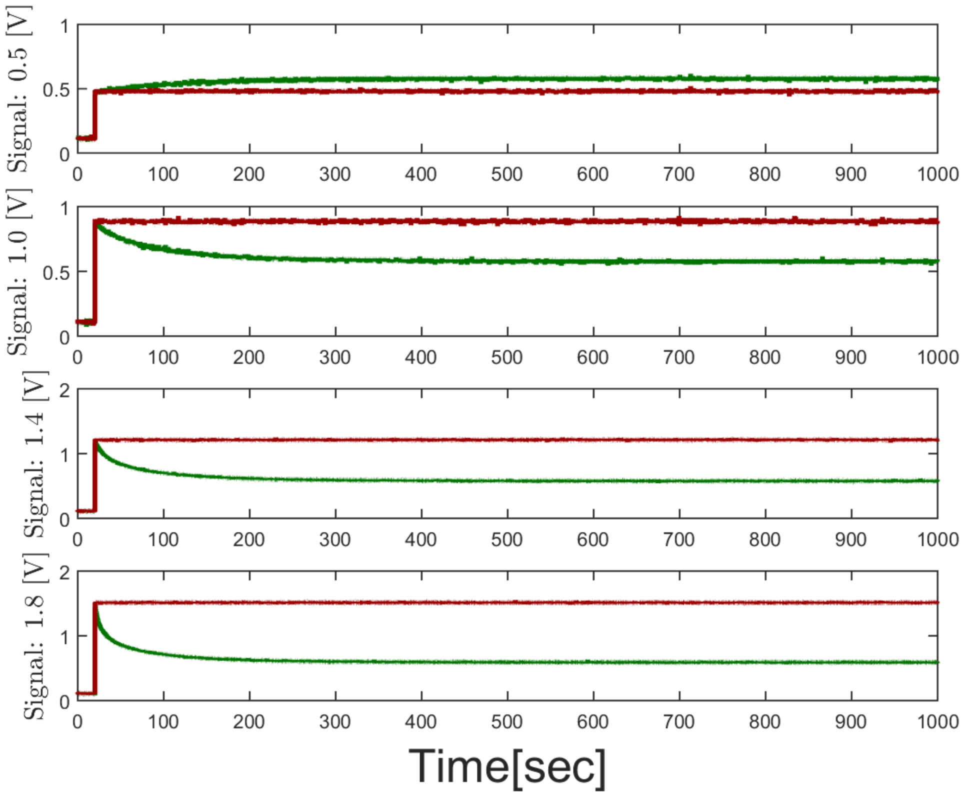


FIGURE 3.3: Voltage drift across the analogue memory capacitance (i.e. WROTA$^+$).

3.1.2.3 WROTA Offset

Due to the modifications mentioned above concerning the WROTA, here comes a test measurement across the latter to show the difference in the offset between both versions. Using a Python script, the input gate voltage is set to a specific value starting from 0.1 V to 1.8 V and 3.3 V depending on the circuit included inside the chip under test. In addition, the two LIFs circuits are disabled in such a way that the NMOS transistors gate biased by 0 V and PMOS with either 1.8 V or 3.3 V. For each input value, the LIFs membrane voltages are recorded after the oscilloscope is programmed to trigger once the input voltage crosses a value greater than a fixed value over. The results are shown in **Fig. 3.4** where both are normalized by dividing each acquired value by its corresponding power supply value for direct comparison. As shown, both negative terminals of the OTA share the same offset, however, there is an improvement concerning the positive terminal in the new sensor version compared to the previous one where the offset is lower.

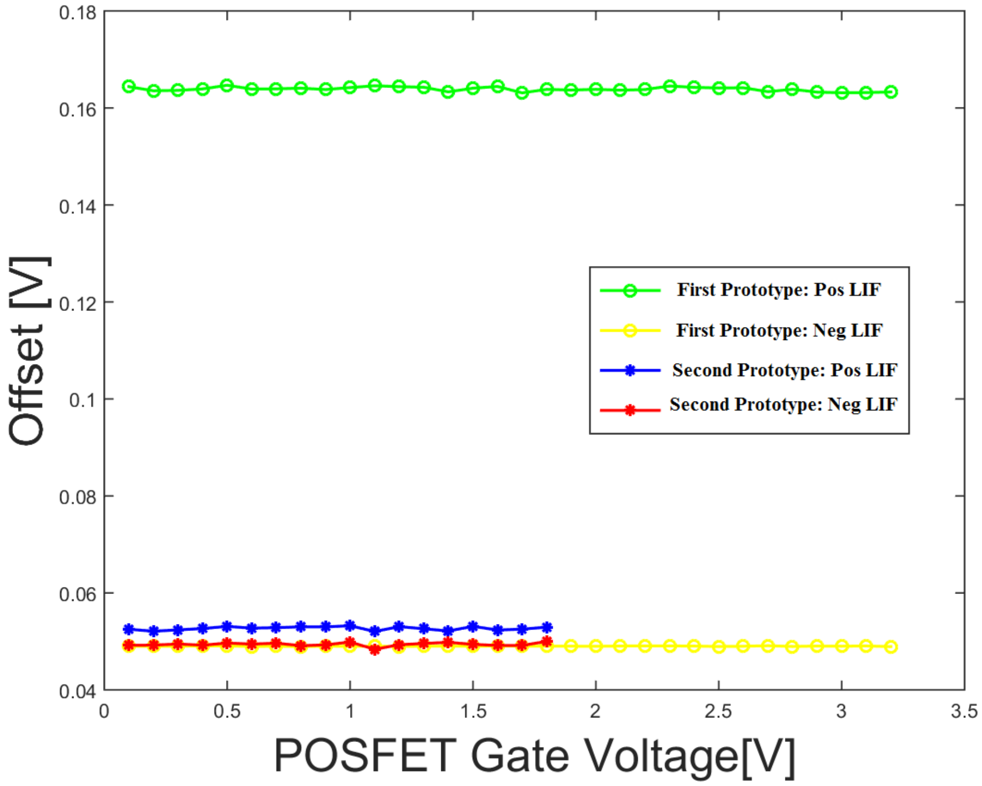


FIGURE 3.4: The WROTA offsets for both prototypes.

3.1.2.4 Sensor Circuit Transfer Function

Fig. 3.5 shows the overall transfer function of the sensor circuit by measuring the mean output firing rate with its corresponding standard deviation of both neurons; where a constant voltage is applied to the gate of the POSFET from three different chips. Each measurement was performed 10 times for every value of the input voltage to the gate of the POSFET. We did the measurements with the help of a Python script that sets the values of the biases specifically the input gate voltage to the POSFET and the triggering mode, collects the data from both neurons outputs and saves them in a text file, and calculates the mean firing rate after calculating the inter-spike interval and finally displaying the plots. The biases of the circuit constituents were tuned to have the most coherent transfer function among the various instances, obtaining a trade-off between sensitivity and linearity.

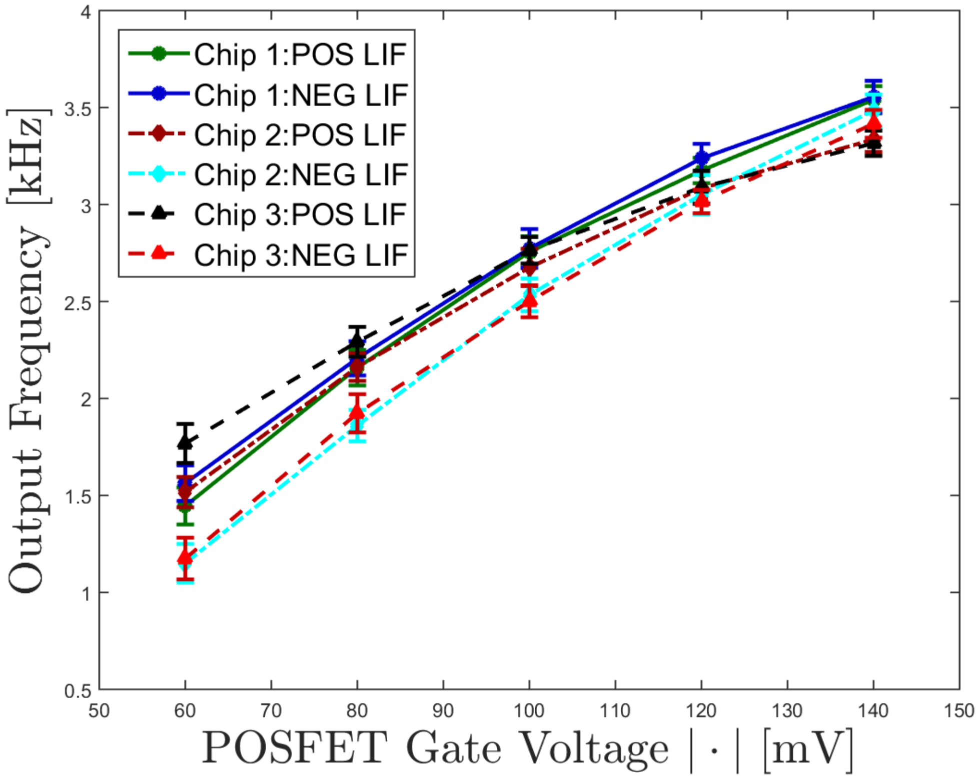


Figure 3.5: Output firing rate of both neurons versus constant input voltages from three different chips.

The transfer function shows an expected linear behavior of the output firing rate of both neurons concerning the applied voltage at the POSFET gate. Compared to[40], the firing rate in the newly designed sensor circuit is one order of magnitude smaller. This in return results in a lower bandwidth, power-saving and better sensitivity to events under which the neuron is triggered to generate a spike as a sign of either pressure or release on or from the POSFET respectively.

3.1.3 Electric Characterization: Dynamic Analysis

In this part, we aim to show the dynamical behavior of the sensory circuit response. With this kind of analysis, the input to the gate of the POSFET is sinusoidal and square waveforms supplied by an arbitrary function generator (i.e. Agilent 33522A 2-Ch, 250 MSa/s) that handles also the bias voltage. The response of the POSFET output (V_sPOS), the positive neuron membrane potential (V^+_{mem}) and the negative one (V^-_{mem}), is recorded and plotted versus a square and sinusoidal waveforms in blue, green and gray as shown in **Figs. 3.6** and 3.7 respectively.

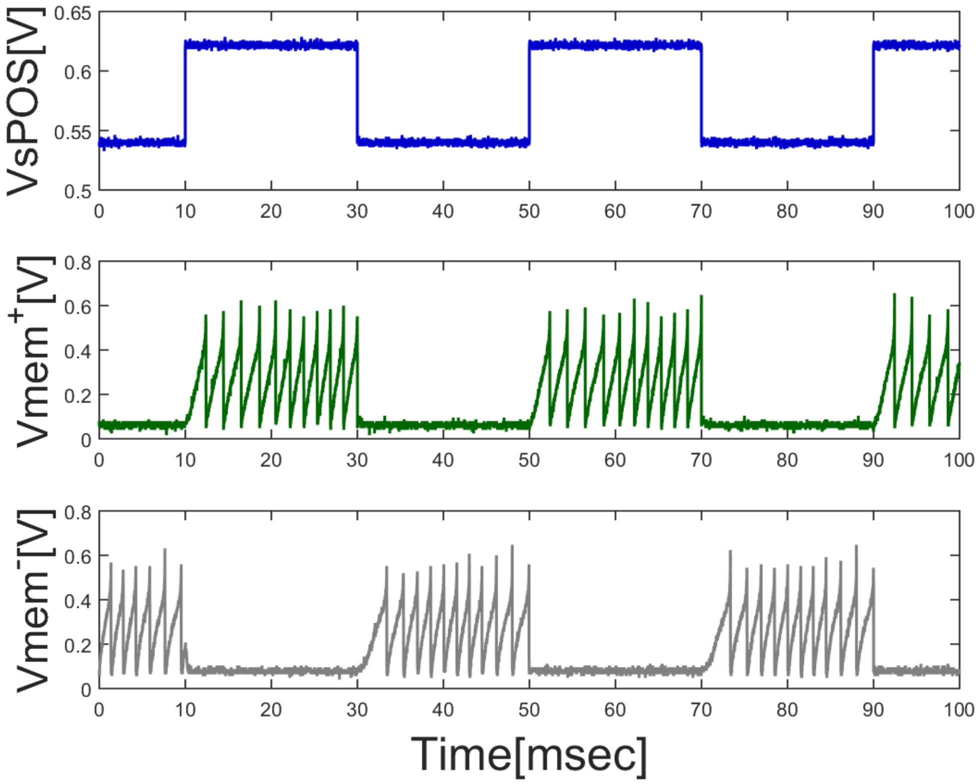


FIGURE 3.6: The response of both LIFs to a square input signal of 25-Hz and 80 mV p-p.

The resulting plots show the constant output firing rate of the LIFs and its proportionality to the input voltage. In the intervals when the input signal is constant in value, one of the neurons is driven to generate spikes while the second remains silent. This proves the proportional behavior of the sensory circuit to the voltage applied to the gate of the POSFET employing a time-varying waveform.

Compared to what is achieved in [40], the sensitivity to the input signals has improved. While the smallest sensed signal reached was of 600-Hz and 100 mV p-p, the smallest achieved sensed signals in the case of the square input signal and sinusoidal are of 25-Hz and 80 mV p-p and 90 mV p-p respectively. These two results confirm the ability of the new sensor to sense small-amplitude input signals and drive both neurons to generate spikes accordingly.

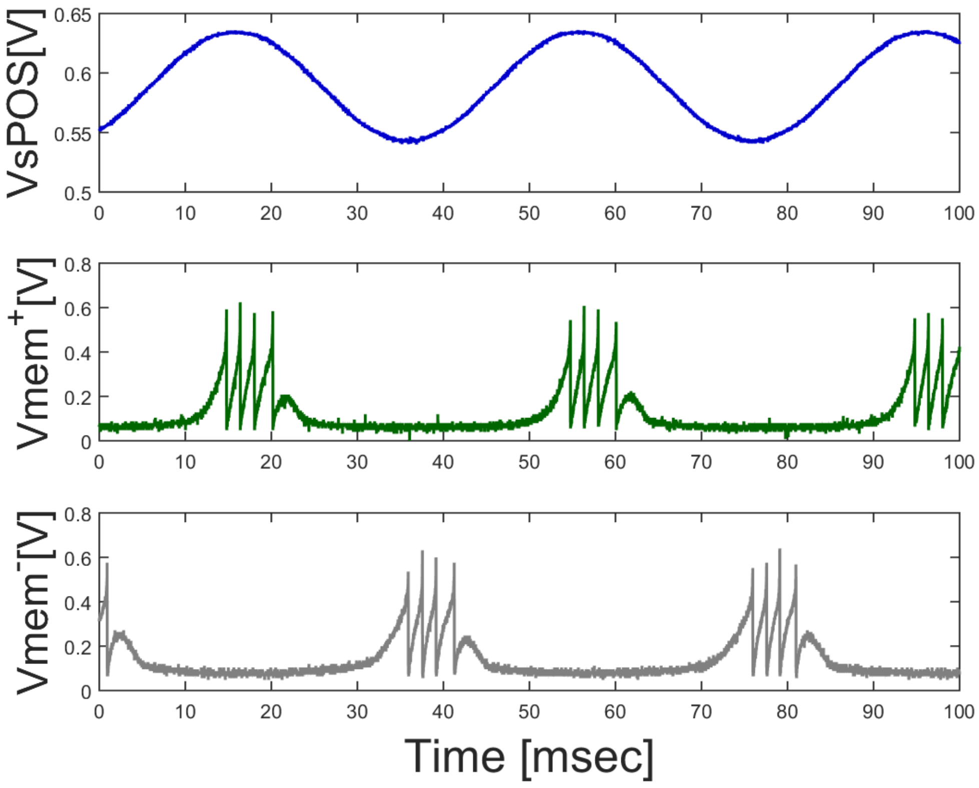


FIGURE 3.7: The response of both LIFs to a sinusoidal input signal of 25-Hz and 90 mV p-p.

3.2 Experimental Setup for Electro-Mechanical Characterization

This section describes the electro-mechanical setup for the characterization of the sensor circuit where the stimulus is replaced by the real piezo-electric material (i.e. PVDF-TrFE) instead of the DAC or the arbitrary waveform generator. The experimental set up developed for testing the tactile sensing chip shown in **Fig. 3.8**. This set up is intended to test the tactile sensing chip for the forces applied in the normal direction.

The piezo-film is 1×1 cm^2 rectangle of metalized PVDF-TrFE film. With a thickness of 110 μm electrodes of Cu-Ni, which is about 700 $\mathring{A}$ (thickness) of copper covered with 100 $\mathring{A}$ (thickness) of Ni-Cu alloy. With this configuration, the film allows considering its mechanical action negligible for the sensor electromechanical modeling. To drive the sensor through the output signal of the film, one face of

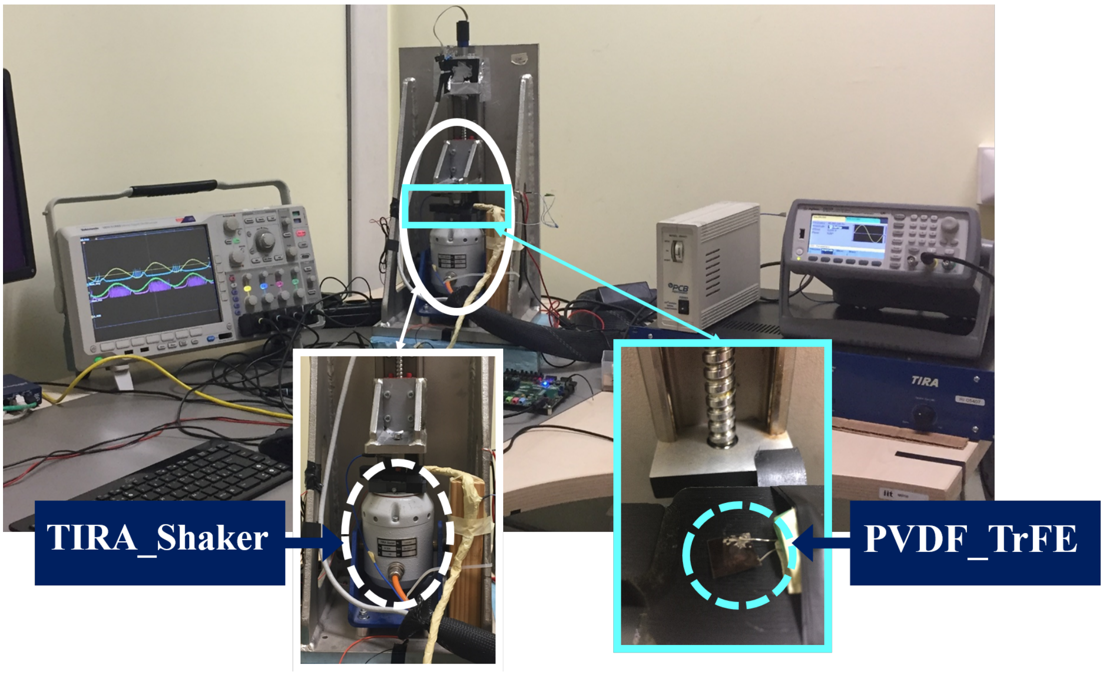


Figure 3.8: Electromechanical characterization setup: The PVDF-TrFE is shown in the part and the interchangeable impact tip with the TIRA shaker (in red), the ICP force conditioner and TIRA amplifier and the chip with the daughter and motherboards.

the film is connected directly to the pinned out gate of the POSFET and the other face to a pin supplied by a DAC that sets that handles the biasing of the POSFET through two shielded cables, glued to the top and bottom electrodes, using a conductive glue (Chemtronics Silver 0.25 oz Tube Epoxy Conductive Adhesive). Such connection provides the extended gate POSFET configuration (see **Fig. 3.8**) (for further information about the POSFET, refer to [40]).

The film is stimulated by the TIRA shaker (Model TV50018), labeled in **Fig. 3.8** which is driven by TIRA power amplifier Type BAA 60. The latter is provided by the input waveform (Agilent 33522A 2-Ch, 250 MSa/s). The shaker can apply random dynamic forces up to 18 N with frequency in the range 2 Hz - 18 kHz. The force generated by the shaker is measured by a piezoelectric load cell (Model–208C01, from PCB Piezotronics of ICP type) which is fixed on a platform that can move along the z-axis. The load cell has a sensitivity of 108.7 mV/N. The latter is displaced so that the interchangeable impact tip is touching the film making sure that the mini-shaker does not lose contact during the stimulation, thus the preload has to be at least as large as the maximum amplitude of the applied force.

A Python script is used as before to set the biases of the circuit under test where the digital reset signal was set low immediately before the beginning of the tactile stimulation. Also, the Ethernet-controlled oscilloscope (Tektronix MSO4104B) was considered to be the tool for the acquisition of the film output, POSFET output, and both neurons membrane potentials.

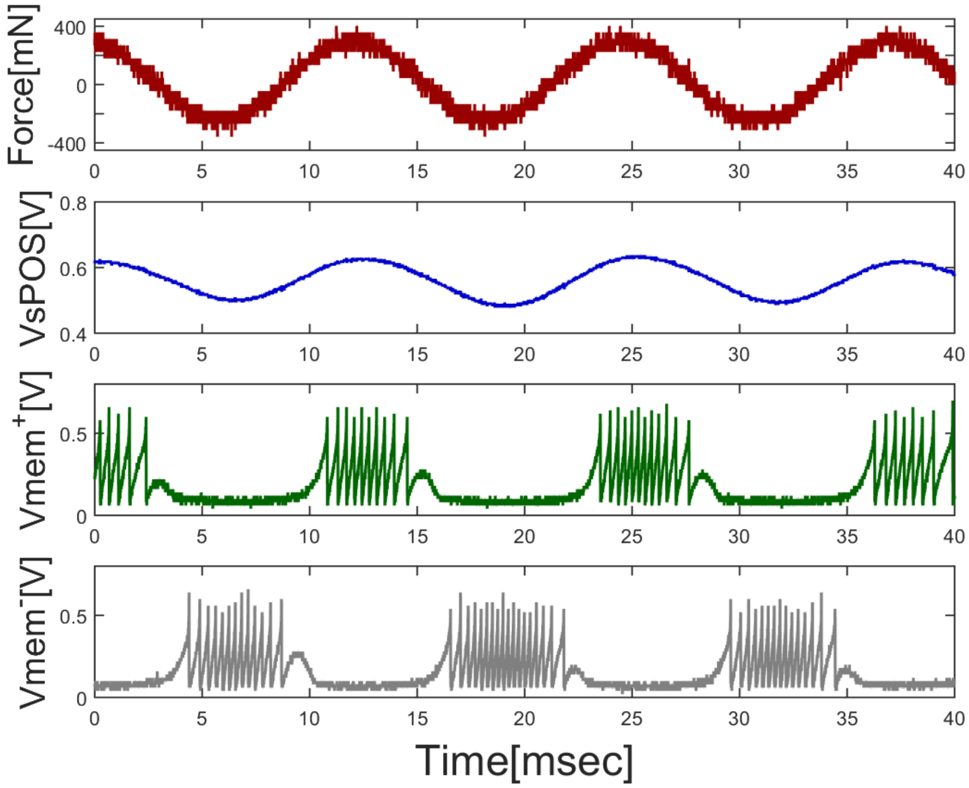


FIGURE 3.9: Electromechanical characterization: Response of the main circuit to an 80-Hz sinusoidal force 313 mN of magnitude applied to the PVDF-TrFE film.

3.3 Electro-mechanical Results

In this section, we introduce the experimental results of the electromechanical characterization of the neuromorphic POSFET with the real piezo-electric material being used as a stimulus. Using the mini-shaker described above, we applied a sinusoidal waveform with different amplitudes and recorded the response of the circuit parts accordingly.

Fig. 3.9 shows the response of the circuit when an 80-Hz sinusoidal force 313 mN of magnitude is applied to the PVDF-TrFE film. The top graph in red represents the force applied to the film, the one in blue represents the output voltage of the POSFET following proportionally the slope of the input force, the response of the positive neuron (V_{mem}^{+}) and the negative neuron (V_{mem}^{-}) in green and gray respectively. The membrane potential of the LIF neurons increases with the increase of the input integrated current, and then it generates an action potential when it crosses a tunable threshold. Both neurons are silent for stimuli below their activation threshold. This observation in return is an additional confirmation of the proportional behavior of the circuit when the film is used as a stimulus.

A difference between the firing rate of both neurons is observed. In addition to the mismatch between the output positive and negative branches of the WROTA that provide both neurons with a current to be integrated over time, the mismatch between the positive and negative neuron circuits, the difference in their spiking thresholds and leakage are responsible for the noticed difference. Moreover, the mean output firing rate of the neuron is lower than in the previous implementation. **Fig. 3.10** shows the system transfer function of the positive neuron over a positive half period of an 80-Hz input sinusoidal. The measurements at nine different input force values are done, and the number of spikes generated by the neuron is recorded from which the frequency is calculated for two different chips. The mean output frequency and its corresponding standard deviation from six trials are plotted versus the input force values.

Increasing the input force yields more current integrated into the neuron membrane over time and thus leading to the higher output frequency. The linearity between the firing rate and the input force on the film is again obvious. Also, the difference in the output of both circuits is also due to the mismatches discussed before.

Table 3.1 summarizes the main improvements that the new sensor version (i.e. First Prototype) achieved compared to the previous prototype (i.e. Second Prototype) [40] in simulations and measurements performed.

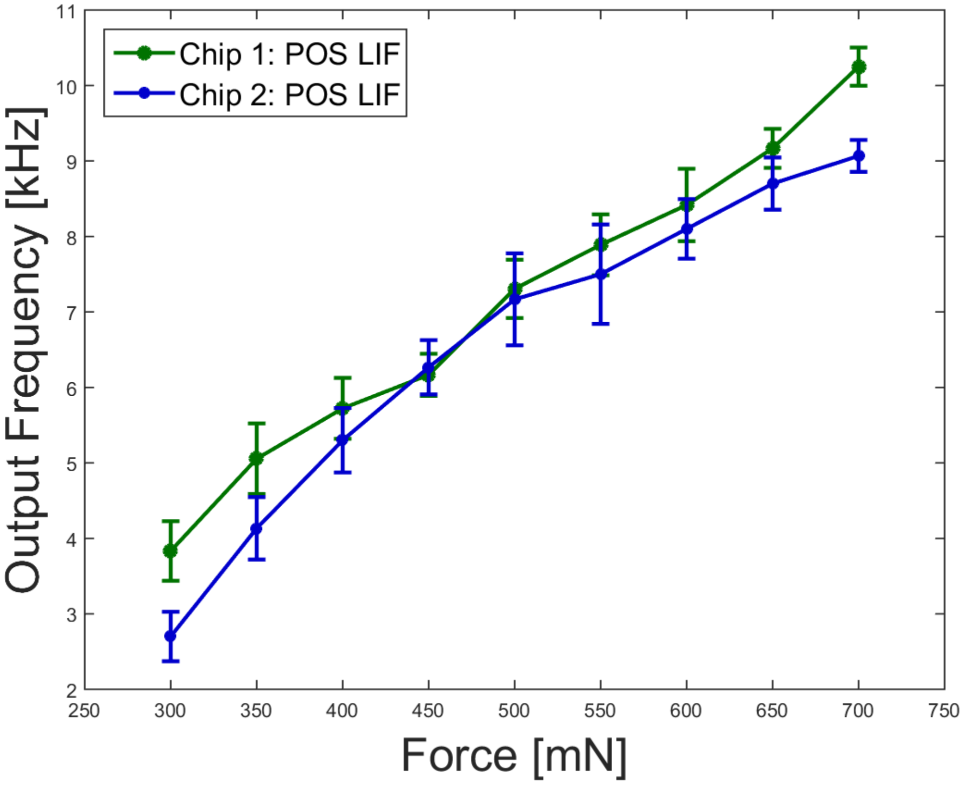


Figure 3.10: Electromechanical characterization: Circuit transfer function: The firing rate of the positive neuron over a half positive period for different input force values.

Table 3.1: **Comparison between Both Fabricated Prototypes**

	First Prototype[40]	Second Prototype
Technology	AMS CMOS 350 nm	AMS CMOS 180 nm
Circuit Size	75 × 227 μm^2	68 × 178 μm^2
POSFET Size (W,L)	(1250 μm, 2.5 μm)	(651 μm, 1 μm)
Bias Current	1 μA	100 nA
Max Output Frequency	33 kHz	3.5 kHz
Input Square Signal	100 mV	80 mV

Chapter 4

Edge Orientation Detection

Future robots are expected to work closely and interact safely with humans as well as real-world objects. Among various sensing modalities needed for this purpose, the sense of touch is particularly important. Unlike other senses (e.g., vision,audio), it involves complex physical interaction, and plays a fundamental role in estimating properties such as shape, texture, hardness, material type and many more. Such properties can be better estimated by touching or physically interacting with the objects—as humans do. The sense of touch also provides action related information, such as slip, and helps in carrying out actions, such as rolling an object between fingers without dropping it. Therefore, tactile information is needed during trajectories like manipulation and exploration [44]. Thus, the interactive way of robots with real-world objects is an important issue - as such interactions depend on how heavy or light and smooth or hard the objects and in which direction it is moving across the sensors. Acquiring the proper information through the acquisition system in time and space lead the robot to accurately perform the trajectories and missions assigned to it in a proper way.

Edge orientation detection of moving objects is one of the hot topics these days for the impact of the latter on the behavior of artificial devices in controlling the movement, activating the useful and needed sensors for enabling the device to precisely control and manage the interactive way with the sliding edge in different parts of the device [45].

In the following sections, I will include the work done for detecting the orientation of a sliding object on a complete skin patch of capacitive sensors of the iCub robot using a machine learning technique.

4.1 Methodology

To achieve the desired goal, the work is divided into three stages (see **Fig. 4.1**). The work combines electronics through the acquisition system, mechanics by using and controlling the Omega.3 device, and programming through ROS software [46] using C-language and Brian, a module in Python as a simulator for neural network, and MATLAB using machine learning technique.

The skin patch was stimulated by sliding a tip at different speeds ranging from 1.27 cm/s to 17.873 cm/s and forces from 0.4 N to 2.1 N. The output of each taxel is the analog value of the capacitance and is sent every 10 μ. The conversion of this value to a spiking encoding by means of a software implementation of LIF neuron using Brian simulator. The analog value is treated as an input current injected into the membrane of the neuron. The resulting spike trains are fed to a machine learning pipeline in MATLAB. The algorithm is composed of offline and online parts. The former serves for training of the dataset and the latter decides in which direction and angle the object is sliding over the skin patch. As a result, it classifies the trajectory of the tactile input.

With the presence of Brian simulator (i.e. a simulator for neural spiking neurons), the latter paves the way to test the feasibility of a neuron model that mimics its biological behavior on software side. So, I used the simulator to characterize the mathematical model and try to match its membrane profile to that in [2] and [43] by omitting the adaptation part. To bridge between the real setup and the simulation part, I did a one-to-one mapping between taxels in the real setup and the neurons set in Brian in a way that each neuron holds the same address and position of its corresponding taxel and is fed by a scaled current from it. By this way, we coupled what is in software with what is available in hardware.

Fig. 4.1 block diagram summarizes the workflow followed to achieve the desired goal.

4.2 Experimental Setup

The experimental setup used for collecting the responses of the taxels is made up of three main parts: electronics, mechanics and software (see **Fig. 4.2**).

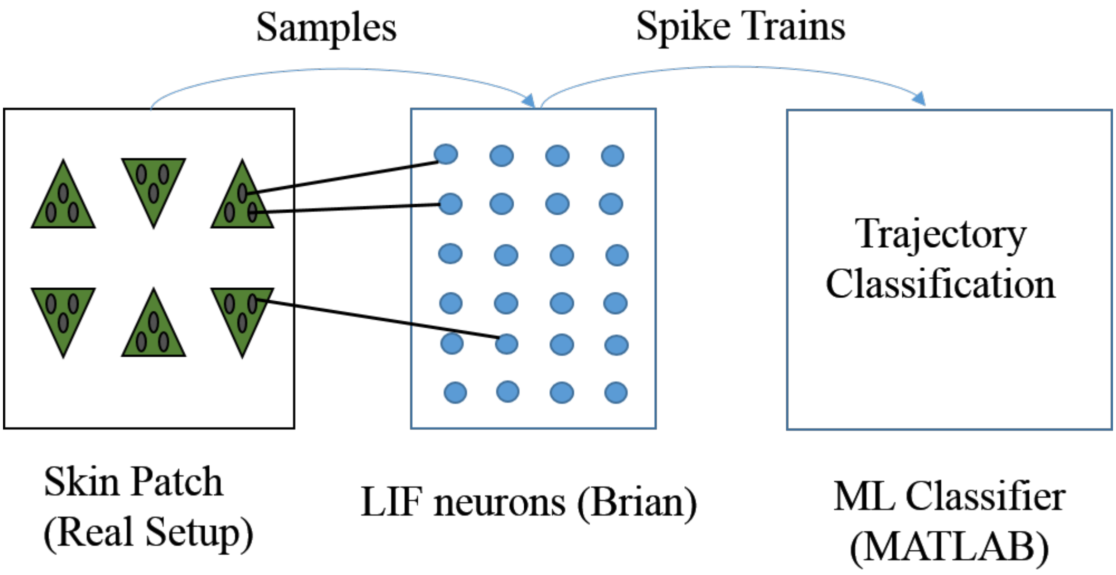


FIGURE 4.1: Work Flow.

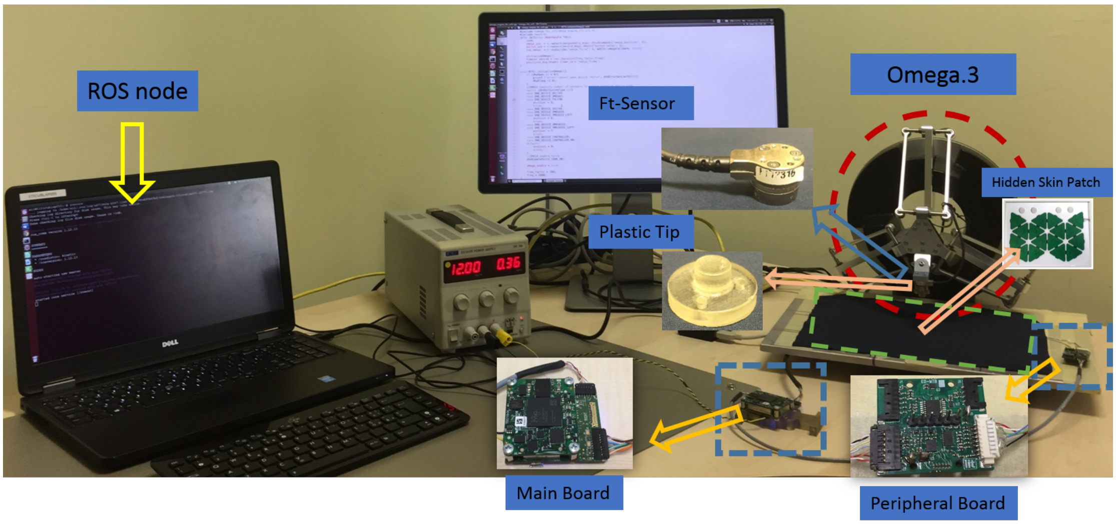


FIGURE 4.2: Elements of the Experimental Setup.

4.2.1 Electronics

Fig. 4.2 shows the skin patch during the experiments. It consists of 16 triangles, each containing 10 sensitive areas (see **Fig. 4.3**). The patch is mounted on a rigid support with holes in correspondence of the readout chip, allowing for a planar setup where electronics cannot be damaged (see Fig.4.3)

The capacitive skin sensors are sampled by a capacitance-to-digital converter (Analog Devices AD7147) every 36 ms. The 16 bits digital samples are transmitted via a shared clock 4-channel I^{2}C interface. The peripheral board receives this data

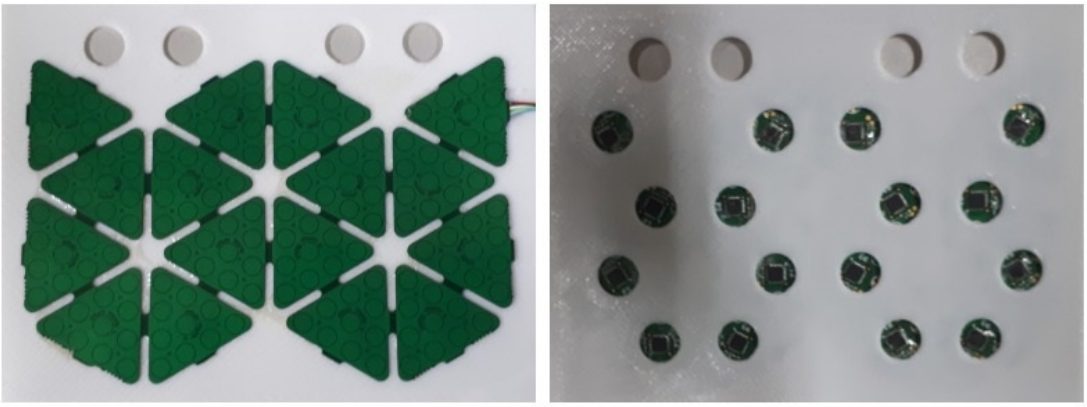


FIGURE 4.3: Diagram of the skin patch considered for doing the experiments and acquiring the dataset from: front and back parts.

and retransmits them in two subsequent 20-bit data packets, containing half data each (see **Fig. 4.2**). The peripheral board issues a Xilinx Artix-7 FPGA (Part n. XC7A35T-L1), which is programmed to rearrange the data and transmit them to the mainboard via a custom high-speed serial protocol [47].

The main board, shown in **Fig. 4.2**, issues a Xilinx Zynq-7000 SoC, which receives the data and timestamps them. The main board communicates with the peripheral one by a single-channel I^{2}C interface. This is used to configure the registers available in the hardware of the Xilinx Artix-7 FPGA. Moreover, the Zynq-7000 Soc has an Ethernet port to communicate with a PC and runs a Linux-based operating system. Therefore, a high-level user can connect to the Xilinx Zynq-7000 via Ethernet and configure, using Linux I^{2}C utilities, the registers of the peripheral board and read their values back.

4.2.1.1 Mechanics Part

The mechanical part includes the use of an Omega.3 device, shown in **Fig. 4.2**, that carries the object that is aimed to slide the patch. The role of the device is to make the object (the tip in our case) to scan the patch from different end to end and control its movement (i.e. the direction) with defined speeds through the horizontal force F_y and the pressure applied on the patch through the vertical force F_z (setting their values will be clarified in the **Results** section). All of this is done using the ROS software (that will be highlighted later) through a C script using a Qt creator editor in Ubunto-Linux based operating system. Through the C script using Qt Creator, the position of the tip mounted on the FT-sensor attached to the omega.3 is set between two endpoints. The chosen tip has a diameter that is good enough to fully cover and press the underlying taxel and partially the adjacent taxels (immediately next to the pressed ones).

4.2.1.2 Omega.3 Device

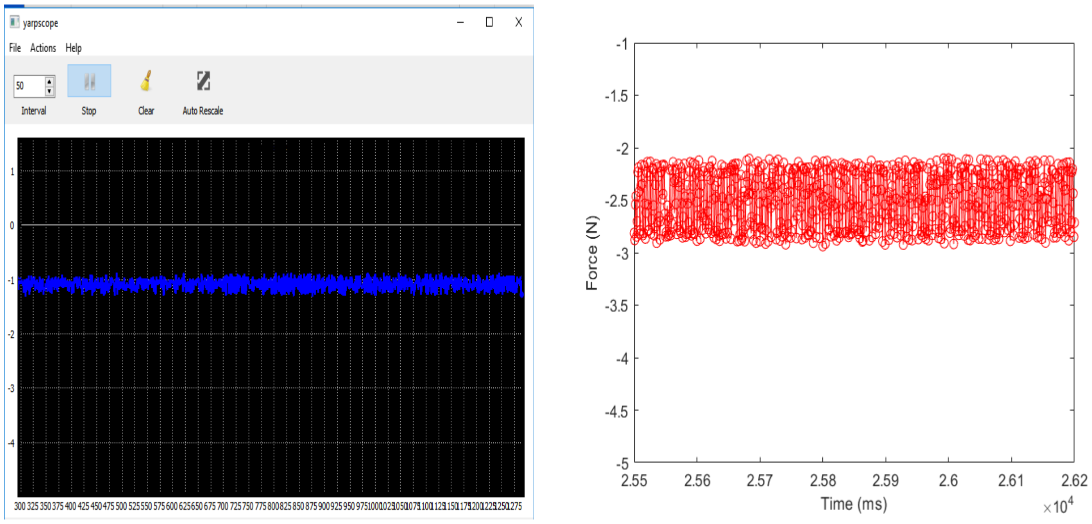
Shown in **Fig.4.2**, the Omega.3 relies on a unique kinematic design that has been optimized for high-end force feedback. It is specifically designed for demanding applications where performance and reliability are critical. Perfect for use within robotics, virtual simulation and nano-manipulation environments amongst others, the omega.3 warrants looking when considering haptic devices. A powerful SDK provides highly advanced control for the omega.3 in both haptics and robotics mode. The Haptics SDK offers all the basic functions to read positions and to program desired forces in Cartesian space. Expanding these fundamental capabilities, the Robotics SDK leverages the Haptic SDK by introducing an advanced set of real-time routines to precisely control the position of the device. The combination of both haptics and robotics capabilities into a single unified framework allows developers to create powerful collaborative interfaces between people and machines. This device is used in a wide range of applications including: medical and space robotics, micro and nano manipulators, virtual simulations and training systems.

4.2.1.3 Force Torque Sensor: Nano17

The force torque (FT) sensor, shown in **Fig. 4.2**, used is the Nano17-E transducer. It has a list of properties that makes it a good choice for this experimental setup and goal. One of them is that it is the smallest commercially available 6-axis transducer in the world. For further information, please refer to (https://www.ati-ia.com).

4.2.2 Software Part

Done with the electronics and the mechanics part in the experimental setup, the following section shows the software used mainly in the mechanics part.



(A) Force detected by the FT sensor visualized using Yap scope.

(B) The force over time plotted using MATLAB.

FIGURE 4.4: Two examples of the membrane potential profiles.

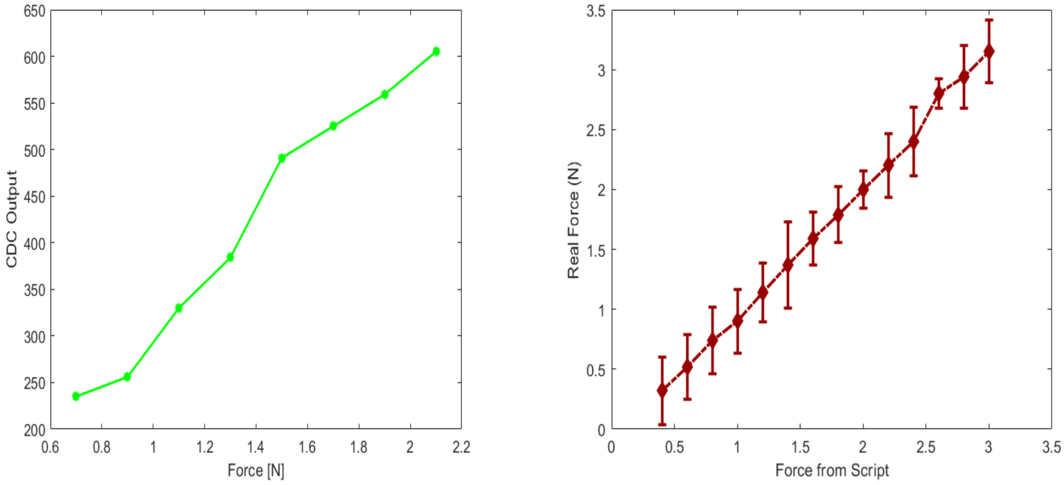
4.2.2.1 ROS Software

Robot Operating System (ROS) is a collection of software frameworks for robot software development. Although ROS is not an operating system, it provides services designed for a heterogeneous computer cluster such as hardware abstraction, low-level device control, implementation of commonly-used functionality, message-passing between processes, and package management. Running sets of ROS-based processes are represented in a graph architecture where processing takes place in nodes that may receive, post and multiplex sensor, control, state, planning, actuator, and other messages [46].

4.3 Results of the First Stage

After highlighting the different parts of the experimental setup and before going to the results, it is time to describe the steps followed to generate the coming figures.

As the forces, along the z and y-axis responsible for the speed and the pressing force respectively, are set through the C-script by numbers, the need to exactly know the correspondence between these numbers and their actual force magnitudes applied arises. The idea comes as follows. The F_y is set to a small value and the F_z is set to a large value. So the tip gets stuck to the patch at a defined position and



(A) (B)

Figure 4.5: (A) The output of the taxels with the applied force.(B) Real force sensed by the FT-sensor versus the force values set in the C-script.

unable to overcome the friction force and thus becomes stuck with the patch. By this way and with the help of the yarpscope shown in **Fig. 4.4a** through reading the F_y and F_z forces from the FT-sensor, I became able to get the real value of the force by saving the signal in a CSV file and getting its average (see **Fig. 4.4b**).

With the properties of the latter device mentioned above, the time the sliding tip takes between its path between two endpoints is calculated. Through the C-script, an initial time is set once the tip hits one end of the patch and final time when it leaves the patch. The difference between these two instants represent the duration that the ip takes to scan the patch from one end to another. This serves for the calculation of the average speed. To change the latter, the horizontal force F_y applied by the FT-sensor is changed. Concerning the pressing force of the tip on the patch, there exists a force value correspond. Through the C-script, the magnitude of the horizontal and vertical force can be changed that will modify the average speed and the pressing force of the tip accordingly.

Fig. 4.5a shows the behavior of the capacitance to digital converter as a function of the input force. The former shows a linear relation in which the increase in the force applied by the tip leads to an increase in the capacitance due to the decrease in the separation of the parallel plates. Consequently, the capacitance to digital samples increase in value as shown in **Fig. 4.5a**.

Sweeping over the force values and getting the average over the readings from the yarpscope, **Fig. 4.5b** shows the relationship between the pressing force set as numbers in the C-script and the real values sensed by the FT-sensor attached to the Omega.3 device.

4.4 Second Stage: Brian Simulator

After covering the constituents of the first stage, it is time to go ahead towards the second stage composed of the Brian simulator to clarify the properties of such simulator and the model considered in addition to the results. Besides, the following sections include the way the simulator is used to serve in achieving the global goal through the link with the first and third stage.

To achieve this, the presence of a simulator at first helps to test the feasibility of a neuron model that mimics its biological behavior as a starting phase. With the electronic system and using a Linux based operating PC, the data is acquired across each taxel and then scaled utilizing feature scaling.

4.5 Brian Simulator for Spiking Neurons: Brief Description

Brian is a fast event-driven software simulator developed for simulating large networks of spiking neurons and synapses. The primitive network elements are designed to exhibit biologically realistic behaviors, such as spiking, refractoriness, adaptation, axonal delays, summation of post-synaptic current pulses, etc. The efficient event-driven representation allows large networks to be simulated in a fraction of the time that would be required for a full compartmental-model simulation. Besides, it is an intuitive and highly flexible tool for rapidly developing new models, especially networks of single-compartment neurons. In addition to using standard types of neuron models, users can define models by writing arbitrary differential equations in ordinary mathematical notation.

Two important points are behind using the simulator. Through Brian, a neuron model can be used to simulate it. Thus, this is linked to the idea of neuromorphic engineering and to event driven approach. Moreover, having the one-to-one correspondence between the taxels and the neurons plays a vital role in the machine learning algorithm. Since the latter is based on probabilistic approach, feeding the neurons in Brian by the outputs of the taxels gives each neuron a certain probability depending on its firing rate. By other words, the neuron with higher rate will have a different probability than the one with lower one. This in return gives a higher weight of the former compared to the latter. Therefore, using the simulator output and using them in the third stage enhances the accuracy of the latter in the detection of the path followed by the tip while scanning the patch.

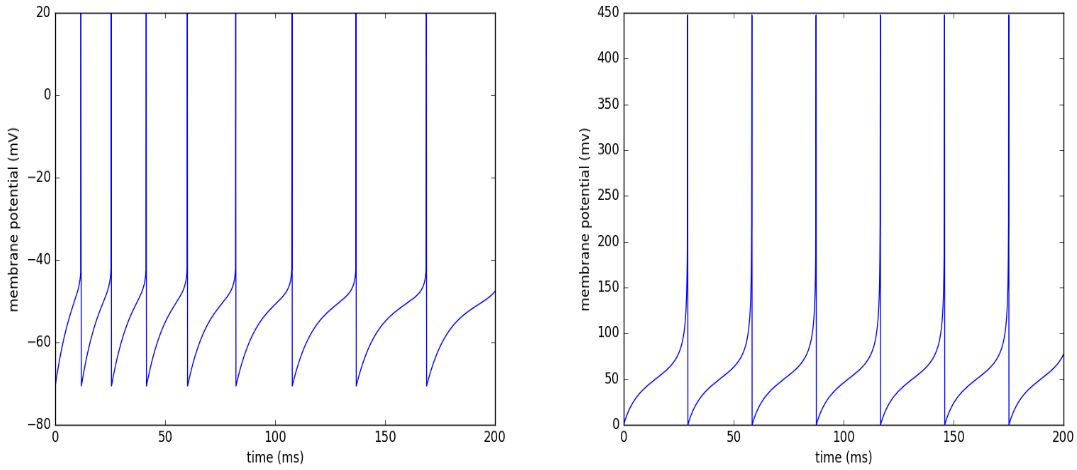
4.6 Neuron Mathematical Model: General Description

The mathematical model is built in Brian using Python object-oriented programming language. Computationally, the former is based around the concept of code generation: users specify the model in Python but behind the scenes, Brian generates, compiles and runs code in one of several languages (including Python, Cython and C++). The simulator is aimed to develop models based on networks of spiking neurons for a wide range of applications.

Users specify neuron models by stating their differential equations in standard mathematical form as strings, create groups of neurons and connect them via synapses. This approach differs greatly from many neural simulators in which users select from a predefined set of neuron models. Such general design is aimed at maximizing flexibility, simplicity and users' development time. (refer to (https://brian2.readthedocs.io/en/2.0rc/index.html).

4.7 Workflow Accompanied by the Use of the Simulator and the Model.

The work in this part started by implementing the model taken from [48]. After that, analyzing the different parameters of the model considered and comparing its output (i.e the profile of the neuron achieved by the simulation and experimental results in both chapters two and three). The first step I did is eliminating the adaptation part (see **Fig. 4.6a**) to reach the desired behavior (see **Fig. 4.6b**). By comparing both figures, the neuron in **Fig. 4.6b** shows that its membrane potential has a fixed interval of charging while in **Fig. 4.6a** this is missing where at different time intervals (say every 30 ms), the charging phase differs in its duration because of the presence of adaptation. The latter step is done by tuning the suitable list of input parameters for the model to match the response of a single (LIF) neuron [29].



(A) Neuron membrane profile with adaptation. (B) Neuron membrane profile without adaptation.

Figure 4.6: Two example of membrane potentials profiles.

Specifying the differential equation with the set of state variables and the numerical integration method makes the contents of a single or group of neurons ready to be initiated. With the capabilities of the simulator for transient analysis, the latter makes it possible to show the input current also the membrane potential over the time duration. Moreover, having the ability to save the spiking instants, spike trains associated with each neuron included within the group can be represented versus the instant the spike is generated.

Thus, after all, what is done concerning the analysis and modification phases, the membrane potential model is expressed by a differential equation shown in **Equ. 4.1**. The latter mimics its real behavior v_m, with a set of state variables such as the input current to the neuron I, reset potential E_L, leak conductance g_L, membrane capacitance C, slope factor $\triangle T$, and spike threshold V_T.

$$\frac{dv_m}{dt} = (g_L * \triangle T * exp(\frac{(vm - V_T)}{\triangle T}) + g_L * (E_L - v_m) + I)/C \tag{4.1}$$

Knowing that working with a single neuron is not sufficient for detecting the orientation of a sliding object, we extended the scope to use through the simulator a group of neurons that are capable of achieving this goal with defined indexes, spatial positions and response over time. This step helps in recording their responses over time utilizing spike trains and build on that to set them as inputs to the machine learning technique discussed latter.

4.8 Results of the Second Stage

Before going to the outputs of the mathematical model in Brian, there is a need to link between the first stage and the second concerning the output of the former and the input to the latter. Thus, **Fig. 4.7** shows the linear relation between current as input after applying the feature scaling step and the output of the taxels (called CDC). This step shows that the increase in the output of the taxels will in return gives a sign that the force increased and as a result an increase in the current fed to the neuron. That is why the relation appears as proportional.

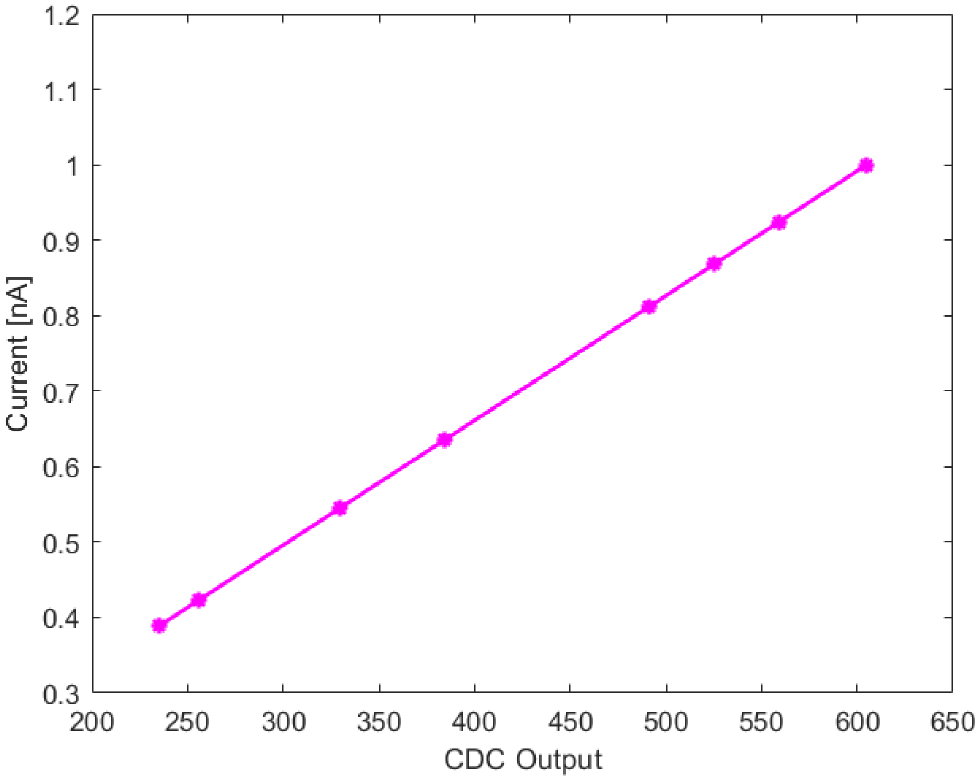


Figure 4.7: The transfer function of the first stage.

After stating the different steps for matching the behavior in the mathematical model to that in the electronic circuit model through simulation and measurement, now it is time to highlight different results coming out from the simulation using the considered model. As the model includes a group of neurons, the following results show the response of one neuron at first and a group of neurons after being fed by constant input, their outputs and the spike train corresponding over time as a second step.

Starting with a single neuron, **Fig. 4.8a** shows a constant input current fed to the neuron along with its response through its membrane potential (see **Fig. 4.8b**). The latter is considered the profile of the LIF neuron. **Fig. 4.8b** comes as result of **Fig. 4.8b** where each spike generated is associated with a digital pulse as shown in **Fig. 4.8c**.

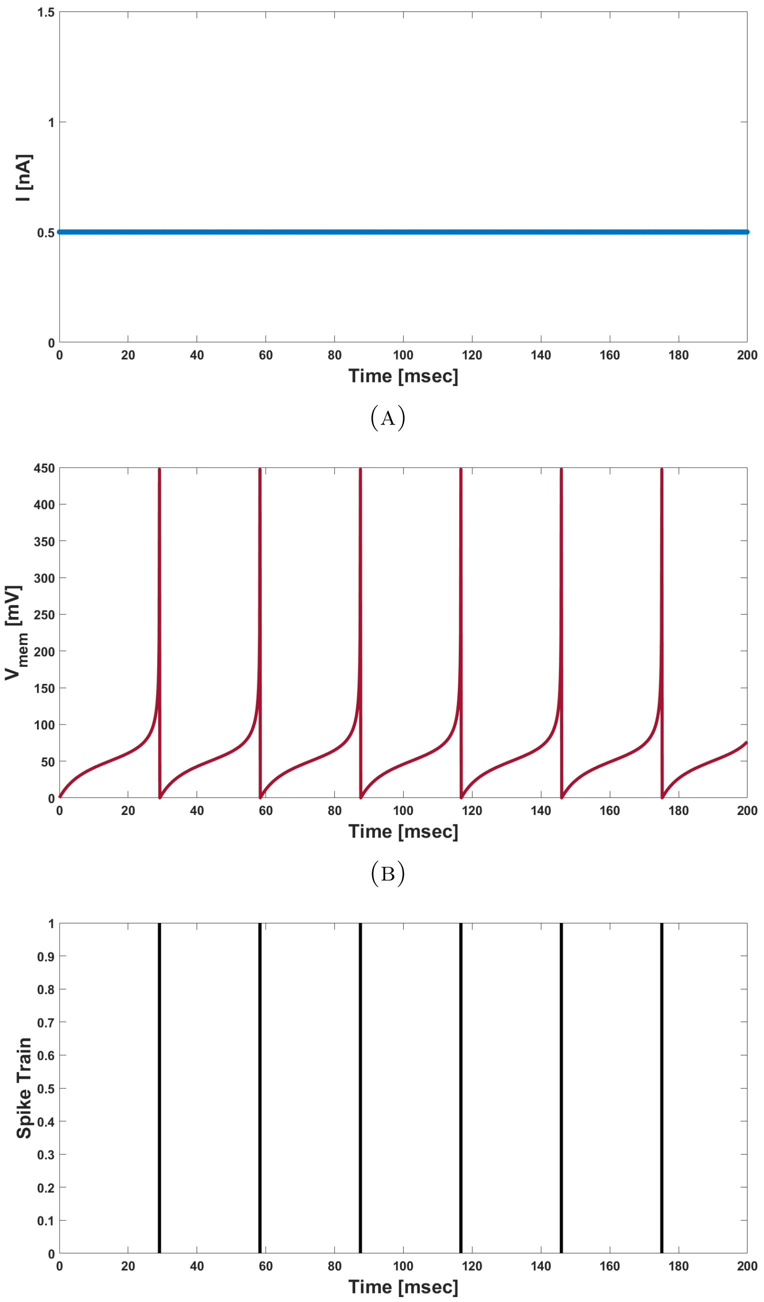
Fig. 4.9a shows a variable input current fed to the single neuron along with its response by means of its membrane potential (see **Fig. 4.9b**) and spike trains associated to it in **Fig. 4.9c**.

In **Fig. 4.10a**, a group of three neurons is considered (i.e. three are included); where different input currents are fed to **Fig. 4.10a** shows their responses by means of their membrane potentials. In **Fig. 4.10b**, a raster plot is displayed. **Fig. 4.10c** includes the transfer function of the model having the firing rate (as output) as function of the current (as input).

After the testing process of the model, the link between the hardware (i.e. in this case the output of the taxels) and Brain simulator is emphasized in **Fig. 4.11**. The current coming out from the taxel output is fed to the corresponding neuron utilizing one-to-one correspondence. As noticed from Fig. 4.11, th transfer function of the model used in Brian shows that the firing rate varies linearly with the input current. This proves the linearity between the input force and the firing rate of the neurons starting from the first stage and ending with the second one.

4.9 Discussion

Concerning **Fig. 4.8a**, the neuron starts charging as the input current is integrated into its membrane until the reset voltage is crossed and it sharply discharges. It continues with this behavior all over the duration of time (i.e. 200ms). Also, this behavior proves the modifications done to the model in such a way that the normal behavior of the LIF neuron is achieved and resembles that found in literature [29] without adaptation.



(A)

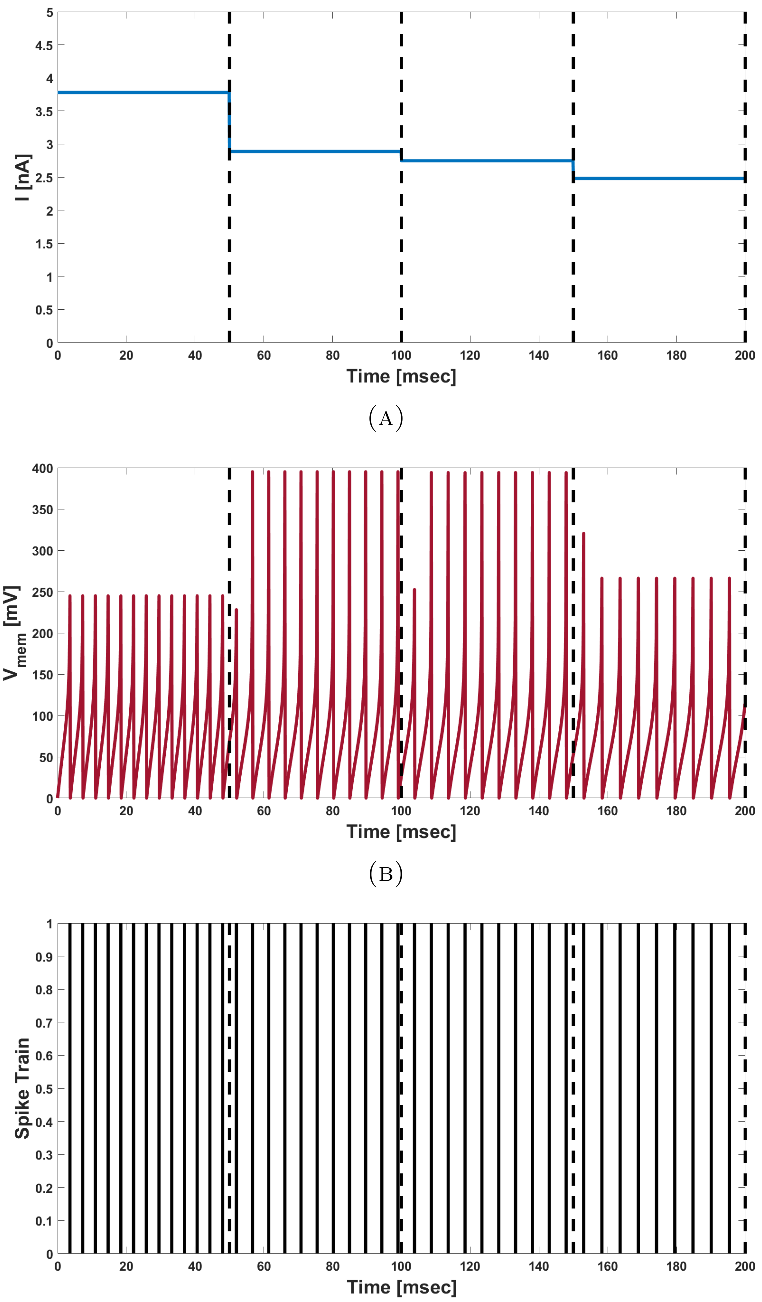
(B)

(C)

Figure 4.8: (A) Input current to the neuron versus time. (B) Membrane potential as a response of the integrated input current over time. (C) Spike train displaying the different instants of each spike.

In **Fig.4.9a**, the duration is divided into four intervals, 50ms per each. The response of the neuron shows that the more current is integrated, the more spikes are generated over the time interval considered.

To show the response of different neurons as a matter of the input current, **Fig. 4.10a** serves for this. Three neurons are considered and fed by three different



(A)

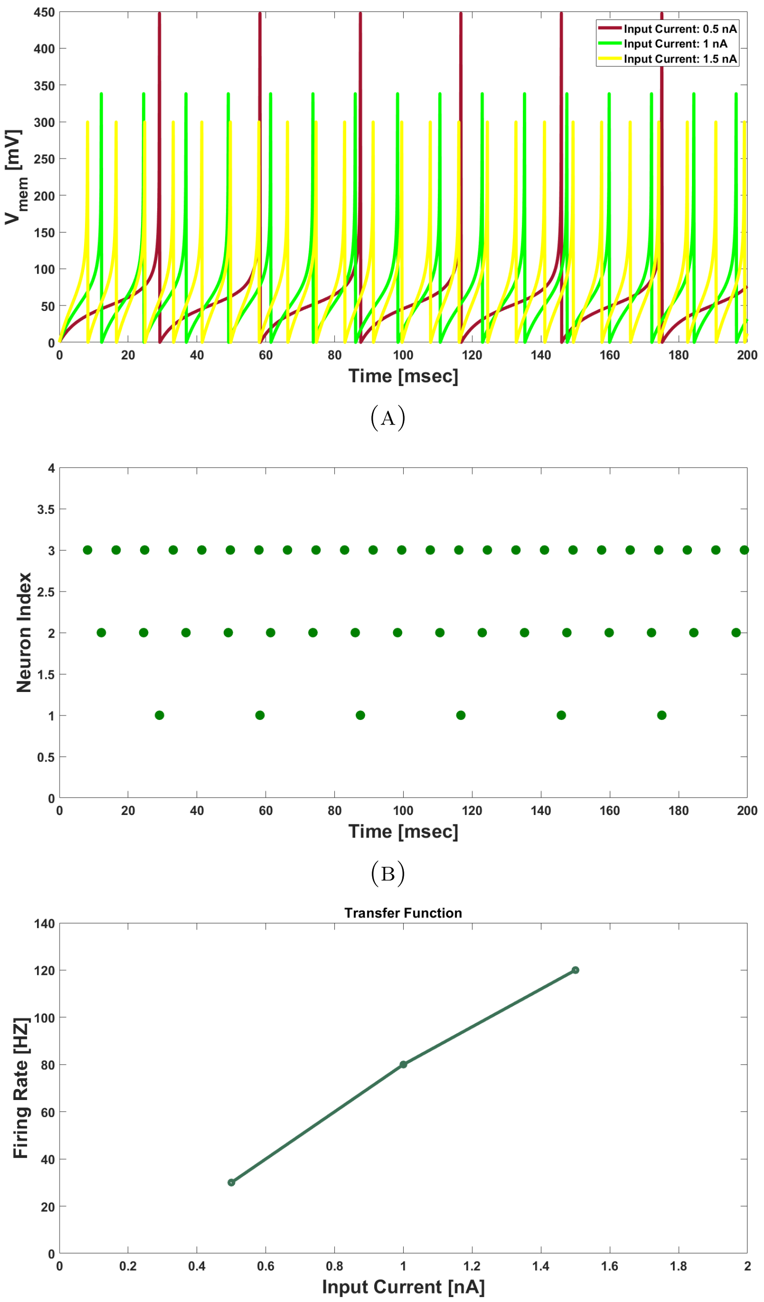
(B)

(C)

Figure 4.9: (A) Variable input current to the neuron versus time. (B) Membrane potential as a response of the integrated input current over time. (C) Spike train displaying the different instants of each spike.

input values (0.5 nA,1 nA,1.5 nA) are set. The response for this of neurons is compatible with that in **Fig. 4.9a**.

Fig. 4.10b shows the raster plot for the three neurons that highlights the instant a spike is generated and the index of its corresponding neuron. The importance of **Fig.4.10c** is to emphasize that the increase in the firing rate of the neuron



(A)

(B)

(C)

Figure 4.10: (A) Membrane potential of three neurons as a response of the input integrated current over time. (B) Raster plot. (C) The transfer function of the model.

comes out as a result of increasing the input current. That is why the firing rate increases proportionally with the increase in the input current.

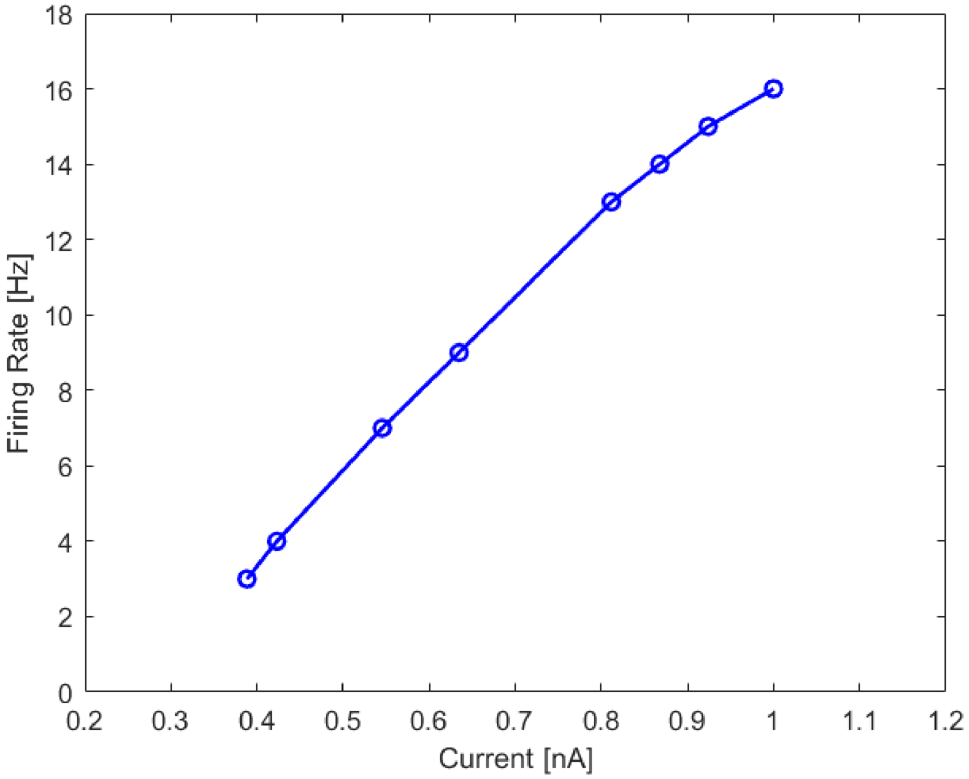


Figure 4.11: The transfer function of the model.

4.10 Properties of the Machine Learning Method

4.10.1 Supervised Learning

The algorithm gives each trajectory (i.e. the movement from one end to another with a certain angle) a specific label. Once the direction is figured out, a pop-up message appears showing the specifications of each trajectory concerning direction and orientation. Besides, the algorithm decides whether the current trajectory is previously learned or not by comparing with the learned data set collected during the learning phase where the algorithm learns L trajectories where each trajectory has a label identifying the direction and orientation of the object moving. Given the following details, the algorithm takes into account supervised learning in its classification method.

4.10.2 Based on Probabilistic Approach

Contrary to the deterministic approach that some algorithms are based on, my algorithm uses a probabilistic approach. It uses probability and random variables in its learning process (i.e. it is the first part) and in the online testing through the particle filter. The main principle in the probabilistic approach is to learn a probabilistic graphical model (**PGM**). A detailed explanation will be included in the offline part latter on.

4.10.3 Dynamic Concerning the Learning Capabilities

As the algorithm predicts at first to estimate the direction and orientation of the sliding tip, so the learning phase may not be sufficient to include all the possibilities of trajectories. That is why such an algorithm has the capability of making use of the incremental learning principle. So in case of the decision after the comparison of the new trajectory with the learned trajectory came to be not one of the learned trajectories, the algorithm will send a notification saying that such a new trajectory is unfamiliar to it and then takes an extra step in adding the latter to its learned data set trajectories. Therefore, the new unknown trajectory once tested again, becomes known to the algorithm and a Transition Matrix (**TM**) will be assigned to it. Moreover, it will be added to the list of trajectories upon which the classification process refers to for the detection purposes.

In this way, the learning process is extended to cover trained or untrained trajectories and would be able to generate for each trajectory a unique **TM** suitable for the online testing phase.

4.10.4 General concerning the Patch Geometry

One of the advantages of the considered algorithm is its independence of the geometry of the skin patch considered. In other words, the algorithm functions properly in case we introduce a triangular or square distribution of the taxels within a patch. The reason behind this comes from the way the algorithm is built where it considers the spatial position and the instant of activation of the taxels regardless of the shape of patch carrying the taxels is.

4.11 Detailed Machine Learning Technique

Given the output of the Brian simulator (i.e. the index and instance of the firing for each neuron along with its spatial location), the next step is to set the latter as input to the ML algorithm in MATLAB. In other words, the latter considers the output of the second stage as its own inputs.

Before going into the details of the algorithm, I would like to show the schematic of the skin patch (drawn and real ones) as shown in **Fig. 4.12**. The figure on the right shows the distribution of the taxels with a given label for each in the x-y plane. So as a remark, whenever it is mentioned that a neuron of index **i**, then this is considered to be taxel **i** shown to the left in **Fig. 4.12**.

The proposed algorithm is split into two parts: offline learning and online testing where both are discussed in the following two sections.

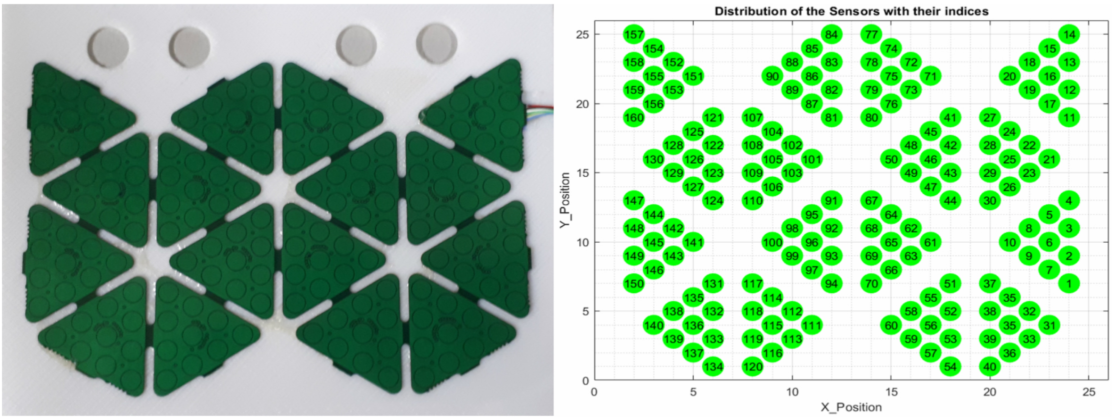


Figure 4.12: The real and drawn schematics of the taxels in the skin patch.

4.11.1 Off line Part: Learning Phase

In this part, the offline part is highlighted. This part is dedicated to the learning phase. The algorithm learns L trajectories where each trajectory has a defined label identifying the direction of the object moving and the angle it makes with the x-axis (discussed above in the supervised learning paragraph).

The inputs are the position of the neurons, the index of the activated ones and the instant of activation (see **Fig. 4.13**). The main idea in this part is to learn a Probabilistic Graphical Model **PGM** that can be represented by a **TM**. For each trajectory, there corresponds an input as above, the algorithm learns a PGM model

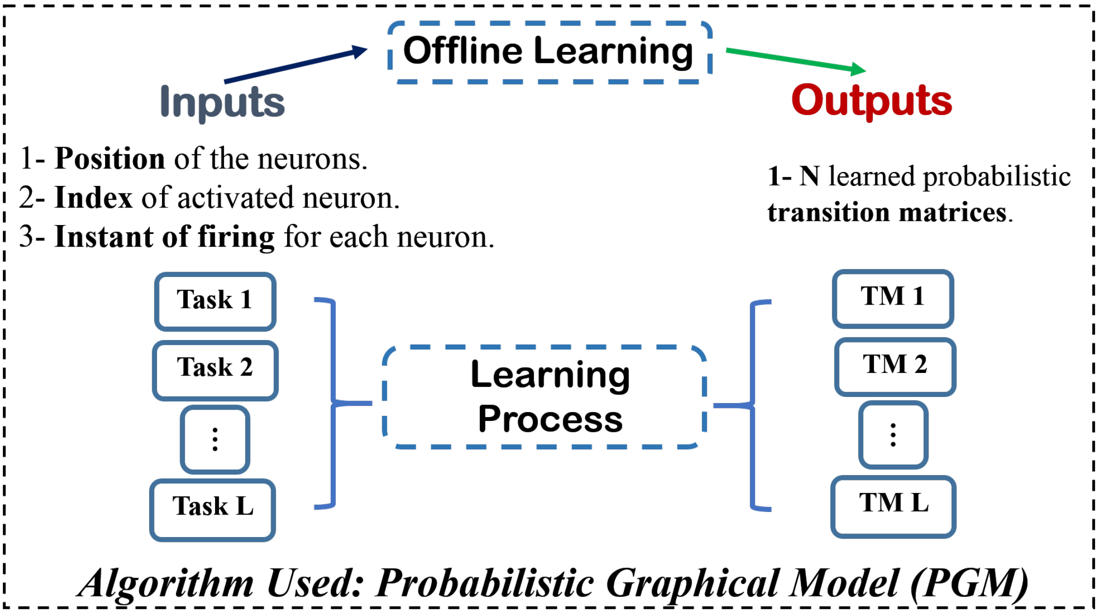

Offline Learning
Inputs
Outputs
1- **Position** of the neurons.
2- **Index** of activated neuron.
3- **Instant of firing** for each neuron.
1- **N** learned probabilistic **transition matrices**.
Task 1
Task 2
⋮
Task L
Learning Process
TM 1
TM 2
⋮
TM L
Algorithm Used: Probabilistic Graphical Model (PGM)

Figure 4.13: First basic part of the model.

as shown in the block diagram where **PGM** explains the relationship between activated neurons over time in a probabilistic way (refer to **Fig. 4.14**).

In such a part, after each trajectory, the algorithm generates a **TM** related to each trajectory. The latter is a **n by n** matrix where n represents the number of taxels under consideration in the skin patch. The entities forming the latter show the conditional probability and the relationship between activated neurons over time. In other words, the elements of the matrix give the dependency of the transition between taxels in terms of conditional probability. The latter is calculated from the response of the neurons in Brian. The algorithm receives a list of activated neurons over time. So, in cases where the taxel is pressed more than its neighbors, the corresponding neuron generates more spikes over a considered time interval. Thus, the same neuron is repeated more than once in the spike train. Therefore, the corresponding taxel will be repeated more in the list. Given this, for each trajectory, each taxel within the **n by n** matrix, will have a specific probability showing the possibility of staying at it or moving to a neighbor one.

That is why for each value included in this matrix, it shows the probability of staying at its current taxel denoted by **i** or moving to one of the neighborhood taxels denoted by **i+1** (refer to **Fig. 4.15** where S represents the state or taxel, 160 states or taxels in our study).

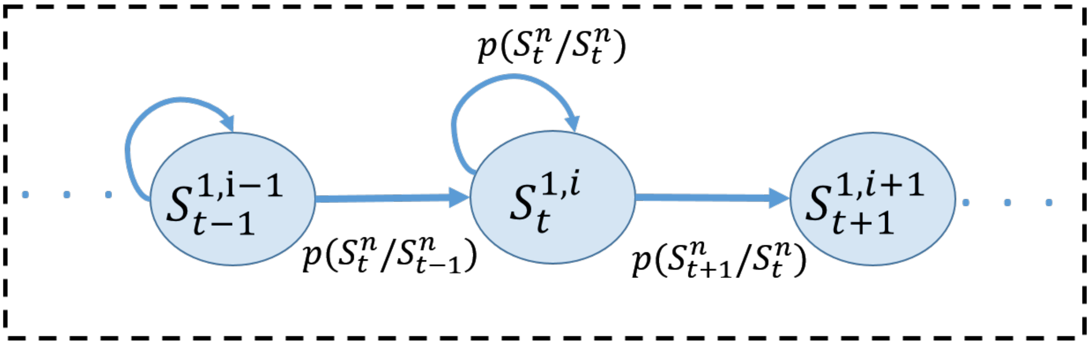
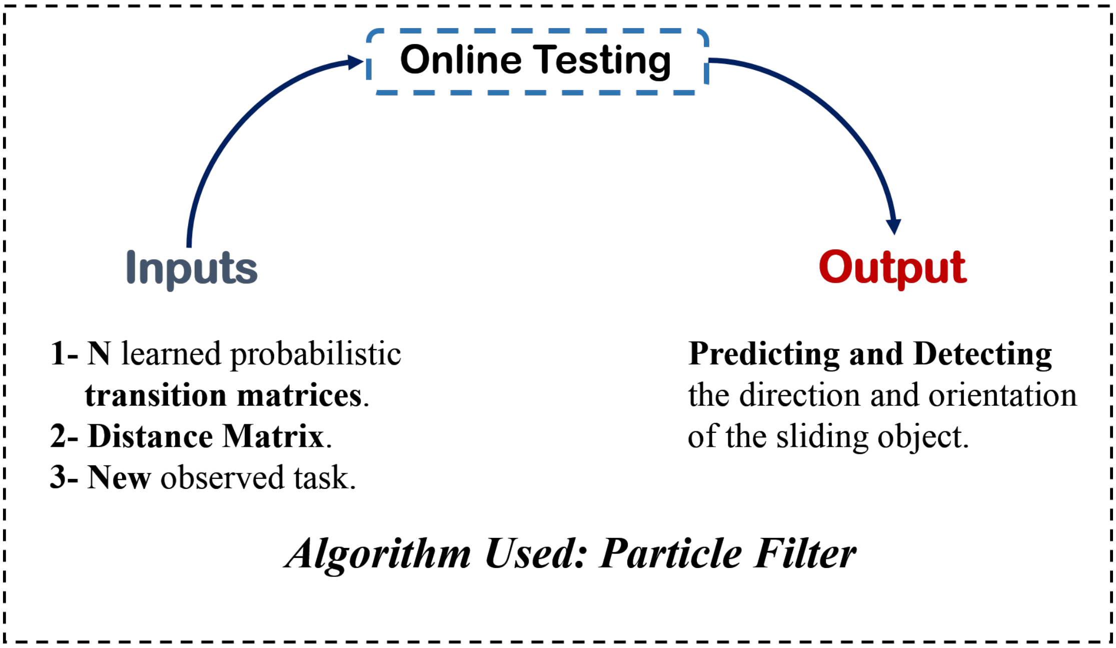


Figure 4.14: Probabilistic Graphical Model (PGM).

$$\begin{array}{c}
 \\ S^1 \\ S^2 \\ \cdot \\ \cdot \\ \cdot \\ S^D
\end{array}
\begin{array}{c}
\begin{array}{cccc} S^1 & S^2 & \dots & S^D \end{array} \\
\left(\begin{array}{c}
p(S^1/S^1)\, p(S^2/S^1) \dots\dots p(S^D/S^1) \\
\cdots\cdots\cdots\cdots\cdots\cdots \\
\cdots\cdots\cdots\cdots\cdots\cdots \\
\cdots\cdots\cdots\cdots\cdots\cdots \\
\cdots\cdots\cdots\cdots\cdots\cdots \\
\cdots\cdots\cdots\cdots\cdots\cdots
\end{array}\right)
\end{array}$$

Figure 4.15: TM associated with each trajectory.

Here is a more detailed explanation of how the algorithm transforms the movement of the sliding tip in each trajectory into a **TM**. Imagine we have a sequence of activated neurons over time. So this will give information about the neuron number and time of its activation. Generating the **TM** originates from these two given points. For example, assume that we received a list of neurons activated where neuron **i** is repeated 7 times in the data file. So, the algorithm will convert the number of counts of neuron **i** in addition to other neurons considered into a probabilistic matrix by doing the normalization over all the neurons. By this, the **TM** gives us information about how the activation of neurons evolves over time. Moreover, it gives us the dependency of the transition between variables in terms of a conditional probability for each trajectory in the learning process.



Online Testing

Inputs

1- **N** learned probabilistic **transition matrices**.
2- **Distance Matrix**.
3- **New** observed task.

Output

Predicting and Detecting the direction and orientation of the sliding object.

Algorithm Used: Particle Filter

FIGURE 4.16: The inputs and output of the on line testing.

4.11.2 On Line Testing with State Estimation

After learning a set of trajectories, now it is time to predict to detect the orientation of a new trajectory (i.e. a new trajectory of the sliding tip in a certain direction on the skin patch).

The output of the offline learning is set as input to the online testing. In addition to that, a distance matrix (**DM**) is calculated based on the spatial separation between each taxel and all the others. After that, the entities of the latter are normalized to perform as probabilities. So, the new values are considered to be probabilities as a function of the distance separating the taxels. This implies that the closer taxels will have a higher probability to go to while the far away will have almost zero probability for the fact that the tip is pressing on one of the neighborhood taxels instead of jumping to far away ones. The importance of the **DM** is for two main reasons: its role in the estimation of the error in the online testing part and in giving accurate differences in the case of very close trajectories (for example nearly activated taxels in two different trajectories).

After learning a set of trajectories, which are represented by **TM** and **DM**, the new observation is set as a new trajectory. So, in this way, I have two sides: the learned trajectories that I can consider as my reference and the new trajectory that I need to classify it according to the learned ones. This step needs an agent that is capable of differentiating both and decides whether the new trajectory is something learned by the algorithm or it is new. That is why I chose the particle filter for this purpose.

4.11.2.1 Particle Filter

While the **PGM** algorithm is clarified and exploited in the offline learning, the **PF** algorithm is adopted. The role of the **PF** is to predict the future state (i.e. the next taxel **i** after being at taxel **i-1**). Therefore, I apply a particle filter to predict the future state and then to calculate the corresponding error to decide finally which trajectory the new one is (either one of the learned or new).

Particle filtering uses a set of particles (also called samples) to represent the posterior distribution of some stochastic process given noisy and/or partial observations. The state-space model can be nonlinear and the initial state and noise distributions can take any form required. Particle filter techniques provide a well-established methodology [49][50] for generating samples from the required distribution without requiring assumptions about the state-space model or the state distributions. However, these methods do not perform well when applied to very high-dimensional systems.

Particle filters update their prediction in an approximate (statistical) manner. The samples from the distribution are represented by a set of particles; each particle has a likelihood weight assigned to it that represents the probability of that particle being sampled from the probability density function. Weight disparity leading to weight collapse is a common issue encountered in these filtering algorithms; however it can be mitigated by including a re-sampling step before the weights become too uneven. Several adaptive re sampling criteria can be used, including the variance of the weights and the relative entropy with respect to the uniform distribution [51]. In the re-sampling step, the particles with negligible weights are replaced by new particles in the proximity of the particles with higher weights.

The objective of a particle filter is to estimate the posterior density of the state variables given the observation variables.

Once there is a new trajectory, then we talk about new states or observations (i.e. Z_t^i where Z refers to a current observation, i to the taxel index and t to a certain instant). The **PF** role is to predict the new state of the sliding tip based on the **DM** from a probabilistic approach (see **Fig. 4.17**).

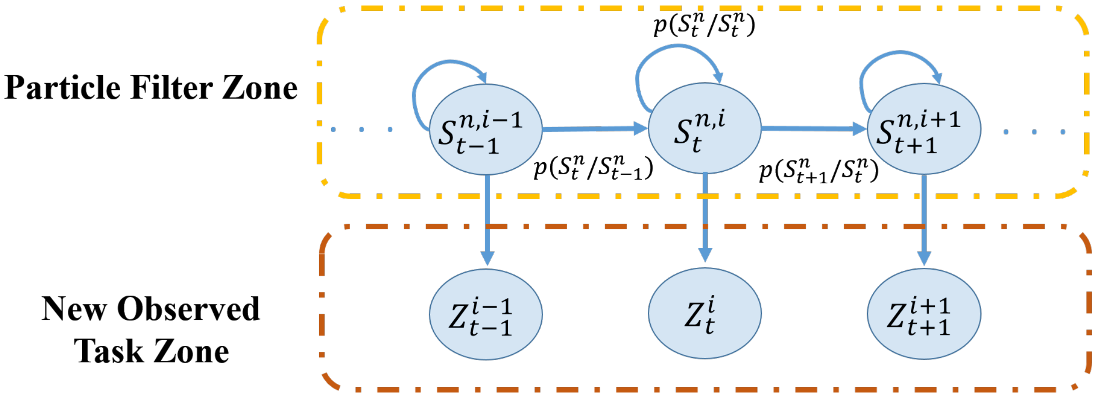


Figure 4.17: Particle filter functionality.

The launching state for the particle filter is based on the first state of the new trajectory. So given that the real state **Z** is at taxel i, then the PF chooses the launching state **S** by referring to the **DM**. It draws a **PDF** and chooses a random variable. With such a random number within the PDF, the intersection corresponds to a given state **S**. After that, given the current observation, the particle filter estimates the future state. This estimation is based on the TM of the learned trajectories. It does the same as the first launching state by drawing a PDf and choosing a random number. However, instead of referring to the DM, in such a case, it refers to the TM corresponding to each state. In this way, the PF calculates the error between the current and the predicted based on the distance separating by referring to the DM previously determined in the offline learning. Based on the local error between every two states (real and predicted), an average is calculated and compared to the threshold previously set as shown in **Fig. 4.18**.

For example, if the new trajectory contains 10 activated taxels, the PF predicts 9 future states. As a consequence, the former calculates 9 local errors for each predicted state. By getting the average, a global error is achieved. Comparing this global error with a threshold, the PF will be able to find if the new trajectory under test matches or not a trajectory from the learned trajectories. A pop-up message will be displayed to convey this. As a result of this, we can deduce whether the new trajectory is strictly comparable to the learned models or a purely a new and unobserved trajectory.

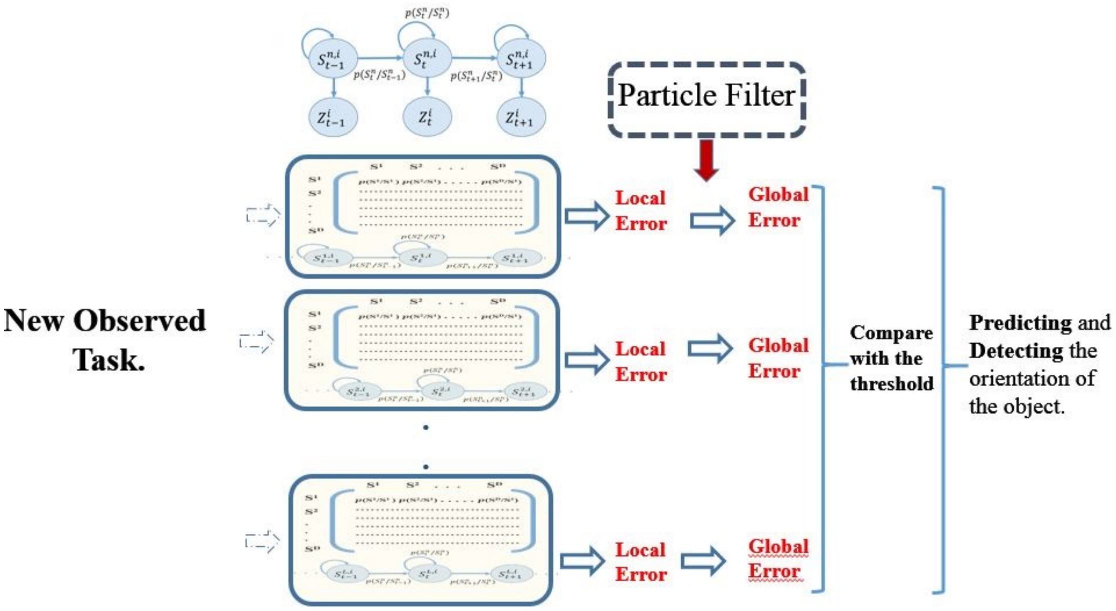

Particle Filter
New Observed Task.
Local Error
Global Error
Compare with the threshold
Predicting and Detecting the orientation of the object.

FIGURE 4.18: Particle filter functioning zone with the observed trajectory.

4.12 Third Stage Results

4.12.1 Offline Part

In this section, the first step that I did is to generate a schematic view for the taxels in the real patch using MATLAB. As shown in **Fig. 4.19**, each taxel is given a specific number along with a spatial position along the x and y-axis. This resembles exactly the real skin path.

Done with the distribution of the taxels scheme, now it is time to highlight the off line learning by giving different outcomes of the algorithm used. I consider in the coming part a limited number of training and testing trajectories for the aim of clarifying the work done and then extending it later ons.

Fig. 4.20 shows two of the movements done by the sliding tip along with their corresponding **TM**s. The training data are in red contour circles, trajectory 1 in **Fig. 4.20a** and trajectory 2 in **Fig. 4.20c** respectively, and in blue the direction of the tip over time. As shown in both figures, the tip while scanning the patch has almost a linear path from one to the other. Besides, it is clear the activation of more than one taxel at a time. This due to the position of the tip while scanning the patch in case it crosses between two taxels and touching them at

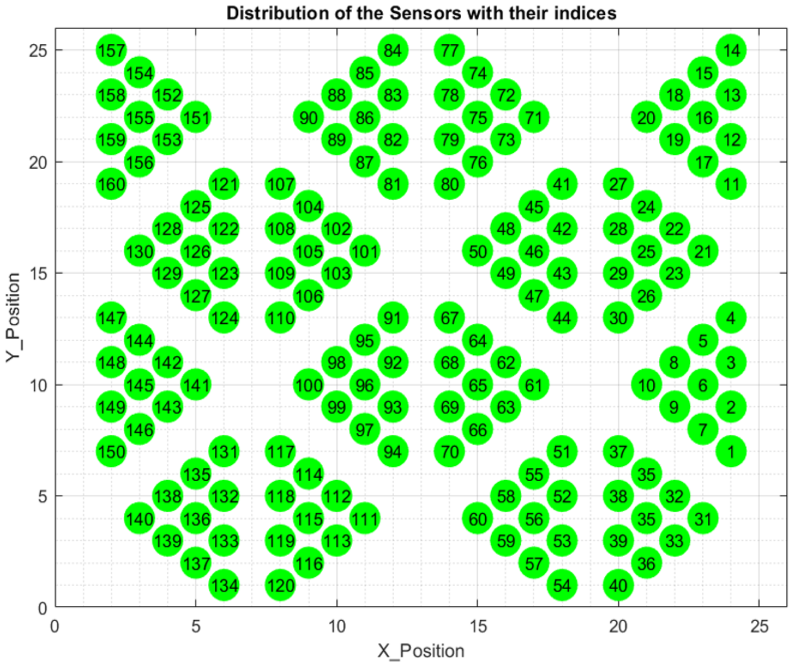


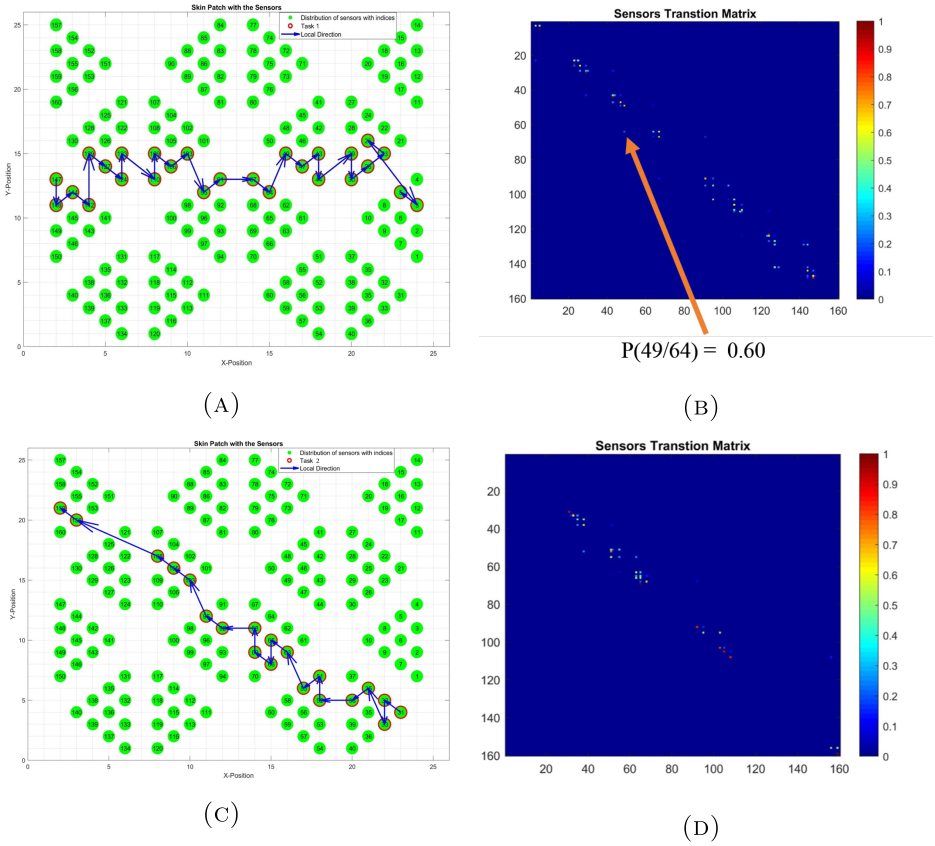
FIGURE 4.19: Schematic view for the taxels distribution in the skin patch.

once. In addition, the fabric, being pressed by the tip, bents a little toward the neighborhood taxels making them activate even if not touched completely.

Utilizing the **PGM** method through its algorithm, the latter represents each trajectory with its corresponding **TM** through a color map. **Figs. 4.20b** and **4.20d** show the **TM**s for the movements of the tip. The x and y axis are of 160 point that count for the number of taxels under consideration.

As shown in both figures, the **TM** shows the conditional probability of a transition between one taxel and another (i.e. 160 taxels in this case). The horizontal axis represents the taxels numbers of the next taxel after being in a previous taxel shown in the vertical axis. It also shows an example concerning the situation in being at taxel number 64 and moving to taxel number 49. In such a trajectory, with the probabilistic approach adopted in such an algorithm, the probability computed is 0.6.

Furthermore, looking at the two trajectories and their corresponding **TM**s, we notice that these two trajectories are different which leads the **PGM** to generate two different **TM**. This in return proves the idea behind the classification way the algorithm does that is the supervised learning way.



(A)

(B)

(C)

(D)

Figure 4.20: (A) The training data for the movement of the sliding tip from left to right and activating specific taxels over time.(B) The **TM** resulting from the movement of the tip in left to right direction.(C) The training data for the movement of the tip along the diagonal.(D) The **TM** resulting from the movement of the tip along the diagonal.

Fig. 4.21 extends from **Fig. 4.20a**. The former shows the linear fit to the trajectory of the tip and the angle separating the linear fitted line with the horizontal axis.

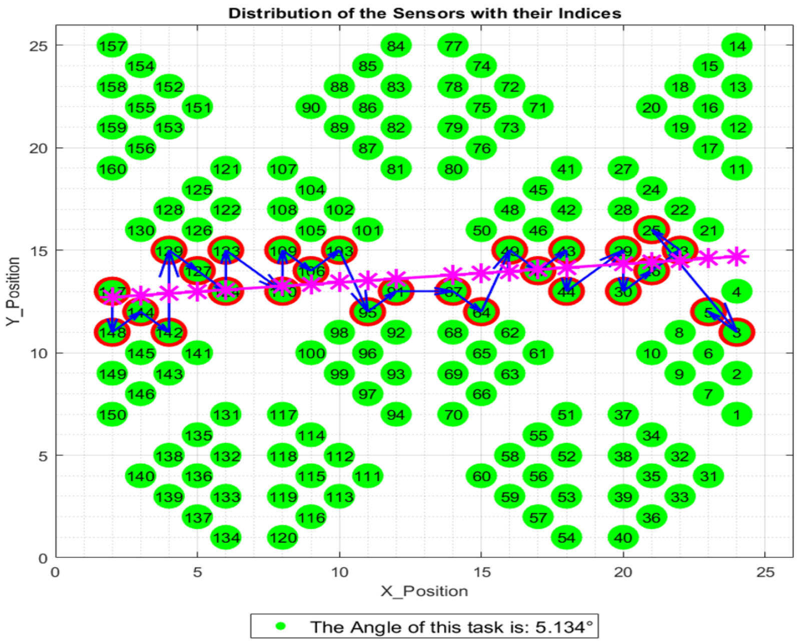


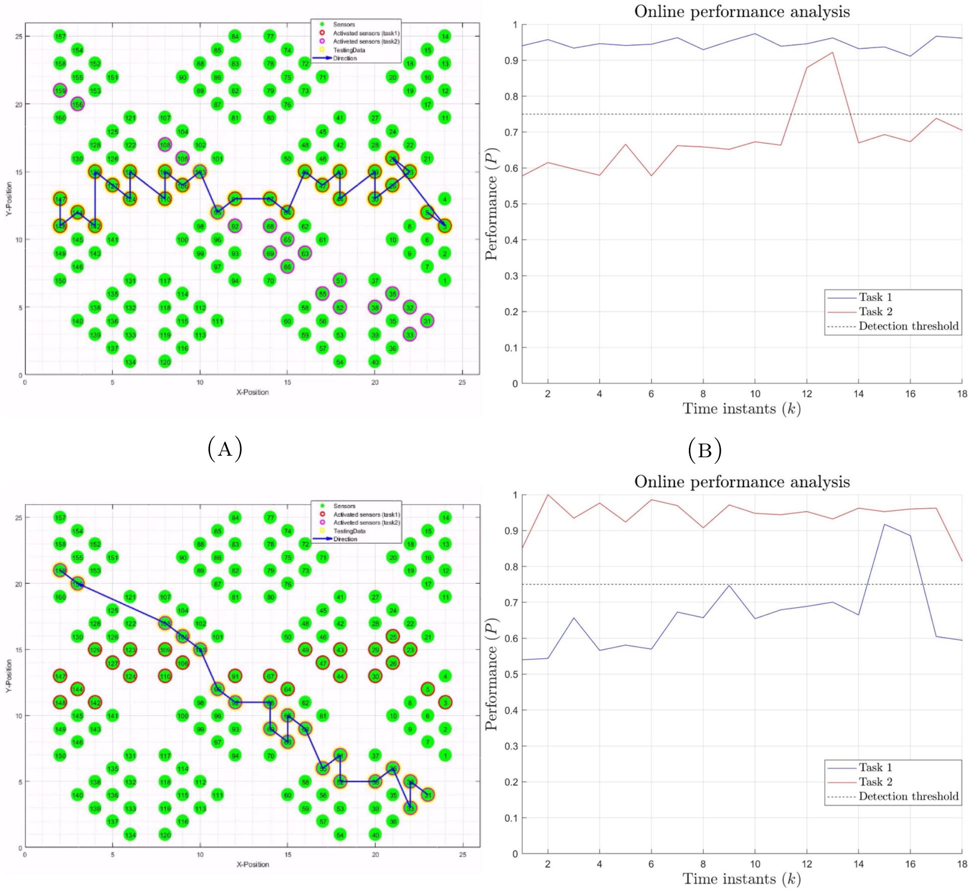
FIGURE 4.21: The linear fitting to the path followed by the tip during its movement from left to right.

4.12.2 Online Part

Done with the learning part using the offline algorithm, now I will shift to the online testing part. I include in this part the testing data and perform an online testing on them.

Figs. 4.22 and **4.23** show the online analysis of 4 different trajectories performed and the resulting performance. **Fig. 4.22a** shows the activated sensors of trajectories 1 and 2 (previously trained in the offline learning phase) colored in red and magenta respectively. The yellow contoured taxels represent the testing data, and the blue arrow shows the activation of the taxels over time continuously. In such figure, the testing data corresponds to the movement of the tip horizontally from left to right.

Fig. 4.22b comes as an extension of **Fig. 4.22a** for the trajectories prediction and detection of the direction and orientation. It shows the performance of the testing data relative to the trained data versus time. As shown, it is compatible with **Fig.4.22a** where there are two trajectories (1 and 2). Just for a quick note, the performance is considered to be opposite to the error (refer to **Fig .4.18** for more information). So, once the local error is high, then the performance is low



(A)

(B)

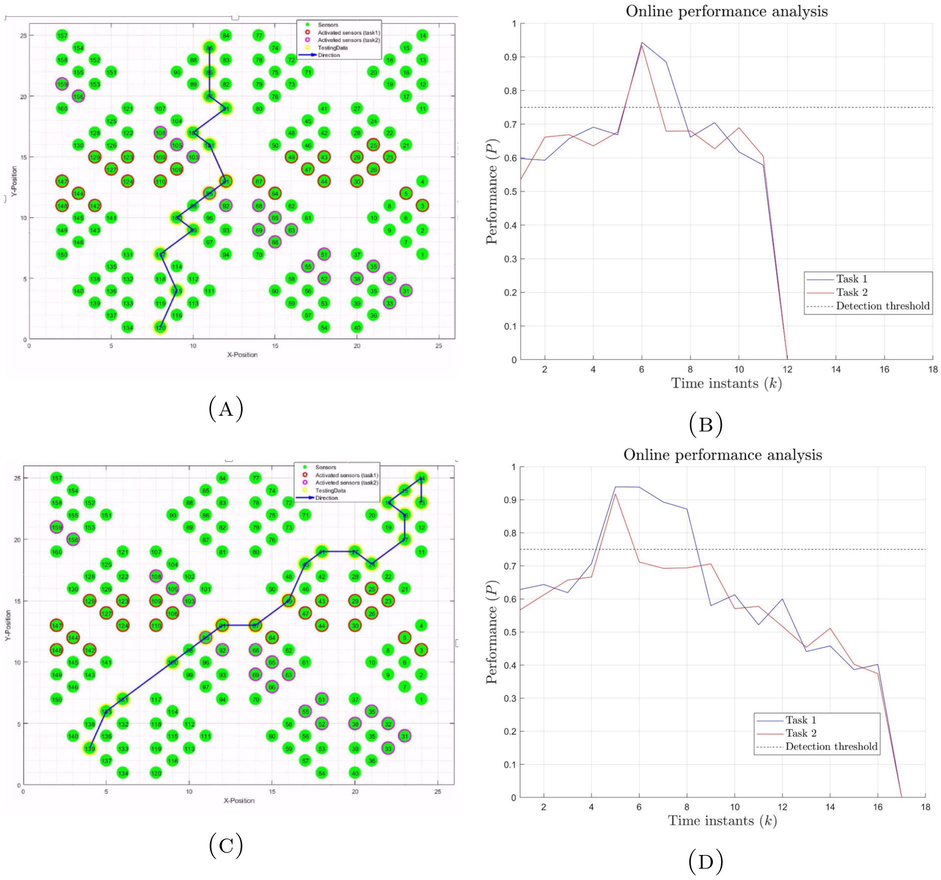
(C)

(D)

FIGURE 4.22: (A) The testing data of the sliding tip from left to right.(B) The online performance analysis of the left to right movement.(C) The testing data along the diagonal.(D) The online performance analysis of the down left to top right movement along the diagonal.

and vice-versa. The latter gives an idea about how much the tested trajectory matches the trained trajectories or not.

The blue and red lines show the performance of trajectories 1 and 2 respectively. Concerning the blue line, the performance exceeds the threshold while that for trajectory 2 is below it. Since the averaging over the local error (26 local error computed by the **PF** in this case because 27 taxels are activated during trajectory 1). The error concerning trajectory 1 is low while trajectory 2 is high. Thus, the global error is greater than the threshold and almost 97% comparable to trajectory 1, then the pop-up message is ”trajectory 1” and not ”trajectory 2”. There is a special jump in the performance of trajectory 2. In this region, called confusion region, the taxels activated are common in both the testing and learning data. Thus, the error is small and the performance is high. That is why the red line



(A)

(B)

(C)

(D)

FIGURE 4.23: (A) Movement of the sliding tip from left to right and activating specific taxels over time.(B) The color map shows the transition between taxels for the path of the tip from left to right in terms of probability.(C)The movement of the tip along the diagonal.(D) The **TM** resulting from the movement of the tip along the diagonal.

crosses the threshold. The same applies to figures **Figs .4.22c** and **4.22d**. The new in **Fig .4.22c** is the new trajectory being across the diagonal.

Fig. 4.23 differ from the previous one by including new trajectories that were not included in the training phase. The same is done by showing the activated taxels and the and the online performance. In such case, the error is high and thus the performance is low except for the common region where common taxesl are shared between the training and testing trajectories. Consequently, the error is high leading the algorithm to display a message that the new trajectory is not one of the trained ones.

4.13 Statistical Validation of the Learning Method

To statistically validate the proposed **ML** method, I considered a data set composed of 78 trajectories. Splitting the former into training and testing data, I considered 56 to be training data and 22 as testing data. I did a set of trials by which I did a training on the training data (call it first case) and testing on the testing data (call it the second case). I computed the error and how it changes during the first and the second cases.

4.13.1 Testing on the Training Data

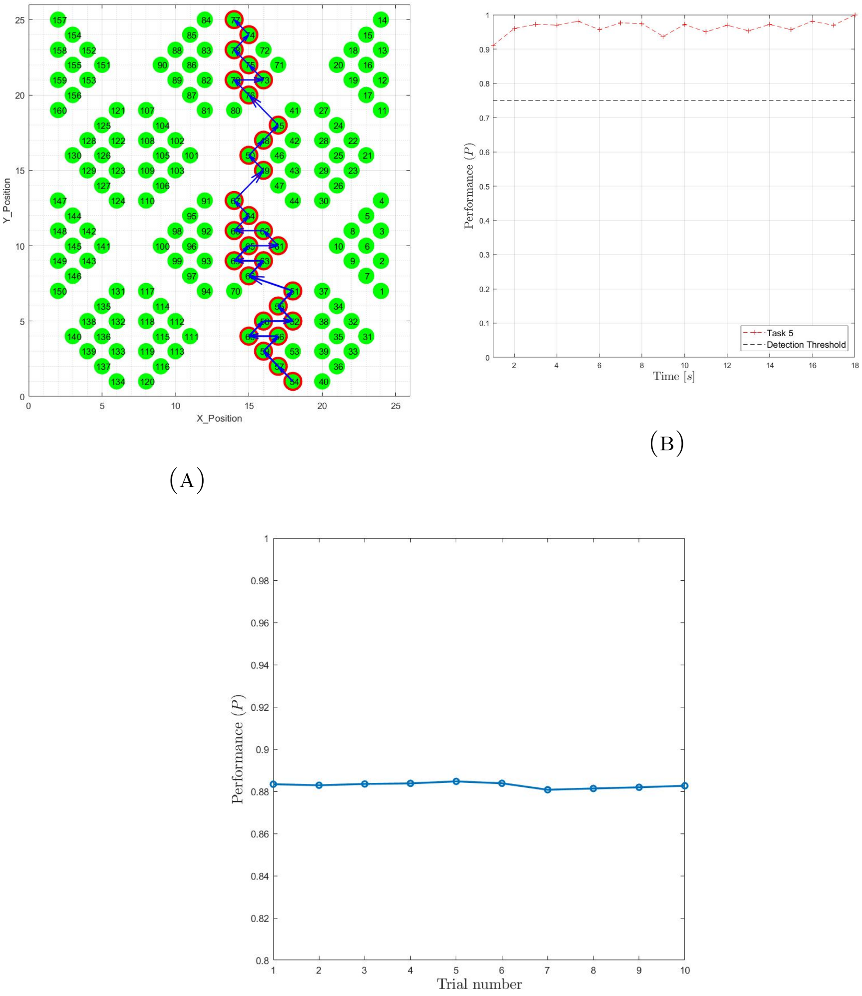
Concerning the training on the training data, I considered to do the same procedure for 10 times. **Fig.4.24a** shows the path followed by the tip from bottom to top. **Fig. 4.24b** presents the performance of the latter in accordance to the **TM** combining the **TM**s of the 55 training trajectory for one execution. **Fig. 4.24c** presents the change in the performance after doing the online testing for 10 times. AS noticed, the performance is almost the same over the 10 trials (average = 0.908789877 out of **1**), and thus the algorithm is quite accurate considering the training data as testing data.

Fig.4.25 shows the average and the standard deviation for each of the 56 training data considered as testing data. The average performance across all data is drawn, and it is calculated to be 0.876247575 out of 1.

4.13.2 Testing on the Testing Data

Done with the testing on training, now I shift to test using new data. **Fig.4.26** shows the average of the performance along with the standard deviation for 22 testing data corresponding to 22 new trajectory. The average for all the trajectories is also plotted and shown as dashed line in the plot. The good performance of the new trajectories reflects the idea of the confusion presented between the new trajectory and the trained trajectories.

In **Fig.4.28**, I set the omega device to scan the patch along the same path (from left to right) with different pressing forces and sliding speeds. The aim behind this



(A)

(B)

(C)

Figure 4.24: (A) Movement of the sliding tip from bottom to top.(B) The performance of the trajectory in Fig.(A). (C) The performance over 10 trials of the trajectory in (A).

method is to test the response of the algorithm towards the same trajectory but with different force and speed for the tip.

I collected the responses of the taxels and tested them following the work flow until the machine learning algorithm. What I realized is that as a matter of increase in the force set value, the tip is able to press the taxels more and hence more spikes are generated by the corresponding neurons in Brian. In addition to that, the number of activated taxels is proportional to the applied force. That is why their number is affected by the force value set for the tip to scan the patch with. The data tested concerns the path from left to right. The largest force value (i.e.

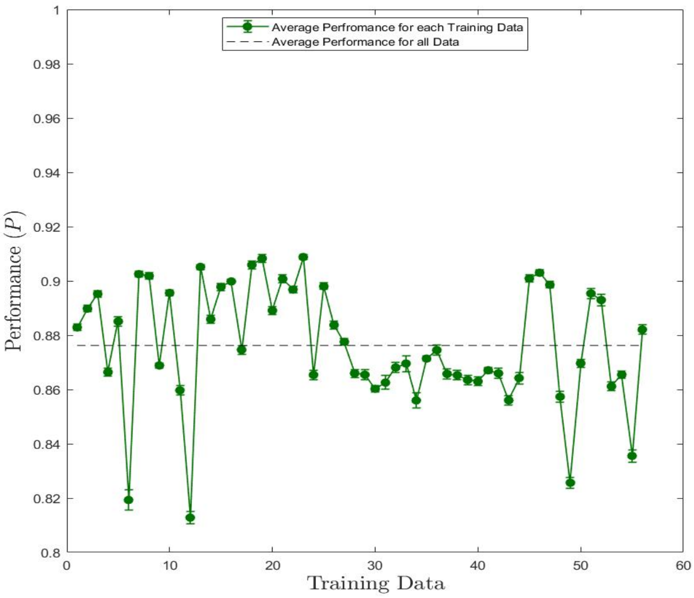


FIGURE 4.25: The average performance over the 10 trials.

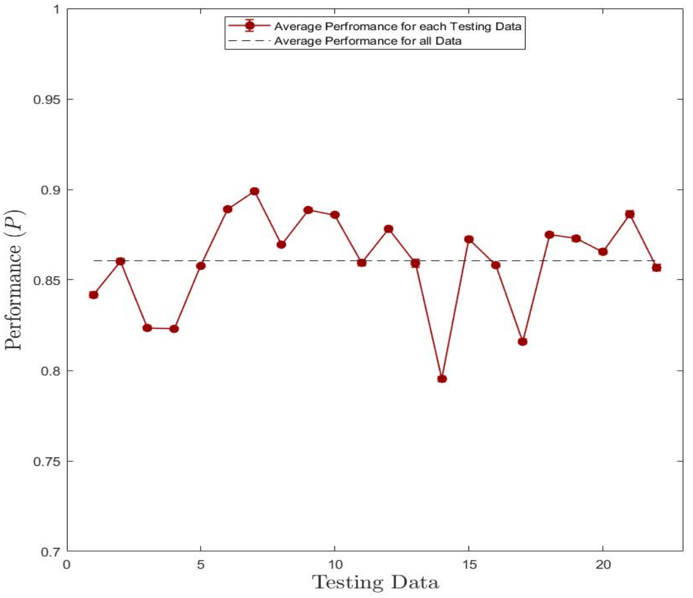
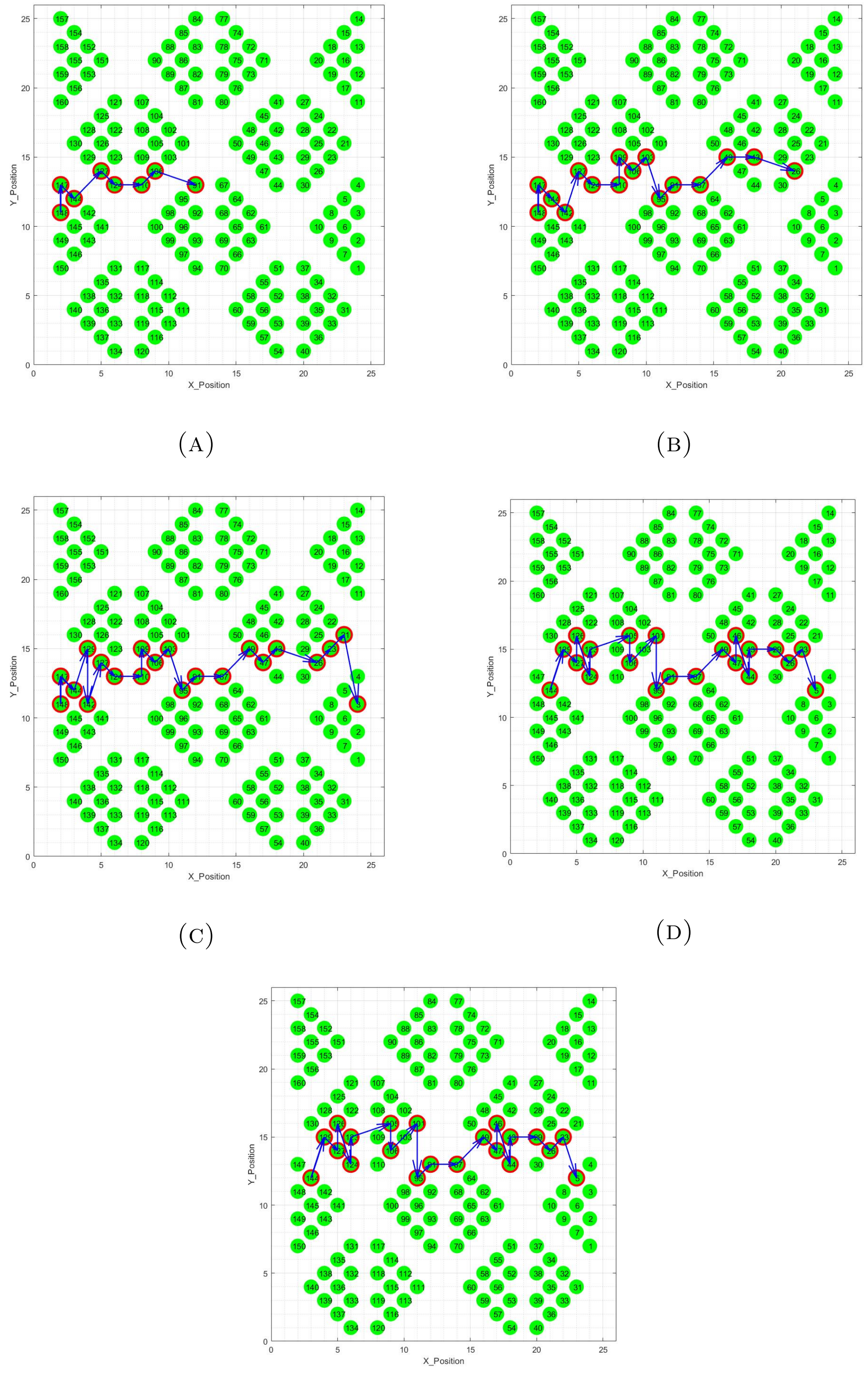


FIGURE 4.26: The average performance over the 10 trials for testing data (i.e. new trajectories).

2.1 N) activated 21 different taxels whereas smallest force value (i.e. 0.612 N) activated 8 taxels (refer to **Fig. 4.27**).



(A)

(B)

(C)

(D)

(E)

Figure 4.27: Movement of the sliding tip from left to right with: (A) F = 0.612 N and V =6.21 cm/s activating specific taxels over time.(B) F= 0.8 N and V = 6.98 cm/s. (C) F= 1.01 N and V = 14.1 cm/s. (D) F= 2.1 N and V = 3.1 cm/s.

The average performance for each new trajectory with different force and speed is shown in **Fig. 4.28**. It shows that the highest force has the highest performance.

The reason is that the algorithm acquired more knowledge about such trajectory due to the large number of activated taxels making the error in the detection lower and thus high performance. That is why we notice the compatibility between **Figs. 4.27e** and **4.28**.

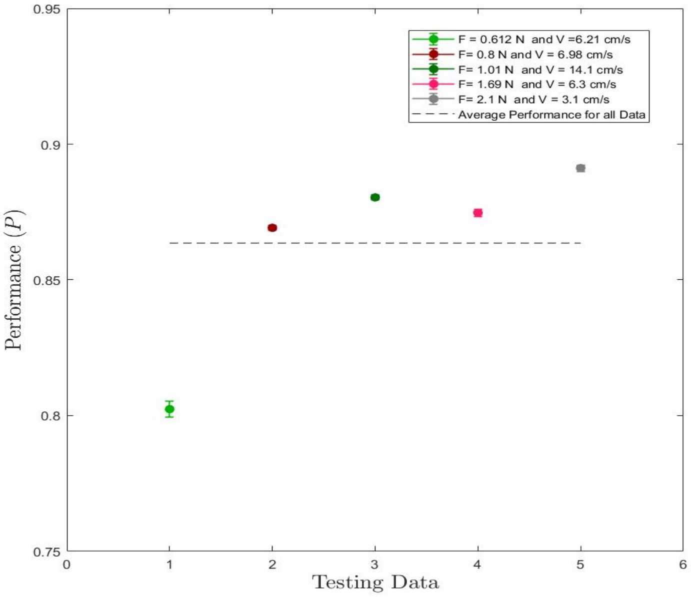


FIGURE 4.28: The average performance over the 10 trials for testing data (i.e. new trajectories).

Chapter 5

Conclusion and Future Perspective

Tactile sensing, like vision, is an important and integral part of the perceptual mechanisms of humans and, for that matter, of artificial devices such as robots. For this reason, the latter tactile sensors are acquiring a valuable role in the contemporary growing technologies and applications; contributing effectively and specifically to a large extent in robotic and prosthesis domains for contact and force feedback. In general, their suitability and development to be implemented on the body of the robot require a lot of studies, techniques, and methods in the fields such as material science, electronics, transducers, and communications based on the aimed application. We are concerned about the study of these sensors, their designs and applications to mimic the sense of touch in humans, specifically and efficiently on the circuit level.

This thesis is divided into two parts: circuit characterization utilizing simulation and experiments and detection of the direction and orientation of a sliding object on iCub skin patch using machine learning technique.

Concerning the first part, we merged in our circuit designs between CMOS technology and neuromorphic circuits being inspired by the biological system inside the human body. We presented the design and the functionality of an event-driven touch CMOS sensor circuit for artificial devices. The latter is designed in CMOS AMS 0.18 μ m technology with modifications across its constituents compared to a previously fabricated and tested prototype. As real-world tasks involve both static and dynamic contact events, the extensive characterization, clearly stated and discussed of the new sensor circuit, confirmed the goal behind its design and highlighted its response to static and dynamic input potential difference and pressure. The pressure on the POSFET is converted to a proportional current fed to two separate neuromorphic neuron circuits that generate spikes as signs of an event (either pressure or release), driving the whole sensor circuit. As it encodes for the absolute value of the applied force, this encoding of pressure levels loosely recalls the sustained type of skin mechanoreceptors (Merkel cells) inside the human body. The included results showed the better sensitivity to smaller input potential difference and force amplitudes compared with the previous circuits designed, area saving and smaller bandwidth and built with newer technology, were the key features of the new sensor circuit design. Rich with these capabilities and improvements, the sensor circuit can be an integral part especially in robotics, prosthesis and bioengineering domains.

We also presented another event-driven tactile sensing element based on the change detector circuit of a vision sensor. The proposed architecture encodes the value of the input force variation over time across two separate outputs using the AER protocol. Working in the “transient” mode, the encoding of pressure levels recalls the skin Meissner Corpuscles based mechanoreceptors. Preliminary simulation results presented in this work confirm the suitability of the new proposed event-based tactile sensor. As a consequence of its ability to encode the variation of the input pressure or force through two separate output trains of spikes [37], it can be considered as an alternative for the event-driven ”transient” taxel proposed in [30]. The Dynamic Tactile Sensor (DTS) design opens a new approach towards a novel artificial tactile sensor that fits properly into robots and autonomous devices where power consumption and high temporal resolution are key design specifications.

In the second part, we worked on a method to detect the orientation of a sliding object on the iCub skin patch. Since working with one sensor is not sufficient to reach the desired goal, we used a kin path full of sensors to achieve this aim. I prepared an experimental setup suitable for scanning the skin patch by an object with different force and speed. After collecting the response of the sensors (i.e. taxels), I set them as input to Brian simulator. The latter is dedicated to neurons and neural networks. The neurons are fed by the output of the taxels output and on in return generate spike once their corresponding sensor on the patch gets activated. The outputs of the neuron model are set as input to the machine learning method. The latter is composed of two main parts: offline learning and online testing. The learning part considers Probabilistic Graphical Model **PGM** that assigns to each path followed by the tip a Transition matrix **TM**. So, for each path followed by the object sliding on the patch will have a specific TM that differentiates it from other paths. Based on that, each path is given a label that gives such an algorithm the principle of supervised learning. The online testing comes as the last step to achieve the desired goal. In this part, a new path is introduced and the algorithm with the help of the Particle Filter **PF** can predict first the orientation and then detects it. The algorithm finds a local error between the predicted state (i.e. a future position computed by the **PF**) and the real state (i.e. the number of the taxel activated). Averaging over all the local errors for different states yields a global error. The latter is compared to a threshold. Once it exceeds it, its performance becomes high and thus it can detect its orientation. I tested the proposed ML algorithm by setting different paths followed by the sliding object and computed the corresponding performances. It showed good results in two cases: testing on the training data and testing on testing data. Moreover, I examined the effect of different force and speed on the output of the algorithm. What I did is to set different input force magnitudes on the tip and its speed while scanning the patch for the same path. The results showed what is expected. In case the force is low in magnitude, a lower number of taxels gets activated and vice-versa. That is why in the testing part, the highest performance of the algorithm was with the highest force magnitude. It is because a higher number of taxels makes the path complete and with more probabilities among each taxel. Consequently, the global error becomes smaller and the performance becomes higher. Therefore, the algorithm becomes more accurate in detecting the direction and orientation of a sliding object.

5.1 Future Work

Concerning the first part related to the circuit design, all circuits before sending to fabrication have to be extensively characterized and tested by professional circuit designers with good experience. Some of the tests to be done such as Layout versus schematic, power and ground nets, corner analysis, parasitics extraction, temperature effect and more.

For the proportional POSFET circuit, the circuit worked well during simulation and experimental characterization even though the prefabrication tests are missing. However, the problem in the drift across the analog memory capacitor has to be solved. One of the possible solutions is to use a floating gate MOS capacitor instead of the current MOS capacitor. Floating Gate MOSFETs (FGMOS) are popular and widely used devices in every single PC, programmable voltage/current references, digital potentiometers, single-transistor DACs, adaptive/learning, and as nonvolatile memory elements in flash memories. Therefore, including this technique in designing the future circuit will solve the problem in the drifting of the saved voltage value and thus stays the same for the long term.

Regarding the differential POSFET circuit, the mentioned tests above should be done to make sure that the circuit functions similar to the simulation results after fabrication. I came up with this conclusion since when I tried to characterize the circuit experimentally, I was not able at all to have something comparable to the simulation results. An additional solution is to try a better dimensioning of the MOSFET transistor and more accurate tuning of the biasing voltage accordingly.

Concerning the edge orientation part, a list of ideas can be added to improve it. One of the things that can be done is to directly connect the output of the taxels to MATLAB script. This step will save time and computation and the idea of online testing will in real-time. For example, achieving such a step will enable the user to slide an object on the patch and the latter will be connected to a PC. The PC will post-process the data collected and visualizes the path followed on its screen at the same time the object is sliding. Meanwhile, the algorithm will be computing the performance of such a task and figuring out its direction and orientation.

Another interesting point is to use a rigid transformation in the algorithm. With the help of the latter, the computation will faster in detecting the orientation and direction of the sliding object. This happens because all the tasks from left to right to be considered having a unique label, and the same applies to other tasks in different directions.

Third interesting point is to use such an algorithm in real-time to resemble Google translator. Using the direct path from taxels to the PC with the algorithm and the rigid transformation principle, the user presses with his hand on the taxels and the algorithm after being trained figures out the letter or number as a result of the path implemented by the user's hand.

Bibliography

[1] RS Dahiya. Touch sensor for active exploration and visuo-haptic integration. *Genova: Phd Thesis Dissertation*, 2008.

[2] Ali Abou Khalil, Maurizio Valle, Hussein Chible, and Chiara Bartolozzi. Improved event-driven touch cmos sensor. In *Ph. D. Research in Microelectronics and Electronics (PRIME), 2017 13th Conference on*, pages 169–172. IEEE, 2017.

[3] Frederic H Martini, Judi L Nath, and Edwin F Bartholomew. Fundamentals of anatomy and physiology. 2001. *Pentice Hall: New Jersey*, pages 538–557, 2015.

[4] Eric R Kandel, James H Schwartz, Thomas M Jessell, Department of Biochemistry, Molecular Biophysics Thomas Jessell, Steven Siegelbaum, and AJ Hudspeth. *Principles of neural science*, volume 4. McGraw-hill New York, 2000.

[5] Miao Li, Kaiyu Hang, Danica Kragic, and Aude Billard. Dexterous grasping under shape uncertainty. *Robotics and Autonomous Systems*, 75:352–364, 2016.

[6] Chiara Bartolozzi, Lorenzo Natale, Francesco Nori, and Giorgio Metta. Robots with a sense of touch. *Nature Materials*, 15(9):921–925, 2016.

[7] C Hartmann, J Linde, S Dosen, D Farina, L Seminara, L Pinna, M Valle, and M Capurro. Towards prosthetic systems providing comprehensive tactile feedback for utility and embodiment. In *Biomedical Circuits and Systems Conference (BioCAS), 2014 IEEE*, pages 620–623. IEEE, 2014.

[8] Shih-Chii Liu and Tobi Delbruck. Neuromorphic sensory systems. *Current opinion in neurobiology*, 20(3):288–295, 2010.

[9] Christoph Posch, Daniel Matolin, and Rainer Wohlgenannt. A qvga 143 db dynamic range frame-free pwm image sensor with lossless pixel-level video compression and time-domain cds. *IEEE Journal of Solid-State Circuits*, 46 (1):259–275, 2011.

[10] P. Lichtsteiner, C. Posch, and T. Delbruck. A 128x128 120 dB 15 μs latency asynchronous temporal contrast vision sensor. *IEEE Journal of Solid-State Circuits*, 43(2):566–576, Feb 2008. ISSN 0018-9200. doi: 10.1109/JSSC.2007.914337.

[11] Alessandro Mortara. A pulsed communication/computation framework for analog vlsi perceptive systems. *Analog Integrated Circuits and Signal Processing*, 13(1):93–101, 1997. ISSN 1573-1979.

[12] Ravinder S Dahiya, Philipp Mittendorfer, Maurizio Valle, Gordon Cheng, and Vladimir J Lumelsky. Directions toward effective utilization of tactile skin: A review. *IEEE Sensors Journal*, 13(11):4121–4138, 2013.

[13] Bart J Kane, Mark R Cutkosky, and Gregory TA Kovacs. A traction stress sensor array for use in high-resolution robotic tactile imaging. *Journal of microelectromechanical systems*, 9(4):425–434, 2000.

[14] Myron A Diftler, CJ Culbert, Robert O. Ambrose, R Platt, and WJ Bluethmann. Evolution of the nasa/darpa robonaut control system. In *2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*, volume 2, pages 2543–2548. IEEE, 2003.

[15] R Walkler. Developments in dextrous hands for advanced robotic applications. In *Proc. the Sixth Biannual World Automation Congress, Seville, Spain*, pages 123–128, 2004.

[16] Vivek Maheshwari and Ravi F Saraf. High-resolution thin-film device to sense texture by touch. *Science*, 312(5779):1501–1504, 2006.

[17] Yoshiyuki Ohmura, Yasuo Kuniyoshi, and Akihiko Nagakubo. Conformable and scalable tactile sensor skin for curved surfaces. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 1348–1353. IEEE, 2006.

[18] John Jeka, Kelvin S Oie, and Tim Kiemel. Multisensory information for human postural control: integrating touch and vision. *Experimental Brain Research*, 134(1):107–125, 2000.

[19] Chiara Bartolozzi, Lorenzo Natale, Francesco Nori, and Giorgio Metta. Robots with a sense of touch. *Nature materials*, 15(9):921, 2016.

[20] Peer A Schmidt, Eric Maël, and Rolf P Würtz. A sensor for dynamic tactile information with applications in human–robot interaction and object exploration. *Robotics and Autonomous Systems*, 54(12):1005–1014, 2006.

[21] Alexander Schmitz, Marco Maggiali, Lorenzo Natale, Bruno Bonino, and Giorgio Metta. A tactile sensor for the fingertips of the humanoid robot icub. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2212–2217. IEEE, 2010.

[22] Carlo Ciliberto, Luca Fiorio, Marco Maggiali, Lorenzo Natale, Lorenzo Rosasco, Giorgio Metta, Giulio Sandini, and Francesco Nori. Exploiting global force torque measurements for local compliance estimation in tactile arrays. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3994–3999. IEEE, 2014.

[23] Joan Kangro, Silvio Traversaro, Daniele Pucci, and Francesco Nori. Skin normal force calibration using vacuum bags. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 401–406. IEEE, 2017.

[24] Paolo Dario and Danilo De Rossi. Tactile sensors and the gripping challenge: Increasing the performance of sensors over a wide range of force is a first step toward robotry that can hold and manipulate objects as humans do. *IEEE spectrum*, 22(8):46–53, 1985.

[25] ES Kolesar, CS Dyson, RR Reston, RC Fitch, DG Ford, and SD Nelms. Tactile integrated circuit sensor realized with a piezoelectric polymer. In *1996 Proceedings. Eighth Annual IEEE International Conference on Innovative Systems in Silicon*, pages 372–381. IEEE, 1996.

[26] Yoji Yamada, Takashi Maeno, Isao Fujimoto, Tetsuya Morizono, and Yoji Umetani. Identification of incipient slip phenomena based on the circuit output signals of pvdf film strips embedded in artificial finger ridges. In *Proceedings of the SICE Annual Conference*, pages 3272–3277, 2002.

[27] Jun-ichiro Yuji and Chikara Sonoda. A pvdf tactile sensor for static contact force and contact temperature. In *SENSORS, 2006 IEEE*, pages 738–741. IEEE, 2006.

[28] J Randall Flanagan and Alan M Wing. Modulation of grip force with load force during point-to-point arm movements. *Experimental Brain Research*, 95(1):131–143, 1993.

[29] G. Indiveri. A low-power adaptive integrate-and-fire neuron circuit. In *International Symposium on Circuits and Systems, (ISCAS), 2003*, volume IV, pages 820–823. IEEE, May 2003. doi: 10.1109/ISCAS.2003.1206342.

[30] Stefano Caviglia, Luigi Pinna, Maurizio Valle, and Chiara Bartolozzi. An event-driven posfet taxel for sustained and transient sensing. In *Circuits and Systems (ISCAS), 2016 IEEE International Symposium on*, pages 349–352. IEEE, 2016.

[31] S. Caviglia, M. Valle, and C. Bartolozzi. Asynchronous, event-driven readout of posfet devices for tactile sensing. In *Circuits and Systems (ISCAS), 2014 IEEE International Symposium on*, pages 2648–2651. IEEE, 2014.

[32] S. Caviglia, L. Pinna, M. Valle, and C. Bartolozzi. Spike-based readout of posfet tactile sensors. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 64(6):1421–1431, 2017.

[33] Ali Abou Khalil, Maurizio Valle, Hussein Chible, and Chiara Bartolozzi. Improved event-driven touch cmos sensor. PRIME 2017. [Accepted for Publication].

[34] Shih-Chii Liu, Jörg Kramer, Giacomo Indiveri, Tobias Delbra, Rodney Douglas, et al. *Analog VLSI: circuits and principles.* MIT press, 2002.

[35] D Ben Dayan Rubin, Elisabetta Chicca, and Giacomo Indiveri. Firing proprieties of an adaptive analog vlsi neuron. 2004.

[36] Alan L Hodgkin and Andrew F Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of physiology*, 117(4):500–544, 1952.

[37] P. Lichtsteiner, C. Posch, and T. Delbruck. A 128x128 120 dB 15 μs latency asynchronous temporal contrast vision sensor. *IEEE Journal of Solid-State Circuits*, 43(2):566–576, Feb 2008. ISSN 0018-9200. doi: 10.1109/JSSC.2007.914337.

[38] Juan Antonio Leñero-Bardallo, Teresa Serrano-Gotarredona, and Bernabé Linares-Barranco. A 3.6μs latency asynchronous frame-free event-driven dynamic-vision-sensor. *IEEE Journal of Solid-State Circuits*, 46(6): 1443–1455, 2011.

[39] Tobi Delbrück, Bernabe Linares-Barranco, Eugenio Culurciello, and Christoph Posch. Activity-driven, event-based vision sensors. In *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, pages 2426–2429. IEEE, 2010.

[40] S. Caviglia, L. Pinna, M. Valle, and C. Bartolozzi. Spike-based readout of posfet tactile sensors. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 64(6):1421–1431, 2017.

[41] Ravinder S Dahiya, Maurizio Valle, Giorgio Metta, and Leandro Lorenzelli. Posfet based tactile sensor arrays. In *ICECS*, pages 1075–1078, 2007.

[42] Francois Krummenacher and Norbert Joehl. A 4-mhz cmos continuous-time filter with on-chip automatic tuning. *IEEE Journal of Solid-State Circuits*, 23(3):750–758, 1988.

[43] Ali Abou Khalil, Maurizio Valle, Hussein Chible, and Chiara Bartolozzi. Cmos dynamic tactile sensor. In *CAS (NGCAS), 2017 New Generation of*, pages 269–272. IEEE, 2017.

[44] Roland S Johansson and J Randall Flanagan. Coding and use of tactile signals from the fingertips in object manipulation tasks. *Nature Reviews Neuroscience*, 10(5):345, 2009.

[45] J Andrew Pruszynski and Roland S Johansson. Edge-orientation processing in first-order tactile neurons. *Nature neuroscience*, 17(10):1404, 2014.

[46] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y Ng. Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5. Kobe, Japan, 2009.

[47] P. Motto Ros, M. Crepaldi, C. Bartolozzi, and D. Demarchi. Asynchronous DC-free serial protocol for event-based AER systems. In *2015 IEEE International Conference on Electronics, Circuits, and Systems (ICECS)*, pages 248–251, Dec 2015. doi: 10.1109/ICECS.2015.7440295.

[48] Romain Brette and Wulfram Gerstner. Adaptive exponential integrate-and-fire model as an effective description of neuronal activity. *Journal of neurophysiology*, 94(5):3637–3642, 2005.

[49] Pierre Del Moral. Nonlinear filtering: Interacting particle resolution. *Comptes Rendus de l'Académie des Sciences-Series I-Mathematics*, 325(6): 653–658, 1997.

[50] Pierre Del Moral et al. Measure-valued processes and interacting particle systems. application to nonlinear filtering problems. *The Annals of Applied Probability*, 8(2):438–495, 1998.

[51] P Del Moral, A Doucet, and A Jasra. On adaptive resampling procedures for sequential monte carlo methods. hal-inria rr-6700-2008. *Bernoulli*, page 82, 2011.