A NEUROMORPHIC APPROACH TO TACTILE PERCEPTION

LEE WANG WEI

NATIONAL UNIVERSITY OF SINGAPORE

2016

A NEUROMORPHIC APPROACH TO TACTILE PERCEPTION

Lee Wang Wei

B.Eng (Hons.), NUS

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

NUS GRADUATE SCHOOL FOR INTEGRATIVE

SCIENCES AND ENGINEERING

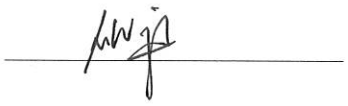
NATIONAL UNIVERSITY OF SINGAPORE

2016

Declaration

I hereby declare that this thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.



Lee Wang Wei

15 August 2016

Acknowledgements

This thesis represents a milestone in my four years of work at SiNAPSE. I have had an amazing experience working at the institute, crossing paths with some of the brightest minds from across multiple disciplines. SiNAPSE has evolved rapidly during this period, for which I was fortunate to be a part of the progress. I vividly remembered my first day to the lab, where almost all the benches were empty and I could sit wherever I wanted. In contrast, SiNAPSE today is filled to the brim with over 70 full-time members and many more collaborators, to the extent that PIs are often jostling for real estate to locate their new students and equipment.

Starting from scratch, I was given tremendous freedom to conceptualize and define my topic, and the liberty to purchase and build all the necessary equipment and tools to aid my research. The whole process, while rewarding at the end, was equally frustrating at the start as there was no legacy to build upon and no seniors to seek advice from. However, I am thankful for unique experiences like these, through which I learnt the importance to be independent, resourceful, creative and responsible for my own project.

First and foremost I wish to thank my advisor, Professor **Nitish Thakor**, director of SiNAPSE. Dr. Thakor was instrumental in guiding my research. With shrewd judgment and a clear sense of audience, he has taught me invaluable skills on how to present my ideas in a concise manner that is both relevant and attractive to readers and reviewers. His great wisdom and experience remains accessible to all of his students despite his busy schedule straddling between opposite sides of the globe. We have spent long hours on weekends debating on topics ranging from research progress to career goals

and even the meaning of life. Through his vast connections, I was exposed to various labs around the world, building important connections with renowned researchers from Hopkins and Stanford in particular. Thanks to him I was also given the opportunity to attend the Telluride neuromorphic workshop twice, increasing the visibility of my project and allowing me to work alongside remarkable individuals in the neuromorphic research community.

A special thanks to Dr.**Yen Shih-Cheng**, my advisor during my undergraduate years and the chairperson for my thesis committee, for the advice and support he has given. Also co-advisor during the first three years of my PhD programme, Dr. Yen's connection with the various faculties in the University and his familiarity with the administrative processes played a critical role in facilitating my research. From approving my purchases in the early days to the borrowing of equipment and sourcing of summer interns to assist my work, Dr. Yen was always there to help get things done. I would never have completed on time without his help and advice.

I have been fortunate to have an outstanding advisory committee to provide guidance along the way. I am particularly grateful to Dr. **Sunil Kukreja** for being my co-advisor during my final semester, and I praise the amazing amount of help and patience he rendered to transform my manuscripts from crude drafts into camera-ready papers. I would also like to extend my gratitude to Dr. **Arindam Basu** for his useful insights at the committee meetings despite his busy schedule at NTU.

The post-docs in the lab have been great role models to me. I am especially thankful to **Garrick Orchard** for being a great source of inspiration, and it was through him that I was introduced to the field of neuromorphic engineering. His presentations and live-demonstrations are always engaging and easy to follow, from which I learnt the importance of being a clear and confident presenter. **Charles Lun De-liao** is the highly driven and enterprising scientist who has shared with me many important insights, from building effective collaborations to tips on ascending the academic career ladder. Also, thanks to **Kian Ann** for sharing his technical expertise in designing and troubleshooting electronic circuitry.

There is a special group of people at SiNAPSE who made my graduate student life much more memorable — my peers! Thanks to **Roy, Aishwarya Bandla, Aishwarya**

Parthasarathy, Anoop, Abhishek, Rohan, Indu and **Mahdi** for their company in lab. From the lively discussions of science to the exchange of gossips, their presence gave warmth and life to the lab and provided me with much needed moral support for when things do not work out. Special mention goes to **Luke Osborn**, graduate student under Dr. Thakor at Hopkins, for reassuring me that life is not much greener on "the other side". I want to especially thank **Josue Orellana**, who despite having left SiNAPSE three years ago, has relentlessly kept up with our fortnightly video conversations even if it means waking up at 6 am on Monday mornings! I will always remember the times we traveled together, both in the US and in Singapore, as well as our creative conversations on all things under the sun. He has been an amazing friend, helping to keep my sanity in check through the ups and downs of graduate student life.

Having been in SiNAPSE during its early days, I have grown to appreciate the research assistants and support staff that have joined the institute over the years. I am thankful to **Freddy** for making SiNAPSE much more orderly than it used to be, and ensuring that the facilities such as printers and computers are always working. Also thanks to **Bernadette** and **Priyanka**, who have made what used to be week-long purchase processes so much faster and simpler. Thanks to **Marshal** and **Li Jing**, the soldering station is now much neater and serviceable than before. I am sure their duties extend way beyond what I have observed, and I am grateful for how they have transformed SiNAPSE into the conducive and productive research environment we see today.

My unforgettable experience as a graduate student would not be possible without the support from the NUS Graduate School of Integrative Sciences and Engineering (NGS). The highly flexible and student-centric curriculum allowed me to pursue subjects from the multiple disciplines necessary to understand and work in the field of neuromorphic engineering. I was also part of the NGS scholars alliance, and had the opportunity to organize our very own students symposium, through which I gained management and leadership skills that are lifelong and transferable. I am especially grateful to the educational funds provided by the NGS scholarship that facilitated my travels to international conferences and workshops during my years of candidature. Special thanks to **Irene** for the valuable advice on administrative matters, and passing down

the experiences and insights from seniors in the programme.

The final paragraphs are reserved for the people close to my heart: my family. I have an amazing and unique family. To my **Mom and Dad**, you are the best parents I could have ever had. You have given me everything, nurturing me and providing all the love and encouragement needed to make this thesis possible. Your support has been unconditional all these years, without which I would never be the person I am today. I love you both!

To my siblings **Wang Hao** and **Wen Qi**, you have been the best siblings I could have ever asked for. Thanks for doing my part of housework while I focused on research, and putting up with me when I work late into the nights. We have had some great times over the course of our childhood and I look forward to more as we grow and mature as adults.

Last but not least, to my fiancée **Yi Fang**, thank you for being so patient and supportive of me since my undergraduate years. You have cherished with me every great moment and supported me whenever I needed it. Having put up with the unnecessarily scientific conversations we had, you probably know more about tactile sensing than any mandarin teacher ever needs to know. Your ability to juggle the roles of being my secretary, counselor, teacher and manager at various stages of my life deserves special mention, and your magnanimity and optimism is humbling to observe. You are my soul mate and I love you!

Contents

Abstract

This dissertation aims to improve the scalability and responsiveness of tactile sensor arrays. Inspired by biology, we promote the utility of temporal structure in tactile signals to encode stimuli and drive learning processes. We challenge the need for accurate pressure readout, proposing instead a strategy of rapidly extracting change events from large numbers of sensing elements, resulting in sparse spatiotemporal representations analogous to spike patterns from mechanoreceptors. The feasibility of our concept is demonstrated with the realization of a large tactile sensor array capable of an unprecedented 5200 frames-per-second readout of 4096 sensor elements.

Utilizing both finite element analysis techniques and the high spatiotemporal resolution afforded by the sensor array, we conducted simulations and physical experiments to highlight the effectiveness of temporal coding in tactile sensing. Specifically, we show that the accuracy of contact discrimination is markedly improved when signals of high temporal precision are incorporated. In comparison, conventional signals obtained at lower sampling rates do not approach the same performance despite having higher pressure resolution. Moreover, our neuromorphic spatiotemporal representations of touch require almost 20 times less bandwidth to transmit, owing to their sparse nature.

Finally, we present an effective learning algorithm for processing the spatiotemporal patterns produced by our sensor array. Based on established convex optimization frameworks, a spiking neuron model is trained to fire at specific times when presented with a spatiotemporal input spike pattern. Through simulations, our formulation is shown to be superior to existing algorithms in terms of memory capacity, while incorporating

parameters that can be intuitively tuned to improve robustness to noise. The efficacy of our technique was demonstrated in a physical gait event detection application, delivering timely responses to gait events within 10 ms of the ground truth. We conclude that neuromorphic processing techniques offer considerable advantages in the development of responsive, robust and efficient tactile sensing systems suitable for robotics and prosthetic applications.

List of Tables

List of Figures

1 Introduction

To imagine that the dexterity of the musician, the artist or the craftsman lies in the special anatomical perfection of the human hand is a delusion that is apparent in the writings of most of those who have given attention to the subject. Such elevating studies can only be undertaken by those who mistakenly ascribe to the primitive human hand those aptitudes conferred on it by the specialized cerebral cortex.

Wood-Jones, 1944

Contents

1.1 Motivation

With rapid technological advancements, expectations of robots and their uses have changed. The role of robots have shifted from merely being an industrial tool in a controlled environment, to smart devices that assist and interact with humans. Applications include artificial limbs and exoskeletons that assist the handicapped, medical robots that perform surgery and social robots that simulate human behaviour for human-robot interaction. This transition gives rise to new challenges that have to be solved in order for robots to safely interact with humans, providing services in manufacturing, entertainment, education, health care and assistance.

In a highly complex and dynamic world, an autonomous system has to effectively deal with environmental uncertainties. Besides visual and auditory sensing, the ability to detect touch is particularly important as it instills bodily awareness, allowing the system to differentiate itself from surrounding objects. The sense of touch differs from vision and audition as it involves complex physical interactions, playing a critical role in the estimation of the physical properties such as surface texture, hardness, material, shape, etc. Rich physical interactions between real-world objects are also better understood when touch sensing is available, such as the way surfaces deform upon contact, the friction between surfaces, and the weight distribution of various objects.

Despite its importance, touch sensing in robotics has yet to receive as much attention compared to other sensing modalities [1]. Early works are largely restricted to intrinsic touch sensing, where joint forces/torques are used to estimate interactive forces. Intrinsic touch sensing is convenient to implement and largely sufficient for industrial applications. Interest in extrinsic touch sensing, or tactile sensing, has only increased recently with the advent of dexterous multi-fingered robotic hands and the rising need for robots to operate in unstructured environments. For the above applications, it is a prerequisite for robots to have whole body tactile sensing capabilities. While several tactile sensitive skin projects have been undertaken, few have achieved the density, sensitivity and responsiveness comparable to that of the human skin [1–3].

Evidently, tactile sensing in biological systems have superior performance and robustness. Human reflexes can respond at a rapid $\sim 65\,\text{ms}$ while processing signals from

hundreds of thousands of mechanoreceptors [4–9]. While mechanoreceptors are noisy, non-linear, hysteric and exhibit large variability within and among individuals, studies have shown that tactile perception of force remains remarkably consistent across different subjects [10]. Knowledge of how information is encoded, transmitted and processed in biological systems may hold key to enabling robust, responsive and affordable full-body tactile sensing.

To identify the factors contributing to nature's computational efficiency and robustness to uncertain environments, a small group of labs in the field of neuromorphic engineering focus on emulating the neuronal organization and function of nervous systems through electronic devices and computer algorithms. One example is the use of events (spikes) to encode and propagate information, much like how neurons communicate through action potentials in biology [11]. Modeled after their biological counterparts, neuromorphic sensors such as silicon retinas and cochleas have been developed to directly capture environmental stimuli as spikes [12]. Outputs from these devices are often times more sparse, and hence better suited for real time applications. Besides sensors, bio-inspired computational frameworks such as spiking neural networks have also been developed [13]. By modeling the complex dynamics of spiking neurons, learning and plasticity mechanisms observed in biology can be implemented. Coupled with custom hardware platforms, the networks could perform computations much faster than conventional computers, while drawing significantly less power [14]. There is thus potential to revolutionize the field of tactile sensing through the use of neuromorphic principles and hardware. Specifically, we aim to reduce sampling latency and improve computational efficiency of tactile sensing for large populations of sensor elements.

1.2 Original Contributions

This research aims to leverage on recent discoveries in biological tactile sensing and combine them with the algorithms and hardware developed in neuromorphic engineering to improve the scalability and computational efficiency of tactile sensing. The original contributions that will be presented in this dissertation are listed below:

1. Optimized processes for multiple designs of low cost flexible tactile sensor arrays, from the initial fabric designs presented in Chapter 3, to the insole sensor described in Chapter 6.5 and finally the inkjet printed sensor in Chapter 4. In a separate publication [15], the design has been adapted to accommodate commercially available prosthetic hands to achieve closed loop feedback in grasp control.

2. Developed and characterized a tactile sensing system with an unprecedented 4096 sensing elements at 5.2 kHz sampling rate. This platform enables the physical validation of our subsequent hypotheses.

3. Conceptualized and established the use of spatiotemporal patterns in tactile sensing, where the importance of temporal precision when acquiring tactile data is highlighted. This includes the use of finite element models and physical experiments on the sensor array developed.

4. Conceptualized and developed a supervised learning algorithm that responds to spatiotemporal spike patterns with temporal precision. The algorithm demonstrates how the spiking representations of tactile signals may be utilized for real-world applications. In an example application, the algorithm responded to gait events within 10 ms of the reference system.

1.3 List of Publications

The scholastic publications generated during the course of this research are listed below:

1. **Lee W. W.**, Kukreja S. L., Thakor N. V., "Discrimination of Dynamic Tactile Contact by Temporally Precise Event Sensing in Spiking Neuromorphic Networks," In preparation for *IEEE Trans. on Biomedical Circuits and Syst.*

2. **Lee W. W.**, Kukreja S. L., Thakor N. V., "CONE: Convex-Optimized-Synaptic Efficacies for Temporally Precise Spike Mapping," *IEEE Trans. Neural Netw. Learning Syst.*, vol.PP, no.99, pp.1-13, doi:10.1109/TNNLS.2015.2509479, 2015.

3. Roy, S., San, P. P., Hussain, S., **Lee, W. W.**, & Basu, A. "Learning Spike Time Codes Through Morphological Learning With Binary Synapses," *IEEE Trans. Neural Netw. Learning Syst.*, vol.27, no.7, pp.1572-1577, 2015.

4. **Lee, W. W.**, Chen H. Y., Ren H., Kukreja, S. L., & Thakor, N. V. (2016, June). "FPGA Implementation of a FA-1 Mechanoreceptor Model for Efficient Representation of Tactile Features". in *Int. Conf. Biomed. Robot. Biomechatron (BioRob)*. Singapore: IEEE, Jun. 2016, pp. 243-246.

5. **Lee, W. W.**, Kukreja, S. L., & Thakor, N. V., "A Kilohertz Kilotaxel Tactile Sensor Array for Investigating Spatiotemporal Features in Neuromorphic Touch," in *Biomed. Circuits Syst. Conf. (BioCAS)*, Atlanta, GA: IEEE, Oct. 2015, pp. 1-4.

6. **Lee, W. W.**, Kukreja, S. L., & Thakor, N. V., "Live demonstration: A Kilohertz Kilotaxel Tactile Sensor Array for Investigating Spatiotemporal Features in Neuromorphic Touch," in *Biomed. Circuits Syst. Conf. (BioCAS)*, Atlanta, GA: IEEE, Oct. 2015, pp. 1-1.

7. **Lee, W. W.**, Yu, H., & Thakor, N. V., "Gait Event Detection through Neuromorphic Spike Sequence Learning". in *Int. Conf. Biomed. Robot. Biomechatron. (BioRob)*, Sao Paulo, Brazil: IEEE, Aug. 2014, pp. 899-904.

8. **Lee, W. W.**, Cabibihan, J. J., & Thakor, N. V., "Bio-mimetic Strategies for Tactile Sensing," in *Sensors*, Baltimore, MD: IEEE, Nov. 2013, pp. 1-4.

9. Khin, P. M., Low, J. H., **Lee, W. W.**, Kukreja, S. L., Ren, H., Thakor, N. V., & Yeow, C-H., "Soft Haptics Using Soft Actuator and Soft Sensor", in *Int. Conf. Biomed. Robot. Biomechatron. (BioRob)*, Singapore: IEEE, Jun. 2016, pp. 1272-1276.

10. Low, J. H., **Lee, W. W.**, Khin, P. M., Kukreja, S. L., Ren, H., Thakor, N. V., & Yeow, C-H., "A Compliant Modular Robotic Hand with Fabric Force Sensor for Multiple Versatile Grasping Modes", in *Int. Conf. Biomed. Robot. Biomechatron. (BioRob)*, Singapore: IEEE, Jun. 2016, pp. 243-246.

11. Rasouli, M., Ghosh, R., **Lee, W. W.**, Thakor, N. V., & Kukreja, S., "Stable Force-myographic Control of a Prosthetic Hand using Incremental Learning", in *37th Annu. Int. Conf. Eng. Med. Biol. Soc. (EMBC)*, Milan, Italy: IEEE, Aug. 2015, pp. 4828-4831.

12. Osborn, L., **Lee, W. W.**, Kaliki, R., & Thakor, N. V., "Tactile Feedback in Upper Limb Prosthetic Devices using Flexible Textile Force Sensors", in *Int. Conf. Biomed. Robot. Biomechatron. (BioRob)*, Sao Paulo, Brazil: IEEE, Aug. 2014, pp. 114-119.

1.4 Organization of Dissertation

The rest of the thesis is organized as follows. Chapter 2 presents an overview of biological and artificial tactile sensing. An introduction to neuromorphic spike based processing is provided at the end of the chapter. Chapter 3 highlights the results of our early attempts at demonstrating applicability of our neuromorphic approach. Chapter 4 presents a high-speed tactile sensing array developed to capture tactile signals with unprecedented spatiotemporal resolution. This is followed by an investigation on the use of temporal structure for representing tactile data, discussed in Chapter 5. We then introduce a novel learning algorithm for spike-based tactile signals in Chapter 6. Finally, Chapter 7 summarizes the thesis and discusses the open problems for future development.

2

Literature Review

Despite the fact that the skin is, from the evolutionary standpoint, the oldest of the sensitive tissues of the body, it has yielded up its secrets reluctantly.

Gerald, 1972

Contents

2.1 Overview of The Human Cutaneous System

2.1.1 Introduction

Touch sensing in humans is a distributed process that involves the entire area of the body. Unlike vision and hearing, touch requires physical interaction with the external world. While the blind and deaf could still adapt to their handicap, those who lost their sense of touch constantly suffer bruises, burns and broken bones due to the lack of warning provided by touch and pain [16]. Experiments have shown that with their hands temporarily anesthetized, subjects tend to apply much more force than necessary when carrying out tasks with their fingers and hands [16]. The sense of touch also plays a critical role in direction and spatial orientation. Astronauts in a microgravity environment have reportedly relied on artificial tactile cues, such as pressure applied on the sole of the feet, to determine up from down [17]. These examples demonstrate the importance of the sense of touch in understanding the environment and guiding motor behavior. To design a useful robotic tactile sensing system, it is prudent to understand how tactile sensing is achieved in biological systems.

The sense of touch consists of two sub-modalities — cutaneous and kinesthetic senses. Kinesthetic sensing refers to force and torque information provided by receptors in the muscles, tendons and joints, and is analogous to intrinsic touch sensing in robotics. They primarily provide proprioception, which is the knowledge of the position and orientation of limbs in space. On the other hand, cutaneous sensing is achieved by sensory receptors embedded in the skin, providing information such as pressure and thermal distribution as well as pain sensations. This review primarily focuses on the cutaneous/tactile component of touch, hereby referred to as tactile sensing.

In tactile sensing, mechanical pressures and distortions are picked up by mechanoreceptors, enabling the perception of features such as vibrations, texture, friction, softness, etc. Other receptors responsible for thermal gradients (thermoceptors) and pain from mechanical, thermal or chemical stimuli (nociceptors) are also present, but are of less interest in the context of tactile sensing.

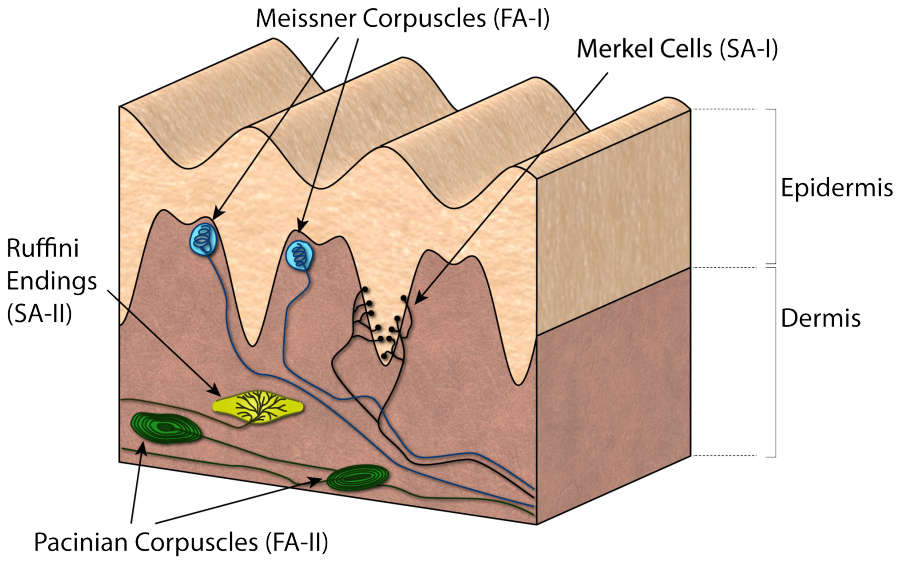


Figure 2.1: Illustration of mechanoreceptors in the glabrous skin

2.1.2 Mechanoreceptors

There are four main types of mechanoreceptors in the glabrous (hairless) areas of the human skin — Meissner corpuscles, Pacinian corpuscles, Merkel cells and Ruffini corpuscles [18]. Meissner corpuscles and Merkel receptors are located closer to the surface of the skin, near the epidermis. Meissner's corpuscles are the most common mechanoreceptors in the glabrous skin, accounting for about 40% of the mechanoreceptors of the human hand, while the Merkel cells account for 25% of the receptors. On the other hand, the Pacinian corpuscles and Ruffini endings are embedded more deeply in the dermis and subcutaneous tissue, with the Pacinian corpuscles making up 10-15% of receptors in the hand (See Fig 2.1 for illustration). The density of innervation varies across the body. Fingertips are estimated to contain about 241 mechanoreceptors per square centimeter, while the palm contains only 58 mechanoreceptors per square centimeter [4].

Mechanoreceptors, or tactile afferents, can be characterized by two main properties — rate of adaptation and size of the receptive field. Mechanoreceptors respond to stimuli by firing action potentials (AP). Fast adapting (FA) mechanoreceptors, such as Meisnner corpuscles (FA-I) and Pacinian corpuscles (FA-II), fire bursts of APs during stimulus onset and offset, but remain silent during steady state. On the other hand, slow adapting (SA) mechanoreceptors such as Merkel cells (SA-I) and Ruffini corpuscles (SA-II) will

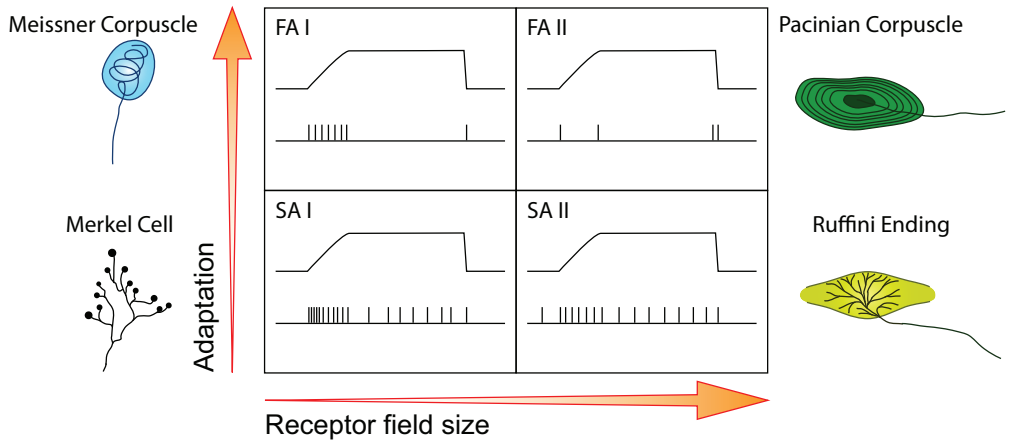


Figure 2.2: Spike response of mechanoreceptors.

actively fire when the stimulus is in contact with its receptive field (Fig 2.2). FA-I afferents exhibit sensitivity to dynamic skin deformations of relatively high frequency, while SA-I afferents respond more to lower-frequency skin deformations. There are more FA-I than SA-I afferents in fingertips, thus indicating the importance of detecting dynamic mechanical events (i.e. contact forming or breaking, or scanning across a textured surface) [4]. On the other hand, FA-II can be excited when hand-held objects contact or break contact with other objects, while SA-II afferents respond to remotely applied lateral stretching of the skin, which usually corresponds to tangential shear strain during object manipulation. SA-II afferents also contribute to kinesthetic sensing by reacting to tangential forces that act on the skin situated at the joints.

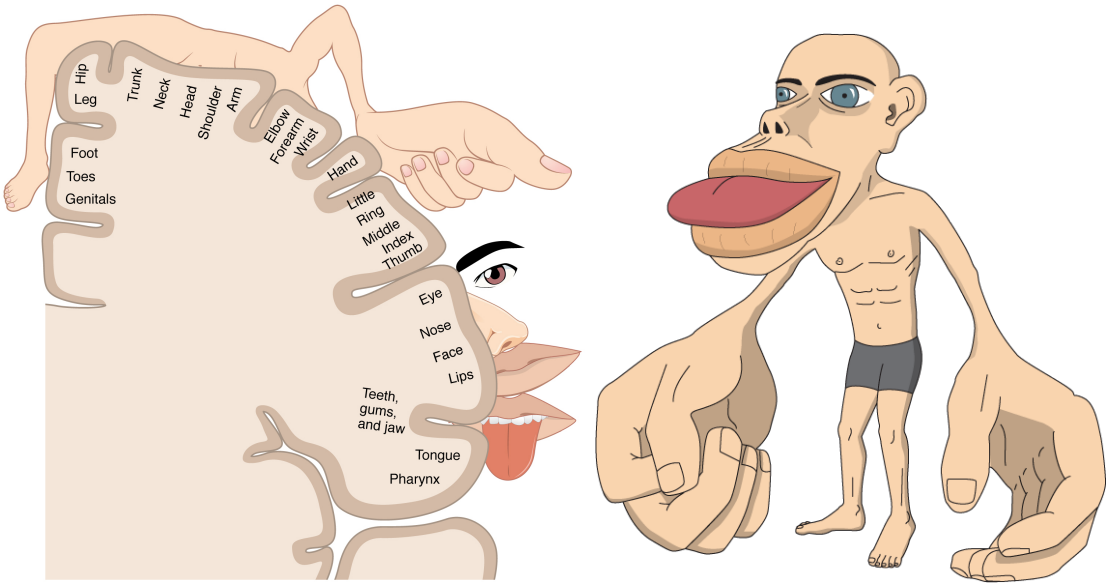
Receptive field refers to the region of skin within which a tactile stimulus will evoke a response from an innervating mechanoreceptor. Consistent with how mechanical forces are dispersed in elastic media, receptors that are closer to the surface of the skin, such as Meissner and Merkel receptors, have smaller receptive fields that are well defined. In contrast, deep receptors like the Pacinian corpuscles and Ruffini endings have large receptive fields with less defined boundaries (Fig 2.2). It should be noted that receptive fields frequently overlap, thus providing an intrinsic mechanism for fault tolerance and improving robustness [4].

The sensitivity of human tactile sensing may be characterized based on three attributes — spatial resolution, temporal resolution and pressure resolution. Spatial resolution is traditionally measured using the two points threshold method [19], and

has been found to vary from one region of the body to another. The fingertips have the highest spatial resolution at 1 mm, while the thigh, shoulders and belly have the lowest resolution (30 mm) [16]. On the other hand, the temporal resolution, i.e. the smallest detectable temporal interval between two events, is 1.4 ms [20] for stimuli delivered at the same location, and 30-50 ms for stimuli delivered at different locations [21].

Pressure thresholds vary not only at different locations on the body, but also across gender. Controlled pressure sensitive studies reveal that mean threshold values average about 0.158gm on the palm and 0.055 g on the fingertips of men, and 0.032 g and 0.019 g on the corresponding sites of women [22]. It is also estimated that 90% of the SA-I and FA-I mechanoreceptors will respond to stimulus forces of as low as 5 mN [23]. However, the range of sensitivities have been found to vary significantly from 1 mN to 100 mN [23]. Despite the apparent inconsistency in mechanoreceptor properties, psychological studies demonstrate striking similarity in the perception of force magnitude across individuals, suggesting that the central nervous system is highly involved in calibrating the force sensations.

It has been argued that traditional studies in characterizing the sensitivity of receptors offer little insight into the use of the signals during object manipulation [24]. This is because the forces that are involved in manipulation are typically different in comparison to those experienced during gentle touch. For instance, distributed patterns of stresses and strains that develop in the skin during contact can activate receptors situated away from the contact area [25–27]. This implies that the effective receptive field of a receptor is significantly larger than suggested by earlier studies. More importantly, the larger and overlapping receptive fields can enhance the encoding of spatiotemporal information [28]. As such, the mapping between contact events and afferent responses are highly complex, and should not be visualized as a two-dimensional pressure map alone. While researchers have created mechanical models of the fingertip for predicting mechanoreceptor responses [29–31], no model can provide a sufficiently realistic prediction as of now [24].



(a) Map of skin regions on the somatosensory cortex.[1]

(b) Homunculus[2]

Figure 2.3: Map of the body on the cortex.

2.1.3 Maps of the Body on the Cortex

Like the visual cortex, the somatosensory cortex is spatially organized into maps that correspond to locations on the body. However, the maps do not represent the body in actual proportions. Some areas on the skin occupy a disproportionately large area of the cortex (Fig. 2.3). For instance, the region corresponding to the index finger is as large as the region representing the entire forearm. This cortical magnification reflects the variable density of mechanoreceptors in the skin, leading to a much higher spatial resolution in the fingertips than other regions of the body.

The somatotopic map is plastic and self-organizing. Research has shown that the size of a representation changes according to how often the area of skin is used. When monkeys are trained to complete a task that involved the extensive use of a particular location on one fingertip, the respective area in the cortex was shown to have greatly expanded over the course of 3 months [32].

Temporal relationships between stimuli play an important role in maintaining the organization of somatotopic maps. A classic experiment on monkeys showed that co-

[1]Illustration from `http://cnx.org/content/col11496/1.6/`
[2]Illustration from `http://www.maxplanckflorida.org/fitzpatricklab/homunculus/`

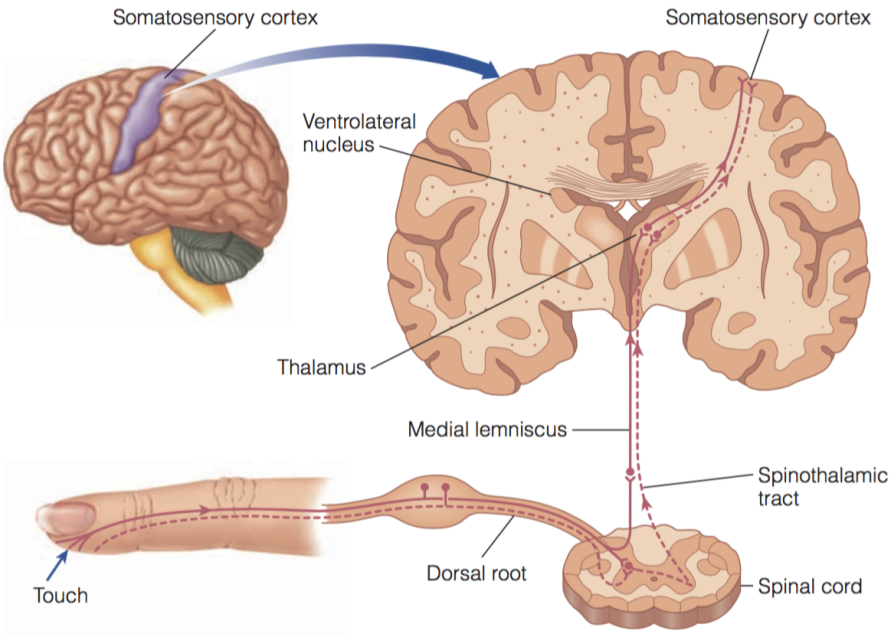


Figure 2.4: Illustration of the signalling pathway from mechanoreceptors to somatosensory cortex, obtained from [16].

stimulation of skin-patches that are remotely apart establishes new spatial relationships in the cortex [33]. Humans experience the *cutaneous rabbit* illusion, where tactile stimuli presented in quick succession are perceived to be moving closer together [34, 35]. In a separate example on human subjects, separate areas of skin are stimulated immediately one after another, resulting in a reduction in the perceived distance between the stimulated regions [36]. The adaptive ability of the somatosensory cortex contributes greatly to the robustness of biological tactile sensing, achieving consistent representations even as skin properties change over time.

2.1.4 Coding and Transfer of Tactile Information

When an object is touched, the skin conforms and maintains the same local contour as the surface in contact. This deformation is projected to a population of receptors in the skin, each responsible for a small area of the contour. Receptors then respond to its preferred stimulus (i.e. static or dynamic stress/strain) by emitting action potentials. This signifies that no single afferent, or class of afferents can provide the complete information of the contact event.

Signals from receptors in the skin travel in bundles of peripheral nerves that enter the spinal cord through the dorsal root. The axons from each receptor do not synapse until the spinal column, suggesting minimal compression on spatial resolution at this juncture. There are two major pathways in the spinal cord for the cutaneous system, namely the medial lemniscal pathway and the spinothalamic pathway (Fig. 2.4). The medial lemniscal pathway consists of large fibers that carry signals pertaining to touch and proprioception, while the spinothalamic pathway consists of smaller fibers that carry signals related to temperature and pain [16]. The medial lemniscal pathway propagates information faster than spinothalamic pathway due to heavy myelination. Majority of fibers from both pathways synapse in the ventrolateral nucleus in the thalamus, which are then projected to the somatosensory receiving areas (S1 and S2) in the parietal lobe of the cortex [37].

Researchers have long believed that the firing rate of mechanoreceptors is the primary method of conveying information. This would require at least two action potentials per afferent. However, the speed at which the brain detects and corrects prediction errors in the tactile modality suggests that afferents only have enough time to fire one spike [5, 38]. In recent studies, researchers explained this discrepancy by demonstrating how the relative timing of the first spikes elicited from a population of mechanoreceptors provides precise information about the shape of the contacted surface as well as the direction and magnitude of the force exerted on the hand [38]. This method of information encoding, known as relative latency coding [11], could convey information fast enough to account for the speed at which the tactile signals are used in object manipulation tasks. Such a code arises naturally, as the high density of mechanoreceptors respond asynchronously when the skin deforms under pressure [39].

In a recent paper, Pruszynski *et. al.* [40] demonstrated that FA-I and SA-I receptors are also selectively sensitive to the orientation of edges. These receptors have multiple transduction sites that are distributed non-uniformly, resulting in complex receptive fields with enhanced sensitivity towards certain edge orientations. Through multiple overlapping receptive fields, edge orientation is thus encoded in both the intensity and temporal structure of a population response.

A separate paper by Calgero *et. al* [41] investigated the importance of temporal

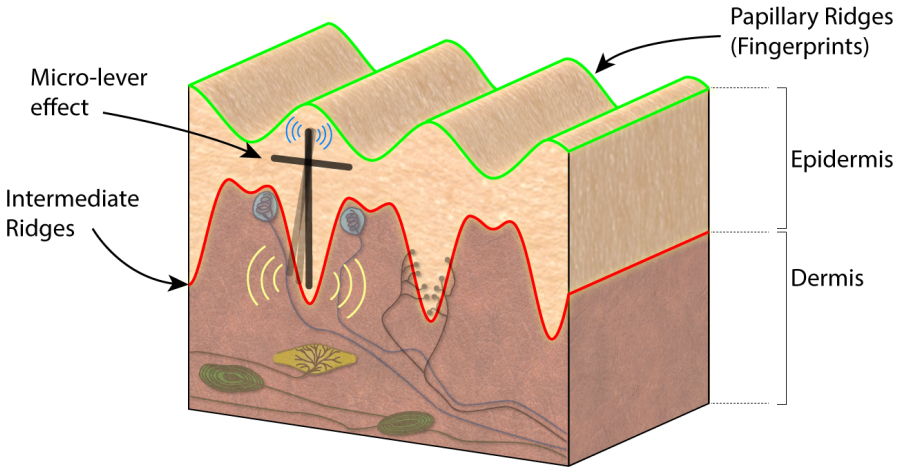


Figure 2.5: Intermediate and papillary ridges form a system of micro-levers to amplify surface displacement. The SA-I and FA-I receptors are typically situated near the intermediate ridges, thus enhancing their spatial resolution and sensitivity to skin deformation.

structure in signals used for touch restoration through peripheral nerve stimulation in human subjects. They demonstrated improved discrimination of surface coarseness when temporally consistent signals were used for stimulation. These converging evidences suggest that temporal structure in tactile signals carry important information that is utilized at various stages of somatosensory processing.

On the other hand, information contained in rate codes should not be disregarded. The lack of consistent relationships between first spike latencies and rate coding suggests that they carry different information, which may be processed by separate pathways [24]. Studies in the primary somatosensory (S1) cortex of Rhesus macaques suggests that vibratory stimuli from texture scanning is best represented at high temporal resolutions, while the amplitude of pressure does not require the timing of spikes to be as precise [42]. The ability to multiplex several streams of information through the rate and timing of spikes allow for more efficient use of the available communication bandwidth.

Spike codes and processing methods will be discussed in more detail in Section 2.3.2.

2.1.5 Skin Mechanics

The mechanics of the skin affects the relationship between surface indentations and the stress/strain that triggers the various mechanoreceptors. The performance of the

tactile sensing system is hence greatly dependent on the physical attributes of the skin. The human skin has been shown to be multilayered, nonlinear, nonhomogeneous and viscoelastic. Its thickness varies from 0.6-0.8 mm with a Young's modulus of about $4 \times 10^5\,\mathrm{N/m^2}$ for the epidermis [43]. Situated beneath the epidermis is a much softer dermis layer with Young's modulus 10-10000 times less [44]. The epidermis and dermis rest on a deformable system of muscles and fat, allowing the skin to readily conform to the surface in contact [45].

The skin at the fingertips contain papillary ridges commonly referred to as fingerprints. By having an uneven external surface, vibrations are generated when the finger is moved across a textured surface, enhancing the perception of texture and grasp stability [46]. Besides fingerprints, intermediate ridges exist between the dermis and the epidermis, forming in a physical interlock (Fig. 2.5). The interlocking patterns prevent one layer from sliding over the other, and also works in tandem with the papillary ridges to form a lever-arm mechanism which amplifies the fine displacements occurring at the surface [47]. With the SA-I and FA-I receptors situated around the intermediate ridges, spatial resolution and sensitivity is enhanced [45, 48]. In contrast, SA-II and FA-II receptors are buried deep within the dermis, resulting in large receptive fields and low spatial resolution. Various attempts at modeling the effects of transducer placement along the intermediate ridges have been reported [44, 49, 50].

2.1.6 Discussion

To achieve full body tactile sensing, it is almost universally agreed that a skin-like layer made up of a distributed array of transducers is needed. Here we provide some design hints for the development of tactile sensors that cover large surfaces, based on our observations of the human tactile sensing system.

Firstly, the sensing elements need not be homogeneously distributed throughout the skin. Instead, the density may vary depending on the function and body site. Sensitive regions such as end effectors should be equipped with high density arrays, while sensor arrays with lower spatial resolution would suffice for less important areas such as the shoulders and the torso.

Secondly, the biological cutaneous system has demonstrated high sensitivity over a wide dynamic range. However, this was achieved by having many mechanoreceptors each with a different threshold and functional range [23]. By leveraging on the huge plasticity and learning capabilities of the brain, the distributed response is fused to provide a remarkably consistent estimate [10]. Likewise when developing electronic skins it is not necessary for all the transducers to behave uniformly, as long as the processing algorithm is sufficiently intelligent to handle the varied responses. This distributed approach would be intrinsically more robust and less costly compared to the use of precisely calibrated sensors, since the performance of the system is less dependent on the consistency of individual sensing elements.

We have observed mechanoreceptors of various response behaviors in biological tactile sensing. A similar approach may be taken when designing artificial skins, where sensing elements from a heterogeneous mix of transduction principles may be used within the same system. For instance, piezoelectric sensors are ideally suited for sensing dynamic changes, while capacitive or piezoresistive elements would be useful for sustained pressure measurements. The signals of these events can also be transmitted using separate pathways with different transfer rates. Dynamic signals usually indicate a transition from one control state to another [24], and should be communicated under strict timing constraints, while pressure intensity measurements can be transmitted using progressive encoding techniques, adapting to the available bandwidth of communication channels.

Recordings from the somatosensory cortex reveal neurons tuned to progressively complex stimuli with each successive layer of projection [51]. In designing large tactile arrays, it is prudent to include a hierarchical architecture of modules to perform signal pre-processing and local computation to reduce the amount of information transfer to the central processing unit of the robot. In addition, certain reflex actions such as force control during grasping may be implemented at lower levels to reduce the overall workload of higher order computational units of the system.

The human skin is found to be a soft, compliant and highly conformable medium. Intelligent placement of mechanoreceptors, together with a system of papillary and intermediate ridges have been shown to significantly enhance texture and edge detection. By introducing an elastic covering over the sensors, better conformability and protection

can be achieved in artificial tactile arrays. Moreover, the covering can also be textured with ridges to improve its sensitivity towards textures [46].

While properties such as linearity and low hysteresis are desirable, they may not be necessary. The fact is that mechanoreceptors are non-linear, hysteric, noisy and slow. However, the somatosensory cortex actively learns, organizes and intrinsically calibrates the system, even as mechanoreceptors undergo wear and tear. Similarly, a robust and fault tolerant artificial skin would require intelligent learning algorithms to extract useful information from a population of low quality sensors.

In summary, this section has presented the ways which biological systems acquire and process somatosensory data. While mimicking every aspect of the biological tactile system is probably impractical due to engineering constraints, the understanding of tactile sensing in humans nevertheless provides useful insights on the organization and behavior of organisms in dynamically changing environments. The performance of biological tactile sensing also helps to define the specifications when designing robotic tactile sensors and electronic skins.

2.2 Artificial Tactile Sensing

2.2.1 Overview

Research on artificial tactile sensors has been active for over 40 years [47]. In its early days, touch sensing in robotics was largely accomplished using intrinsic touch sensors. The term refers to sensing elements that measure contact forces without physically contacting the object. Examples include torque sensors or encoders embedded in the joints of robots. While sufficient for industrial applications, intrinsic touch sensors do not have enough spatial resolution for manipulation and motion planning in unmodelled environments. This review focuses instead on extrinsic touch sensors, which are dedicated sensing elements distributed on the external surface of the robot that measure contact parameters. Extrinsic tactile sensors covering a significant surface area are also commonly referred to as an artificial/electronic skin, or e-skin in the literature.

A large variety of tactile sensors capable of handling and measuring various contact parameters have been reported in literature. Despite the wide spectrum of sensors currently available, touch sensing has been of limited use in existing robotic platforms. Challenges unique to tactile sensing, such as the distributed nature of sensors, the varying spatiotemporal resolutions required at different parts of the body, and the need to detect multiple contact parameters meant that an all rounded solution has yet to exist. This section aims to summarize existing technologies and approaches for tactile sensing, highlighting the bottlenecks and discussing possible solutions.

In general, research on robotic tactile systems can be hierarchically categorized as shown in Fig 2.6, where the system is organized into transduction, readout, feature extraction, data transmission and sensor fusion stages before reaching the high level controller.

At the lowest level, transduction refers to the conversion of mechanical signals such as pressure and vibration into a measurable change in electrical property, such as resistance or capacitance. Typically, dedicated circuitry is used to further convert changes in electrical property to a variation of electrical current or voltage. The circuit may also perform signal conditioning such as amplification and cross-talk cancellation to improve signal to noise ratio. The next stage involves the conversion of voltage

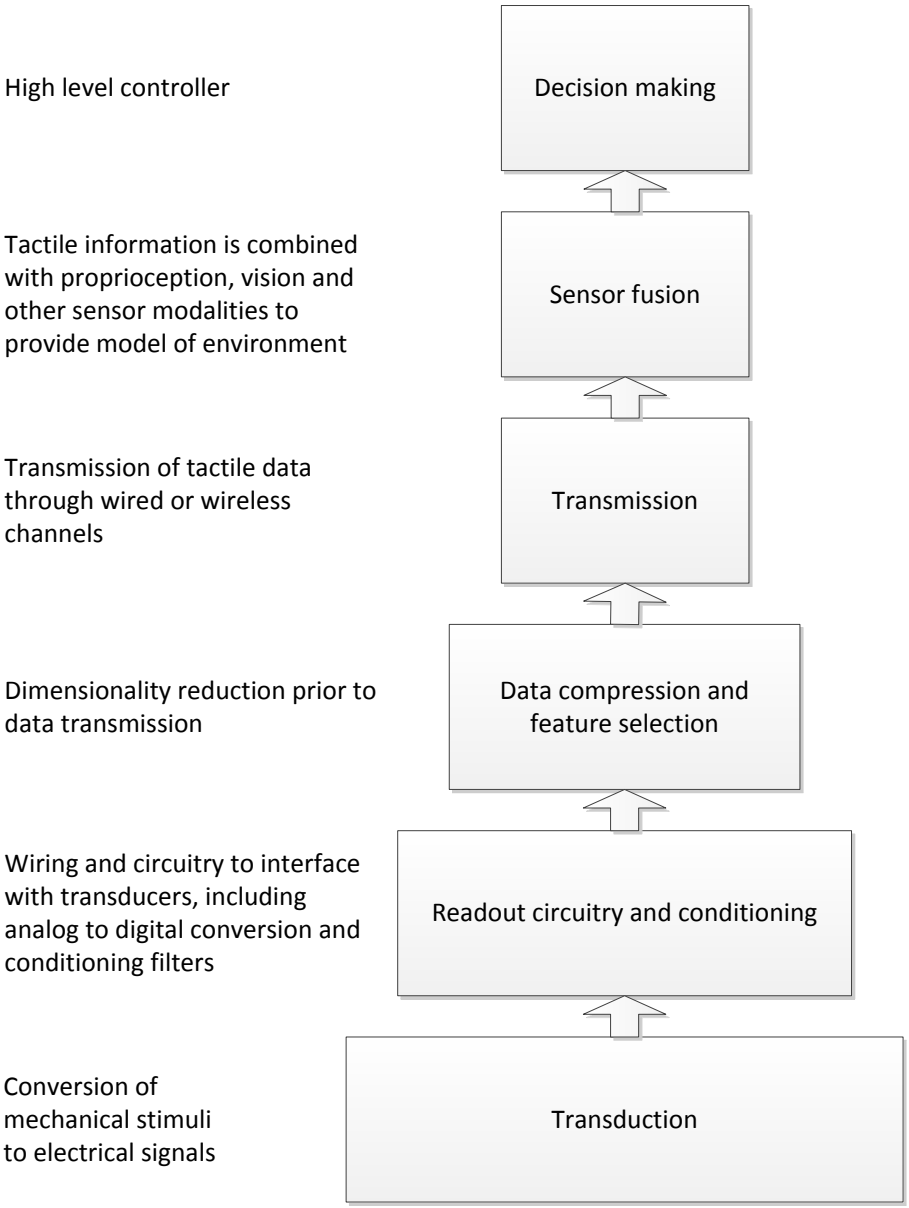


Figure 2.6: Hierarchical functional diagram of a typical tactile sensing system.

variations to digital representations, an analog to digital conversion (ADC) stage is needed. After digitization, feature selection is typically applied to derive concise and reliable representations of the tactile data. Finally, depending on the application, tactile stimuli are recognized by algorithms and decisions are made at the highest level.

The rest of the section is organized as follows: A survey of the state-of-the-art transducer technologies for artificial skins is given in Section 2.2.2. This is followed by a review of physical and mechanical issues such as sensor distribution, placement, routing of wires and conformability of sensors in Section 2.2.3. Interfacing electronics, communication and related issues are discussed in Section 2.2.4, while topics such as higher order processing and tactile perception are discussed in Section 2.2.5. Finally, a general discussion and summary is provided in Section 2.2.6.

2.2.2 Transduction

Tactile sensors of nearly all modes of transduction have been reported [47,52], with their working principles described in detail. This section presents a selection of techniques that are potentially scalable to cover non-uniform surfaces.

Piezoresistive

Piezoresistive pressure transducers are made from materials whose resistance changes with force/pressure applied. These materials include conductive rubber, conductive polymers, conductive gels, conductive fabrics and yarns, force sensitive resistors (FSR) and pressure sensitive ink etc. These materials are usually low cost, and a number of products are also commercially available from companies such as Interlink and Tekscan. Piezoresistive tactile sensors have been featured in several robotic hands [53, 54]. Due to its low cost, good sensitivity, low noise and simple interface electronics, piezoresistive sensors are very appealing.

Quantum tunnel composites (QTC), available commercially from Peratech [55], are materials capable of transforming from a perfect insulator to a metal like conductor when deformed by compressing, twisting and stretching. The circuitry for integration of QTC tactile arrays are similar to piezoresistive sensors. QTC based tactile sensors are reported in [56] and in [57], the sensor density is better than that of human fingertips.

The piezoresistive effect can also be achieved through mechanical means. In [58], a sheet of polydimethylsiloxane (PDMS) was patterned with micro-structured deformable pyramids and coated in conductive ink. When stacked with a flat conductive surface, applying pressure leads to a deformation of the pyramids which increases contact area, resulting in reduced resistance between the top and bottom surfaces.

Sensors based on the resistive effect tend to suffer from non-linearity and hysteresis, and also tend to have a comparatively higher power consumption.

Capacitive

Capacitive pressure sensing is a popular alternative to piezoresistive techniques. Capacitive sensors consist of two conductive parallel plates separated by a dielectric material. The capacitance value is represented by $C = A\epsilon_0\epsilon_0/d$, where A is the area between the plates, ϵ_0 is the permittivity of free space, ϵ_1 is the relative permittivity of the dielectric material and d is the distance between the plates. During stimulus onset, capacitance changes due to a change in d as the material is compressed.

Widely used in robotics, capacitive pressure sensors arrays can be made very dense. An 8×8 tactile array with $1mm^2$ area is reported in [59]. Capacitive sensor arrays are also commercially available from companies such as RoboTouch and DigiTacts from Pressure Profile Systems [60]. Components used for capacitive touchscreens such as "capacitance to digital converter" chips like the AD7147 from Analog Devices [61] can also be adapted for pressure sensing. While capacitive sensors are very sensitive, care must be taken to minimize parasitic capacitance which exist between wires and components besides the sensor elements. Moreover, severe hysteresis may be present depending on materials used.

One should be careful not to confuse capacitive pressure sensing to the capacitive sensing widely used in touchscreen interfaces, which measures the distortion in electrostatic field when a conductor is brought close to the electrodes. In most configurations, capacitive touchscreens do not measure pressure and do not respond to non-conductive objects. Advanced touchscreen controller chips such as the mXT1188S (Atmel, CA) and FingerTip (STMicroelectronics, Switzerland) are highly optimized to detect finger contact over large areas, reporting only the number of fingers and their locations on

the screen. Many of these chips perform sensor calibration autonomously and are thus able to reject artifacts due to temperature changes or moisture. While it is technically possible for the chips to be reconfigured to also report pressure, it is not the goal of manufacturers to support such a feature.

Optical

There are two general approaches to tactile sensing through optical means. The first approach involves vision, where a digital camera is used to track deformations on a flexible membrane surface [62]. This approach leverages on the maturity of the field of computer vision, utilizing readily available camera systems and algorithms to reconstruct the surface texture in 3D [63]. Although this approach is very sensitive to surface texture, it is not suitable for large scale implementations due to space constraints and the inability to cover curved surfaces.

Another optical based approach relies on photo detectors to measure scattered light intensity as a material deforms [64]. The sensor is made from conformable urethane foam, and is irradiated with light from LEDs. Deformation of the foam results in differences in scattered light intensity, which is picked up by photo detectors underneath the skin. An implementation involving 1864 sensing elements distributed over a humanoid robot was demonstrated [65]. However, sampling rate of such a system is below 20 Hz, which is too slow for manipulation tasks.

Optical based tactile sensors are desirable mainly because they are not affected by electromagnetic interference (EMI). A commercial product based on optical light scattering is available under the KINOTEX brand name [64].

Piezoelectric

Piezoelectric materials generate charge in proportion to the applied force/pressure. Materials such as lead zirconium titanate (PZT), polyvinylidene fluoride (PVDF), etc are especially suitable for dynamic tactile sensing, as charge is generated proportional to the change in pressure. PVDF is also preferred due to its flexibility, workability and chemical stability [66]. However, PVDF based sensors are not suitable for static pressure

measurement, since no charge is produced without changes in pressure. Moreover, PVDF is sensitive to temperature fluctuations, which may be of concern.

Nanowires can also be used to detect normal and shear forces due to their piezoelectric properties [67, 68]. This technique can achieve up to 8464 taxels per cm^2, much higher than the density of mechanoreceptors in the human fingertip. Although the respond rate of 0.15s is much slower than the response time of mechanoreceptors, the authors claim that this could be further improved [68].

Other Notable Technologies

A fluid pressure based tactile sensor has been reported in [69], where a micro-electrical-mechanical system (MEMS) based barometer is placed within a chamber filled with non-compressible fluid. While highly sensitive to vibrations, the approach lacks spatial resolution and is vulnerable to fluid leakage caused by wear and tear.

A tactile sensor array based on MEMS barometers embedded in polyurethane rubber was reported in [70]. Commercially marketed as the TakkTile sensor [71], the sensors leverage on advancements in the miniaturization of digital barometer chips to achieve spatial resolutions of around 8 mm. The product is easy to interface, since each barometer chip is factory calibrated, has its own ADC and communicates over the standard I2C protocol. However, spatial resolution is dependent on the size of the barometer chips, which currently measures 3×5 mm and is unlikely to be comparable to the density of mechanoreceptors.

A stretchable tactile sensor array reported in [72] uses a novel sampling and imaging technique known as electrical impedance tomography (EIT). The sensor was constructed using stretchable conductive fabrics with electrodes placed at the boundaries. Current is injected into the sheet through the electrodes and the impedance distribution is recorded. During contact, the force and distribution of pressure can be recovered through inverse calculations on changes in the distribution of impedance. The advantage of this approach is that no wires are needed within the sensing region, allowing the sensor to easily stretch and wrap around contoured surfaces. However, the approach suffers from high power consumption due to the need for constant current injection.

There are also researchers who make use of tissue engineered skin to embed their sensors. In [73], Cheneler *et. al.* made use of microfluidic channels to deliver nutrients, oxygen and heat to the cells, while local conductivity sensors are used for transduction. Similarly in [74], keratinocytes from rats were cultured on a 4×1 linear sensor array fabricated by MEMS techniques. Flexible bio-hybrid sensor *skins* were reported in [75] as well. All these systems report the ability to sense load and pressure distribution. However, complex microfluidic systems are required to manage nutrient and waste materials, making the system overly complicated for robotic applications.

2.2.3 Mechanical Design

Unlike sensing modalities such as vision and audition, building an e-skin requires the distribution of tactile sensing elements over a broad and non-uniform surface area. The material for constructing tactile sensor arrays thus need to be flexible and sufficiently thin in order conform to the surfaces without impairing the range-of-motion of the robot. Another major impediment in e-skin development is the wiring of the sensors [52]. With increasing number of taxels, more wires are needed to address, power and acquire information from them. The routing of wires through robotic joints is non trivial given the limited available space in the robotic shell.

Flexibility and Compliance

Advancements in fabrication techniques and material science have enabled the construction of sensors which are highly dense and flexible. Extremely thin sensor arrays have been made by MEMS fabrication techniques on polymer substrates such as polydimethylsiloxane (PDMS) and polyimide (PI) [76–78]. Multimodal polymer-MEMS sensors have also been developed, capable of sensing pressure, temperature and conductivity [79]. Recent work by Park *et. al.* involving the fabrication of interlocked microstructures in ferromagnetic films also resulted in highly sensitive multimodal sensors that can detect static and dynamic pressures as well as temperature variations [80].

Besides being flexible, it is also desirable for artificial skins to be stretchable in order to conform to irregular surfaces while being soft and natural to touch. There are two main approaches to producing stretchable conductors. The first involves mounting

circuitry on structures that stretch, such as pre-stretched polymers [76] or nanoscale ribbons. Conductive traces can also be made stretchable by printing them in wavy configurations [81, 82]. Alternatively, stretchable sensors can be achieved using inherently stretchable conductors and substrates. In [83] and [58], mixing carbon nanotubes (SWNTs) with fluorinated copolymer results in a printable elastic conductor material. When fabricated on PDMS, the resultant structure will be inherently stretchable. Printable stretchable conductive inks are also commercially available from DuPont [84].

On the other end of the spectrum, researchers have explored the use of low-cost materials and fabrication techniques for building sensors. Roh *et. al.* reported the use of piezoresistive and conductive fabrics to produce force sensitive surfaces that are robust enough for use as musical controllers [77]. Nassar *et. al.* demonstrated a paper-based artificial skin capable of sensing pressure, proximity, humidity, temperature, pH and flow [85].

The above examples represent a shift in research directions from isolated sensor elements to large arrays of so-called artificial skins. Polymer-MEMS based approach also hold the promise of being extremely lightweight, durable and of sufficient density, while the cost could be reduced with increased production volume and improved manufacturing techniques. Alternatively, the low-cost tactile sensing arrays offer easy access to signal acquisition, spurring the development of tactile processing algorithms.

Wiring

Increasing the density of sensor arrays require more wires to power, address and extract information from individual elements. A single nerve bundle can typically carry over 200 axons per μm^2 while remaining highly flexible and stretchable [86]. Similar densities and mechanical compliance is not achievable with current technology using metallic wires.

The one-wire-per-taxel approach adopted by early researchers [87] is both tedious and not scalable. Instead, most designs today utilize time-multiplexing to reduce the number of wires needed. Among the more popular techniques is the row-column multiplexing, which requires $n + m$ number of wires for an array of $n \times m$ taxels [76, 78, 88]. Although time multiplexing reduces the number of wires needed, the sampling rate of the system is also reduced. For instance using the row-column multiplexing technique, reading the

entire sensor involves the activation of individual rows and columns sequentially. As a result, sampling rate of such systems rarely exceed 200 Hz for hundreds of elements [89], which is slow compared to the 1 kHz feedback loops implemented in most robotic control algorithms [90].

Besides the row-column multiplexing, other sampling strategies have been developed to reduce wire complexity. The EIT technique presented earlier has electrodes placed on the perimeter of the sensor, thus reducing wiring complexity. However, the spatial resolution the EIT approach is low, and requires complex algorithms for reconstruction which also reduces sampling rate. Nilsson [91] proposed to arrange all resistive sensing elements in serial, treating each element as a filter and measuring their impulse response to determine the pressure at each point. While this technique requires only three wires, the number of elements is restricted to about 100 before the signal strength is diminished. It is also possible to wirelessly power and communicate with sensing elements, as demonstrated in [92] and [93]. However, cross-talk due to large number of elements, transmission delays and high power consumption are major challenges cited when implementing such a system.

2.2.4 Interface and Readout

After transduction, electrical signals need to be quantized as digital values. Difficulties in acquiring and processing the large amounts of data read from many tactile sensors has been cited as one reason for slow penetration of tactile sensing in robotics [94]. While time multiplexing of communication channels may increase the number of addressable sensors, it decreases the sampling rate of the system. This section surveys some approaches taken to mitigate the problem.

To increase the number of sensing elements without placing too much burden on the central processing unit, distributed sampling approaches are necessary. In [95], triangular modules, each containing 12 capacitive sensing points, are tessellated to achieve a larger coverage. Each module has its own capacitance to digital converter (CDC) chip to achieve distributed sensing. Coordinated by a central controller, each module can sample the average contact force across all 12 points at a higher sampling rate (500Hz) or individual points at a slower rate (50Hz). However, due to limited

communication bandwidth, only a maximum of 16 modules can be connected at once. Distributed sensing of multi-modal sensor modules is also presented in [96], where hexagonal modules with its own ADC and microcontroller are networked together to cover a larger surface area. While the sampling rate of each module can be very high (>2.5 kHz), extracting output from the entire network is much slower (250 Hz for 49 elements), with the bottleneck being the lack of communication bandwidth.

System responsiveness can be improved by performing data compression at the local (module) level. In [89], each module contains a field programmable gate array (FPGA) to compress tactile data into an expression that represents only the centroid, intensity and spread of the stimulus, enabling a 200 Hz sampling rate for 640 taxels. Fukui *et. al.* adopts an evolutionary algorithm to intelligently sample only a subset of the array, potentially reducing the response time of the system [97]. A more extreme case was presented in [90], where only the centroid and magnitude of the force is extracted to achieve a 1 kHz sampling rate for any size of sensor array.

From the above examples, it is apparent that spatial resolution is often sacrificed to improve system responsiveness. However, it is questionable if such a trade-off is optimal. A more prudent approach would be to determine the most important features needed for tactile perception before deciding on an efficient compression scheme. In biological systems (See Section 2.1.4), intensity signals using rate coding alone would be too slow to support our reflex and manipulation abilities. However, the spiking neural representation of signals in the somatosensory cortex allows multiplexing of stimulus information across multiple time scales [42]. This suggests that a more efficient representation of tactile signals may hold key to solving this problem.

2.2.5 Feature Extraction and Perception

Having obtained a digital representation of tactile signals sensed from multiple locations, computational techniques are required to extract features and address complex tasks such as grasp stability. Applications such as object recognition may also require the fusion of sensory inputs from multiple modalities (e.g. tactile, vision, kinesthetic). This section highlights some approaches taken to address the above issues.

A variety of machine learning techniques have been utilized for feature extraction in tactile applications. A comparison between support-vector machine (SVM), regularized least square (RLS) and regularized extreme learning machine (ELM) for classification of materials from raw sensor data was reported in [98]. Neural network algorithms have also been used for recognition of local curvatures [99, 100]. Other reported algorithms include self-organizing maps (SOMs) and Bayes decision trees [101, 102]. Researchers have also worked with signals in the frequency domain, particularly for detecting micro-slips that indicate the occurrence of slippage [103].

Recently researchers have adapted spike based processing techniques commonly used in computational neuroscience for use with tactile data. In [104], pressure intensity signals were converted to voltage pulses mimicking the SA-1 mechanoreceptors. Analysis of the variation in firing rate was used to discriminate between different surface gratings. The work was further extended to include the discrimination of naturalistic textures in [105]. Similarly in [106], simulated spiking output from tactile sensors enabled the recognition of braille characters.

Besides tactile recognition tasks, algorithms were also developed to resolve the complex geometric arrangement and changing positions of tactile sensing elements. In [107], correlation based statistics were used to embed the sensor elements in a 3-dimensional space. However, this requires numerous trials and additional constraints to resolve ambiguities. The authors of [108] utilize accelerometers and at each sensor patch and graph algorithms to calculate the relative position between sensors. This second approach is much faster, but still requires substantial computational resources.

2.2.6 Discussion

Despite the many innovative designs, most tactile sensor arrays remain purely experimental. Much emphasis has been placed on making sensors instead of a sensing system as a whole. While it may still be interesting to investigate transducer technology for specialized applications, the focus is shifting towards technologies for large scale implementation, such as sensor fabrication, readout circuitry and interpretation algorithms. However, due to constraints in wiring complexity, slow readout speed and high costs, the adoption of these arrays have not been widespread.

The complexity of tactile signals are expected to rise steeply with increasing coverage of the sensors. For instance, artifacts will arise due to changes in posture, and sensor characteristics may change over time due to wear and tear. Sophisticated algorithms would be needed to fuse proprioception, sensation and memory in real-time to increase reliability of perception. This poses a huge challenge to current technology.

Biological systems have evolved to solve the problem in a very elegant and energy efficient manner. It would be prudent to seek inspiration from nature and investigate the possibility of replicating some mechanisms involved. The next section provides a brief overview on attempts at replicating biological mechanisms in artificial systems for sensation and perception.

2.3 Neuromorphic Concepts for Sensory Systems

2.3.1 Overview

There are numerous examples in nature where compact, adaptable and intelligent systems are also highly energy efficient. For instance, a bee with body weight less than a gram and with a brain consuming as little as 10μW is capable of exquisite flight control and cognitive behaviors. Many researchers have taken to extensive simulations of brain networks using supercomputers in hopes of identifying the features contributing to nature's computational efficiency. While the high precision, high powered digital hardware may do well in detailed biophysical simulations, they are not well suited for emulating the slow asynchronous processes that constitutes biological neural computation.

Researchers in the field of neuromorphic engineering take a different approach, using a mix of analogue, digital and software systems to implement the neuronal organization and function of nervous systems. Compared to conventional microprocessors, neuromorphic systems try to incorporate biologically inspired features such as massively parallel processes [109], distributed memory arrangements [14], asynchronous communication protocols [110] and even built-in learning rules [111].

One of the key distinguishing features of neuromorphic systems is the encoding of information as events (spikes). Analogous to action potentials in biology, spikes are typically asynchronous events that are all-or-nothing. This raises the important issue of spike-based representations of information. This section begins with an overview of the various spike codes observed in biology, followed by artificial algorithms that have been developed to process such representations. Subsequently, examples of neuromorphic systems and hardware are briefly presented.

2.3.2 Spike codes

Rate code

For decades, scientists have assumed that information is encoded by the firing rate of the neuron. This was largely influenced by early physiological recordings in the 1920s, where electrical activity in sensory fibers exhibited increased firing rate with increasing

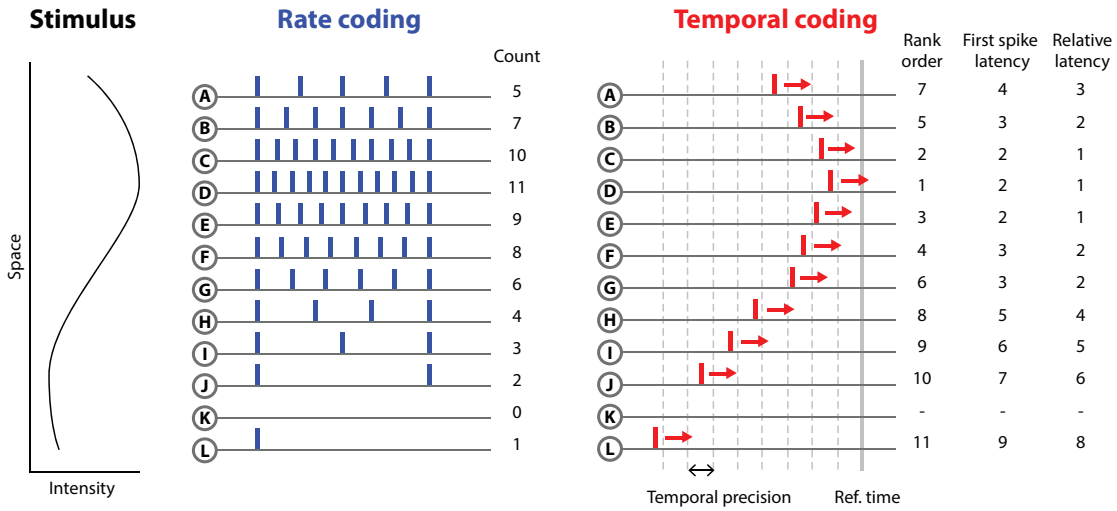


Figure 2.7: Illustration comparing rate code to three distinct temporal codes. Rate coding requires multiple spikes per neuron to encode the stimulus profile, while temporal codes require only one spike per neuron. The rank code is based on the order in which the spikes arrive. First spike latency code looks at the time of arrival of spikes in reference to an external synchronization signal, while relative latency code uses the time of first spike as the reference. Both first spike and relative latency codes assume that the decoding neuron is time sensitive.

stimulus intensity. Indeed, artificial neural network and connectionist models usually involve a population of processing units that are connected together by variable weight connections that represent synapses. Each input is a floating point number, analogous to the firing rate of the pre-synaptic neuron. The output of the unit is also a floating point number, obtained as a non-linear function of the weighted sum of all inputs, often bounded by a range (usually 0 and 1). Even in neuroscience, neural recordings are often summarized in the form of post-stimulus-time-histograms (PSTH) that plots firing rate as a function of time.

Despite the popularity of rate coding, it could not explain the impressively fast response times of the primate brain to complex stimuli. Experiments have shown that subjects can recognize faces or 3D objects within 100-150 ms after stimulus onset [112], while responses to contact events occur in less than 100 ms [24]. Given the multiple layers of neurons that signals have to pass through, and the fact that cortical neurons rarely fire above 100 Hz, individual neurons would only have time to fire at most one spike. This would not be possible if the brain relies on rate coding exclusively, since at least two spikes would be required per neuron. Alternative coding schemes capable of

rapid processing should exist in the brain.

First spike latency code

First spike latency code relies on the time differences between a reference signal and the subsequent spikes received from a population of neurons. Such a coding scheme has been reportedly observed in the auditory, visual, olfactory and somatosensory systems [38, 113–116]. In theory, the first spike latency code requires only a single spike from each neuron, thus allowing signals to be propagated much faster compared to the rate code (Fig. 2.7).

The temporal differences could arise due to several factors. Firstly, the stimuli could inherently carry temporal variations. For instance, during sound localization, a sound pulse from the left of the subject will reach the left ear before the right ear [117]. Objects moving in a certain direction in the visual field would stimulate the receptors in a particular sequence [113]. Upon contact with an object, the order of skin deformation also carries rich information on the curvature of the surface [38]. Indeed, many sensory systems exploit the time-of-arrival differences at different receptors for processing.

Another source of temporal differences is from the receptors. Due to their integrate-and-fire nature, they are particularly suited to perform intensity to latency conversion, since a strong stimulus would cause the membrane voltage to reach threshold earlier.

The first spike latency code assumes that the subject has access to a reliable reference signal. In most laboratory situations, researchers define the stimulus onset as the reference. However, the onset time may not be explicitly available to the subject. Motor signals of retinal saccades or hand movement may be used as reference times for visual and tactile inputs respectively, but it is less clear in situations where the subject has no control of when the stimulus occurs.

Relative latency code

A more robust alternative to first spike latency coding is to use relative spike times. This coding scheme assumes that receptors fire in concert among a subset of neurons within a reasonably short delay, and information is encoded in the relative time differences of the spikes (Fig. 2.7). As such, no absolute reference signal is required. This is a well

studied strategy in auditory cortex, and a possible decoding scheme would utilize delay lines leading to a coincidence detector [117].

However, it has been shown that "first spikes" are significantly more reliable temporally than the subsequent ones. Hence, it is advantageous to have a method of resetting the system especially during a continuous train of stimulation. Researchers have proposed the use of local field potential oscillations to segment the stimuli into fixed temporal windows. Alternatively, active exploratory behavior such as microsaccades may play a similar role, discretizing continuous streams of information into samples.

The information capacity of both the first spike latency and relative latency codes rely largely on the temporal resolution of downstream neurons to resolve the differences in spike timings. In Fig. 2.7, spikes from neurons C, D and E are thought to occur simultaneously as a consequence of insufficient temporal resolution.

Rank order code

Instead of looking at the precise timing of spikes from each channel, it is also possible to consider the order in which the neurons fire [118]. From Fig. 2.7, the order could be thought of as D-C-E-F-B-G-A-H-I-J-L. For a population of N neurons, the rank order code can produce $N!$ different possibilities. Similar to latency codes, a reference signal is also needed to indicate which is the "first" spike in the sequence. A combination of rank-order and latency codes have been demonstrated to provide rapid and reliable information from retinal ganglion cells [119].

From this brief comparison, it is hopefully obvious that temporal codes have a clear advantage over rate codes in the capacity of information they can encode. However, the spike trains are usually treated as point processes due to their stochastic behaviour, thus increasing the complexity in signal processing. The next section discussion techniques for learning and decoding spike patterns.

2.3.3 Spiking Neural Networks and Learning Algorithms

The latest generation of neural networks incorporate temporal features for computation. These networks are known as spiking neural networks (SNN), where networks of neurons and their rich dynamic behavior are simulated artificially [13]. SNNs have been shown

to be capable of solving complex tasks such as recognition of visual patterns, odors, sound properties and tactile stimuli, which are not easily achievable by rate-based models [106,120–123]. SNNs typically require less neurons compared to Artificial Neural Networks (ANN) [124]. Due to their biological plausibility, SNNs are useful for modeling information processes in the brain and investigating various learning mechanisms [125–128]. SNNs have also proven useful in performing pattern recognition and real-time computation for a variety of engineering problems [129–134].

Similar to artificial neural networks (ANN), SNNs involve populations of neurons that are connected via synapses, where each neuron typically receives inputs from multiple neurons through synapses of adjustable weights. Training of SNNs typically involves selecting a set of weights such that the spiking neuron performs the desired decoding function. Training can be supervised, unsupervised or reinforced. In supervised training algorithms, the desired output of each neuron is utilized to compute the required weights, while in unsupervised scenarios, a general learning rule is applied without an explicit goal definition. Reinforced learning uses a global reward (or penalty) signal as feedback to adjust the network parameters.

Unsupervised Learning

Unsupervised learning rules in SNNs are deeply rooted in biologically observable mechanisms responsible for synaptic plasticity. Of the many forms of synaptic plasticity discovered, the most relevant (and popular) form is known as Hebbian learning, where it has been postulated that:

> *When an axon of cell A is near enough to excite cell B or repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased [135].*

In SNNs, Hebbian learning has been adapted to take timing information into account. This has led to models of spike-timing-dependent-plasticity (STDP) [136]. STDP is a physiologically observable mechanism where synaptic weights are adjusted by the time difference between the pre-synaptic and post-synaptic spikes. Several STDP learning

rules have been observed [137]. Typically, long-term potentiation occurs when the pre-synaptic spike leads the post-synaptic spike, while long-term depression occurs when post-synaptic spike leads the pre-synaptic spike. Some algorithms utilize delays instead of weight adjustments [138, 139], or the width of input kernels [140], primarily to ease implementation in digital hardware.

When suitably modeled in SNNs, STDP enabled neurons have exhibited interesting properties. In [141], neurons with STDP could automatically adjust their weights to detect the start of repeating spatiotemporal patterns in continuous spike trains. On a similar note, the authors in [126] demonstrated how STDP based rules could enable neurons to generate internal models of input data. More remarkably, when organized in multiple layers, neurons with STDP have been shown to autonomously extract complex visual features [142]. All these highlight the effectiveness of STDP as a form of self-organization in SNNs for efficient representation of input stimuli.

Reinforced Learning

Reinforced learning is a robust way of training networks. Guided by only a global reward or penalty signal, neurons in the network update their internal parameters based on general rules. One such implementation is through reward-modulated STDP [127, 143, 144].

In reward-modulated STDP, synaptic eligibility traces are used to mark synapses with high causality. However, these synapses are not modified immediately, but only when reward signals are received. The eligibility traces decay over time, thus ensuring that only synapses that are consistently involved with the reward are enhanced the most.

Supervised Learning

Although less biologically plausible, supervised learning is very useful in training SNNs for specific goals such as pattern recognition or stimulus encoding. In [145], the authors derived methods to train a neuron to fire or not fire when presented with a particular input pattern. The algorithm has been used to classify both artificial stimuli and biological spikes recorded from the retina [123, 146].

For SNNs with multiple layers it is often useful for neurons to learn to map specific spatiotemporal input spike patterns to temporally precise output spikes. This capability permits the output spikes to be subsequently decoded by higher order neurons sensitive to spike timing [147]. Learning rules such as the ReSuMe [148], Chronotron [149], SPAN [150] and PSD [151] train neurons based on heuristic mechanisms derived from the Widrow-Hoff rule [135]. A mathematical implementation of Chronotron (E-learning) implements a gradient descent approach where weights are adjusted to minimize an error function defined by a modified Victor Purpura distance metric [152].

Although heuristic approaches may be simpler to implement and are arguably more biologically plausible, training the algorithm tends to be difficult and tedious as they do not guarantee a globally optimal solution because of the non-convex nature of the minimization criteria. There is hence no guarantee that a solution could be found even if it exists, as the solutions are susceptible to being trapped at local mimimums.

2.3.4 Implementation of Neuromorphic Systems

Like most analog and digital computing chips today, neuromorphic devices are built from silicon transistors. Researchers exploit the fact that when operating in the subthreshold region, transistors exhibit many functional similarities with neurons. In particular, the transistor current is exponentially dependent on its terminal voltages, much like how the membrane potential of a neuron is exponentially related to the active populations of voltage sensitive ionic channels [153]. Utilizing this property, electronic models of conductance based neurons and synapses can be implemented in silicon to achieve biologically plausible computation primitives such as logarithmic functions, inhibition, thresholding and winner-take-all selection [154].

Communication

While it is possible to create artificial neurons using transistors, wiring them together to form a network is non-trivial. The brain represents a huge three dimensional network of interconnected neurons, with connection strengths that may vary over time as the brain undergoes learning and adaptation. Constructing a network of comparable complexity in silicon will be a great challenge since transistors are typically patterned on a 2D

substrate with only a few layers of wiring. However, electrical signals propagate 10^7 times faster in silicon compared to biology. By multiplexing signals in time, engineers are able to devise novel communication schemes to overcome the wiring limitations.

The most popular communication scheme in neuromorphic systems is called the address-event representation (AER) [110]. Unlike in biology where neurons are physically connected, AER employs time-multiplexed encoding of spiking data from several groups of neurons into a single communication bus enabling efficient communications. Transceivers encode and decode spikes over a small set of high-speed wires by encoding each axon with a unique binary representation called an address-event. To save hardware real estate, neurons are grouped together to share a common encoder and decoder. The spike packets are delivered by routers, where each packet is sequentially decoded, searched through a lookup table, and forwarded to the next node. Arbiters are employed to resolve conflicts that may arise if multiple packets arrive at a router simultaneously. Variants of AER have been employed in neuromorphic systems such as SpiNNaker [155], CAVIAR [132] and others [14].

Since the communication bus is shared, AER is susceptible to routing congestion if certain groups of neurons fire at an exceedingly high rate [156]. Secondly, the lookup table for each node may require substantial memory to implement. Finally, deadlocks and livelocks may occur when routing packets, which may result in dropped packets or packets delivered with timing errors [110].

An alternative to AER is synaptic time multiplexing (STM), where a complex connectivity map is established by time-multiplexing of multiple simpler networks [157]. Advantages of STM include reduced memory requirements due to the lower number of synapses in the simple networks, as well as immunity to neurons with high firing rate as all synapses have an assigned time-slot to propagate spikes. However, STM requires the routing fabric to be constantly switched based on a global multiplexing clock, which may consume additional power [157].

Sensors

Vision, audition and olfaction are currently the main sensory domains where neuromorphic sensors have been developed. These sensors represent novel methodologies to

achieve low power and sparse capture of information by emulating biological sensory systems.

Silicon retinas are the most prominent examples of neuromorphic sensing. Conventional vision sensors adopt a frame-based approach where photo-receptors are constantly scanned at a fixed rate regardless of whether the scene changes, thus generating significant volumes of redundant data. Conversely, the silicon retina has pixels that operate individually and report only when significant deviations on temporal contrasts are detected. The spiking output is similar to action potentials generated by retinal ganglion cells. Most silicon retinas also adopt the AER protocol to achieve asynchronous communication. Neuromorphic vision sensors today include the 128×128 pixel dynamic vision sensor (DVS) [158], the 304×240 pixel asynchronous time-based image sensor (ATIS) [159] which also adopts a PWM based reporting of pixel intensity, as well as the 240×180 pixel dynamic and active pixel vision sensor (DAVIS) [160] which supports both asynchronous and active-pixel sensing.

Similar to silicon retinas, silicon cochleas model the inner hair cell circuit with a cascade of filter banks that continuously respond to sounds at distinct frequency ranges. This is in contrast to conventional audio sensors, where audio signals are sampled at an application specific Nyquist frequency [161]. Silicon cochleas also output sparse spike trains using the AER protocol, which has been shown to achieve 40 times higher computational efficiency for source localization when compared to conventional cross-correlation techniques [162]. The latest version of neuromorphic auditory sensors is the AEREAR2 which features binaural microphones and 64 channel filter banks [162].

Unlike their vision and auditory counterparts, researchers in neuromorphic olfaction focus on implementing critical olfaction characteristics rather than emulating the biological olfaction pathway. In [163], researchers utilize the firing delay in the spiking output from an array of tin oxide gas sensors as a unique representation that is more robust to sensor drift. Tang et al. [164] implemented spike-timing-dependent plasticity learning rule in a VLSI chip to perform online classification of gases from a commercial gas sensor. The NEUROCHEM project [165] mimics the essential characteristics of biological odor receptor neurons by incorporating a large array of chemical sensors with overlapping specificities and redundancy, although standard techniques are used for recognition of

odors from the output of the sensors.

Computation

In a neuromorphic sensor system, events generated by neuromorphic sensors are propagated in real-time to electronic chips that model networks of neurons with varying complexity and biological plausibility. On the one end, the neurons, synapses and ion channels may be modeled using sub-threshold analog characteristics of transistors [166, 167]. Analog neuron circuits are typically more energy efficient and compact, as the non-linear computational properties of synapses are intrinsically reflected by the characteristics of the transistors. The network also operates in continuous time regardless of size as all computations are performed in the analog domain. The downside to an analog approach is the lack of flexibility of the model, since most parameters will have to be fixed during fabrication. Moreover, the neuron properties are less deterministic and may be affected by environmental conditions such as changes in temperature.

Alternatively, neurons and synapses can be modeled digitally using custom built digital processors that have highly parallel architectures. Projects such as the SpiNNaker [155] and TrueNorth [14] are examples of this latter approach. The SpiNNaker features an array of ARM cores each simulating 1000 spiking neurons networked by AER routers, while each TrueNorth chip contains 4096 custom digital cores each having 256 spiking neurons and 65k synapses. Digital models are comparatively more flexible to analog implementations, since the computational functions can still be changed after the chip is made. Tasks such as visual and auditory pattern recognition have been implemented in various neuromorphic processing hardware to achieve lower latency and power consumption [132, 168].

2.3.5 Discussion

Engineers have often been inspired by the robustness and efficiency of biological systems. While neuromorphic sensors have demonstrated advantages in achieving low-power consumption and sparse output data generation, techniques to process spike-based data are still limited. Moreover, a unified method for integrating signals from

neuromorphic sensors of multiple modalities does not exist. In comparison, advanced computer algorithms and datasets are readily available for conventional computer vision and audio applications. The adoption of neuromorphic sensors will largely be dependent on the maturity of spike-based signal processing techniques.

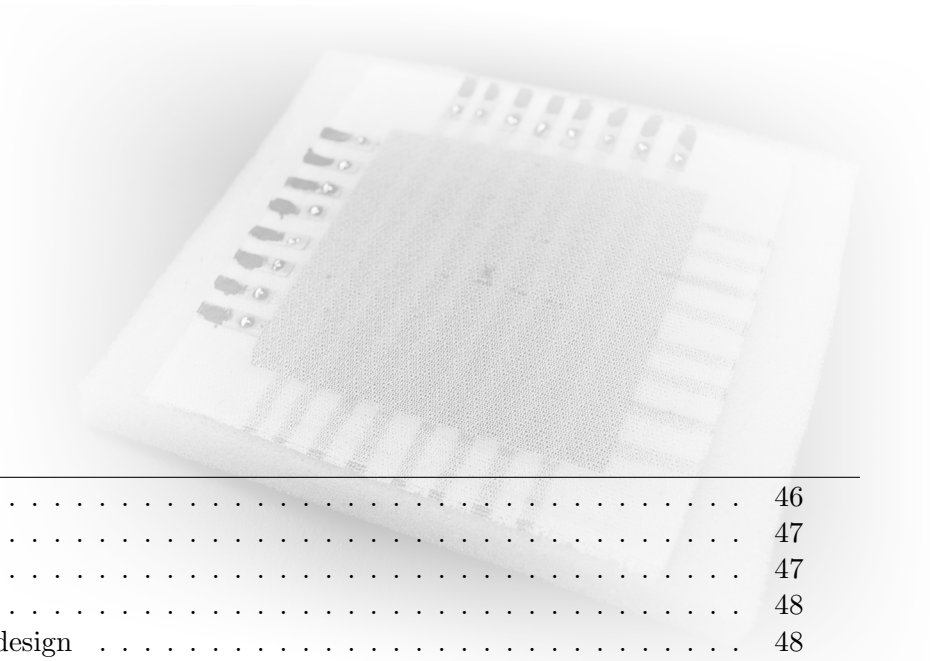
Unlike vision and audition, there appears to be considerable scope for improving tactile sensing through neuromorphic techniques. Firstly, the biological processes for tactile sensing are well understood, providing a solid foundation to guide the design of neuromorphic tactile systems. Secondly, techniques for conventional tactile signal processing are less mature compared to computer vision and audition, thus the introduction of novel signal acquisition and processing concepts would face less resistance. More importantly, large scale tactile sensing would benefit greatly from low-power and sparse signal acquisition, given the limited readout bandwidth of tactile sensors due to challenging wiring requirements.

The rest of the thesis will explore how mechanisms observed in biological tactile sensing can be used to improve robotic tactile systems, in particular to achieve efficient integration of many sensor elements covering a large surface area.

3

Towards a Spatiotemporal Representation of Touch

Lee, W. W., Cabibihan, J. J., & Thakor, N. V. (2013, November). Bio-mimetic strategies for tactile sensing. In *SENSORS, 2013 IEEE* (pp. 1-4). IEEE.



Contents

3.1 Preface

Inspired by the temporal coding of tactile signals observed in biology, our first step is to investigate if a similar effect is achievable in an artificial system. Starting from scratch, this requires us to construct a tactile sensor array and derive a convincing model of pressure to spike conversion. There is also a need to develop a novel method to evaluate the efficacy of temporal codes, since conventional mathematical operations do not apply to spatiotemporal representations due to their sparse and discontinuous nature. This chapter presents our early attempts at overcoming the challenges listed above.

We built several iterations of tactile sensor arrays before settling on a fabric based design. A numerical neuron model was used to convert the output of the tactile sensor array into spatiotemporal spiking patterns that mimic mechanoreceptors in the skin. We programmed a robot to perform physical experiments in a controlled manner, and utilized a supervised spike-timing based learning algorithm to distinguish spatiotemporal spike patterns corresponding to different local curvature of objects indented onto the sensor by the robot. The high classification accuracy (>99%) and fast convergence rate (tens of epochs) of the classifier indicates good separation between different stimuli using spatiotemporal spike representation.

Promising results from this study led to the subsequent development of the Kilohertz Kilotaxel sensor array (Chapter 4) and a detailed investigation into the temporal coding for tactile sensors (Chapter 5). The approach towards data-acquisition and spike conversion presented in this chapter differs greatly from subsequent works. However, the underlying concept of extracting features from the spatiotemporal dynamics during sensor deformation remains consistent with the rest of the thesis.

Author Contributions and Acknowledgements

Lee W. W. designed and developed the experimental prototype, conducted the experiments, analyzed the data and generated the figures and wrote the manuscript. Dr. Thakor N.V. provided guidance on conceptualizing the idea. Dr. Cabibihan J.J. and Dr. Thakor N.V. proofread the manuscript. The authors would like to thank the Institute for Infocomm Research (I2R), Singapore, for the use of the robotic arm in experiments.

3.2 Introduction

Tactile sensing is defined as the detection and measurement of contact parameters in a predetermined contact area, and the subsequent preprocessing of the signals [1]. It is crucial to the understanding of interaction properties of real-world objects, such as its weight, stiffness, surface texture and deformability. Much effort has been devoted to developing tactile sensors for dexterous manipulation in robotics and prosthetics [169], where sensors with increasingly high spatial resolution and flexibility [68] are under continuing development. However, it is increasingly difficult to maintain the temporal resolution as sensors continue to scale up, due to limited bandwidth for sampling and computation. In addition, high manufacturing precision is required to ensure consistency in the response of a large sensor array, while extensive calibration would also be required to ensure its accuracy. All these add to the cost of the sensor.

This chapter presents a novel, bio-inspired approach to improve the scalability of tactile sensor arrays. Inspired by the spiking responses of human mechanoreceptors [24], we postulate that sampling bandwidth and computational resources may be more efficiently utilized through event-based representations. To test this hypothesis, we developed a flexible, stretchable and conformable tactile sensor array using conductive fabric. Sensory output from the array is converted into spatiotemporal spike patterns which are fed to a spiking neural classifier. Finally, we conducted experiments to distinguish between indentors of different sizes using only the spike patterns.

The organization of the chapter is as follows. Section 3.3 provides some background on the mechanoreceptor characteristics and the source of inspiration for this work. Section 3.4 presents the tactile sensing system developed. Section 3.5 describes the experimental design, while Section 3.6 discusses the results. Finally, Section 3.7 concludes the chapter.

3.3 Background

Tactile sensing in humans is achieved by four main categories of mechanoreceptors — Meissner's Corpuscle (FA-I), Merkel Cells (SA-I), Pacinian Corpuscle (FA-II) and Ruffini Corpuscle (SA-II). Fast-adapting (FA) receptors respond only during stimulus onset and

offset, while slow-adapting (SA) receptors are sensitive to sustained contact stimulus [24]. During contact, the surface of the object is projected to these tactile afferents as the skin deforms around it. Each mechanoreceptor represents a small portion of the object and encodes the information through spikes of action potentials - voltage pulses generated when the stimulus is greater than a threshold.

Research has shown that during natural object manipulations, tactile information from the fingertips is used faster than can be readily explained by rate codes [24]. A recent study suggests that the latencies of first spikes from a population of mechanoreceptors can be used to encode important information such as force magnitude, direction and local curvature [24]. This temporal encoding scheme requires only a single spike from each involved afferent, and can thus relay information much faster.

Considering the fact that only a single spike is required per mechanoreceptor, each afferent can be perceived as a binary contact switch that fires a spike when the experienced pressure exceeds a single threshold. The spatiotemporal pattern of emitted spikes would depend on both the sequence by which the mechanoreceptors are recruited and the threshold of the afferent. Due to the high density of afferents and the large degree of receptive field overlap, a single stimulus indentation would activate a large group of mechanoreceptors in a sequence that conveys relevant information about the stimulus. Theoretically, the same technique should also be applicable to artificial tactile sensors, which is the focus of this work.

3.4 Methods

3.4.1 Sensor design

For this work we have created an 8×8 taxel sensor array using fabric based materials. Fig. 3.1a illustrates the construction of the sensor. The transducer material consists of a piezoresistive fabric (LTT-SLPA, Eeonyx). Eight rows and columns of conductive fabric (Silver plated mesh, LessEMF) are arranged above and below the piezoresistive layer respectively. Fusible stretch interfacing made from cotton (G785, Vilene) was fused onto the top and bottom of the sensor using an iron, thus holding the circuit together. The sensing area measures 6×6 cm.

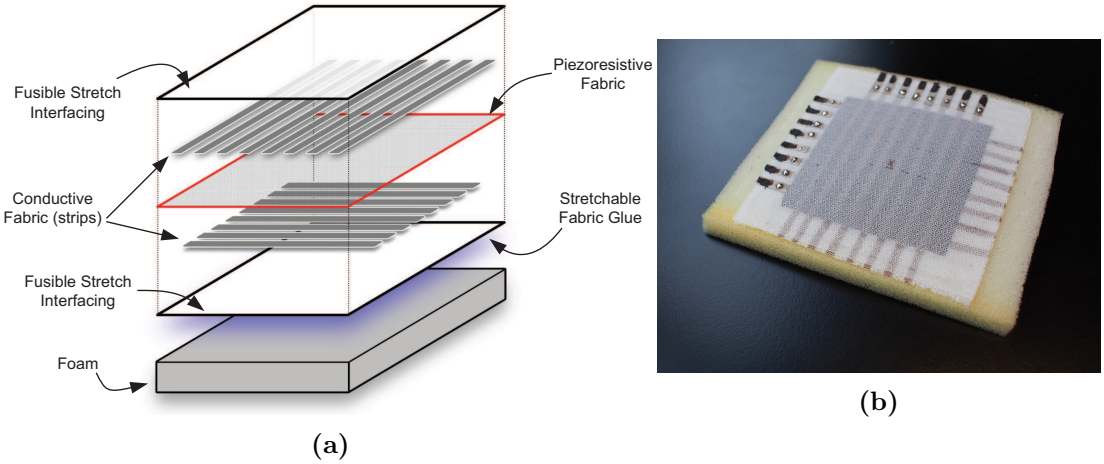


Figure 3.1: (a) The architecture of the fabric sensor. (b) A photograph of the sensor mounted on foam. © 2013 IEEE.

Our sensor differs from conventional designs in that it is placed on a soft backing made of multi-purpose, polyurethane foam (Tontine Fillings, Australia). This novel configuration allows the sensor to conform to objects during contact, thus triggering a larger population of sensing elements Fig. 3.1b.

3.4.2 Data acquisition

Fig. 3.2 illustrates the readout circuitry. As the resistance of the piezoresistive material decreases with increasing applied pressure, each taxel can be modeled as a variable resistor (R_s). To perform a raster scan of the array, the row to be addressed is excited with a logical high (5 V), while all other rows are set to a logical low (0 V). A potential divider circuit is then used to obtain a voltage readout that corresponds to the resistance at the particular cross section of the sensor.

A data acquisition board (USB-6356, National Instruments) was used to provide excitation signals for the rows and perform analog-digital conversion (ADC) on the columns. The sampling rate was set to 10 kHz.

We applied a binary threshold to the readout to simulate a digital output. Fabrication inconsistencies in the sensor meant that the actual threshold of individual elements will vary. This is analogous to natural mechanoreceptors, where the individual thresholds are not consistent [23].

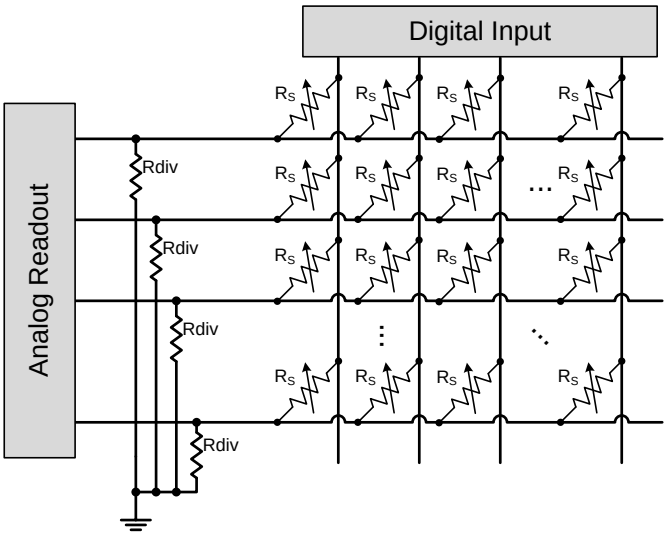


Figure 3.2: Sensor readout circuitry. © 2013 IEEE.

3.4.3 Spike conversion

We convert the digital output to spikes by applying two Izhikevich neurons models [170] per channel, with one neuron receiving the digital signal as input while the other receiving the inverse of the signal. The outputs were then combined to form the spike response. This ensured that the system was sensitive to transitions from low to high, as well as high to low, as expected of a FA-I mechanoreceptor. The parameters for both the Izhikevich neuron models were: a=0.02, b=0.25, c=-65, d=6, gain=0.5. These parameters used correspond to the *phasic spiking* mode of the model, where a single spike is fired for each change in input from low to high. Fig. 3.3 illustrates the conversion from analog input to spiking output.

3.5 Experiment: Distinguishing between different local curvatures

Local curvatures play a useful role in identifying features during manipulation. As discovered in [24], first spike latencies from mechanoreceptors can encode local curvature information. In this experiment, we investigate the discrimination of local curvatures using only first spike latencies from the biomimetic sensor.

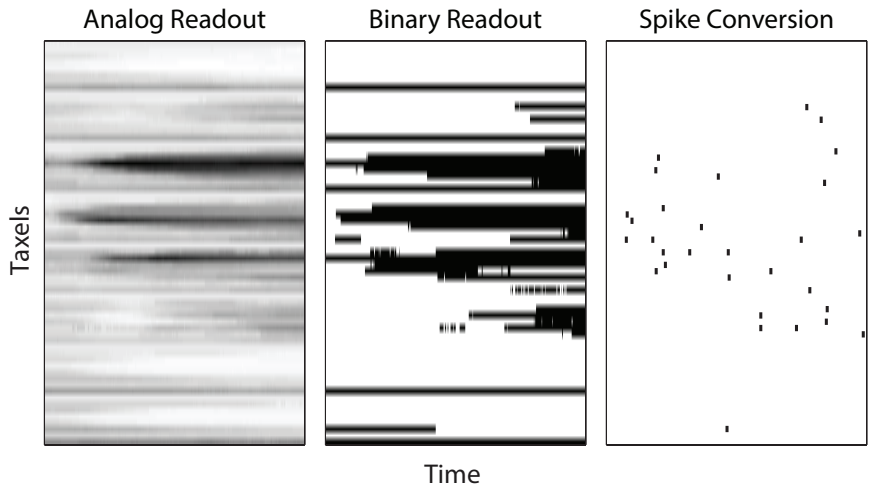


Figure 3.3: Signals conversion from analog readout to spikes. Leftmost column indicates analog signals measured in volts, with higher (darker) values indicating higher pressure. The middle column are signals after thresholding, and rightmost column presents the spiking data. © 2013 IEEE.

3.5.1 Stimulus delivery

We programmed a 6-axis robotic arm (Jaco, Kinova Robotics) to deliver the stimulus. Glass spherical indentors of diameter 65 and 105mm were used (CTech Glassware). Two indentation forces were applied per indentor — 1N and 4N, as measured using a load cell placed below the sensor (accuracy 0.01N). We used displacement control instead of force control, due to the non-linear response of the foam. The displacements were determined as the indentation depth that corresponds to the desired force readout after the reading has stabilized. Fig. 3.4 illustrates the experimental setup.

We began recording signals just before the indentation, and stopping before the indentor was lifted. A 5 second pause was inserted in between indentations to allow the foam to recover. A total of 200 recordings were made per combination of indentor and force.

3.5.2 Spike pattern classification

The recorded spike patterns were classified using the Tempotron classifier [145]. The learning rule is based on the biologically observable Spike Time Dependent Plasticity (STDP) mechanism, and has been used for classifying a several spatiotemporal patterns.

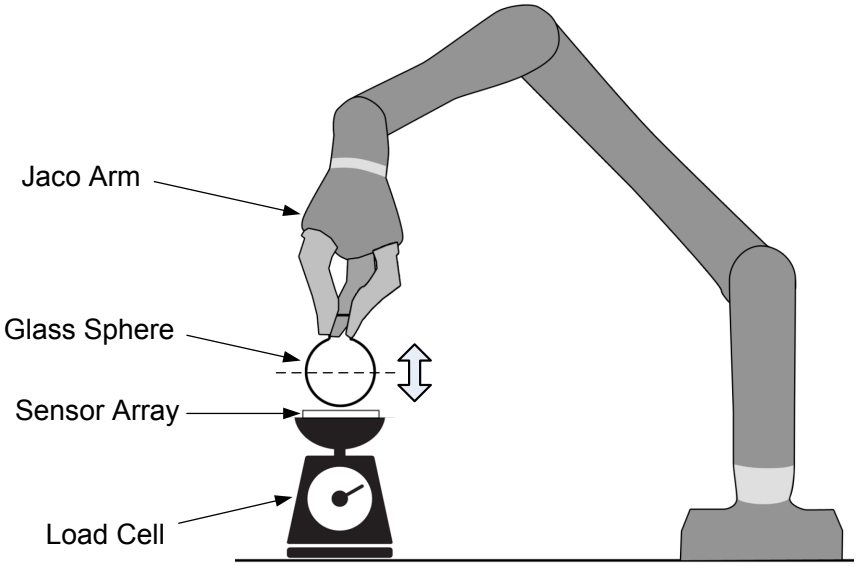


Figure 3.4: Experimental setup. The glass sphere was held by the gripper arm and pressed onto the sensor array at a fixed indentation depth corresponding to 1N or 4N forces. © 2013 IEEE.

The classifier consists of a leaky integrate-and-fire (LIF) neuron, where the weights of input synapses are adjusted to emit a single spike if the input pattern corresponds to the preferred stimulus, and remain silent otherwise. The sub-threshold membrane voltage $V(t)$ of the neuron is the weighted sum of the postsynaptic potentials (PSPs) from all incoming spikes:

$$V(t) = \sum_i \omega_i \sum_{t_i} K(t - t_i) + V_{rest} \tag{3.1}$$

where ω_i and t_i are the synaptic weight and firing time of the i^{th} synapse respectively. V_{rest} is the resting potential of the neuron. K denotes the normalized PSP kernel:

$$K(t - t_i) = K_0(\exp(\frac{-(t - t_i)}{\tau_m}) - \exp(\frac{-(t - t_i)}{\tau_s})) \tag{3.2}$$

where τ_m and τ_s are decay time constants, and K_0 is a constant that normalizes the maximum value of the kernel to 1. When $V(t)$ crosses the threshold from below, all subsequent input spikes are ignored.

The Tempotron learning rule relies on a supervisory signal to indicate whether the

input spike patterns correspond to the target category (C^+) or not (C^-). Ultimately, the neuron should emit a spike for a category C^+ input, and remain silent for a category C^- input. According to the rule, an incremental adjustment $\Delta\omega$ is performed on the synaptic weights whenever an error is made:

$$\Delta\omega_i = \begin{cases} \lambda \sum_{t_i^j < t_{max}} K(t_{max} - t_i^j) & \text{if no spike emitted for } C^+ \\ -\lambda \sum_{t_i^j < t_{max}} K(t_{max} - t_i^j) & \text{if spike emitted for } C^- \end{cases} \tag{3.3}$$

where $\lambda > 0$ is the learning rate of the algorithm.

For each indentation force, patterns created by the 105 mm indentor were assigned to target class C^+ while recordings from the 65 mm indentor were designated as class C^-. The total number of trials were randomly split in half to obtain the training and testing datasets, and the results were averaged across 10 such tests with random initial variables and splits. Training stops after 200 epochs, or when the algorithm has converged. Parameters used for the classifier were: $\tau_m = 25\,\text{ms}$, $\tau_s = 6.25\,\text{ms}$, $V_{rest} = 0\,\text{mV}$ and $\lambda = 0.0001$.

3.6 Results and Discussion

3.6.1 Results

The results of the classification are presented in Fig. 3.5. On average, the classification accuracy of indentations at 1 N was 99.65%, with standard deviation of 0.41%. Accuracy of indentations at 4N was 99.1% with standard deviation of 1.41%. From Fig. 3.6 it is apparent that training error was reduced to below 5% within tens of iterations for both 1 N and 4 N cases. This shows that good inter-class separability is achievable using first spike latency information alone.

3.6.2 Discussion

A major impediment to the scaling up of tactile sensing arrays is the limited sampling rate due to the lack of readout bandwidth. Our neuromorphic approach mitigates this constraint by using a binary tactile array, resulting in greatly reduced output

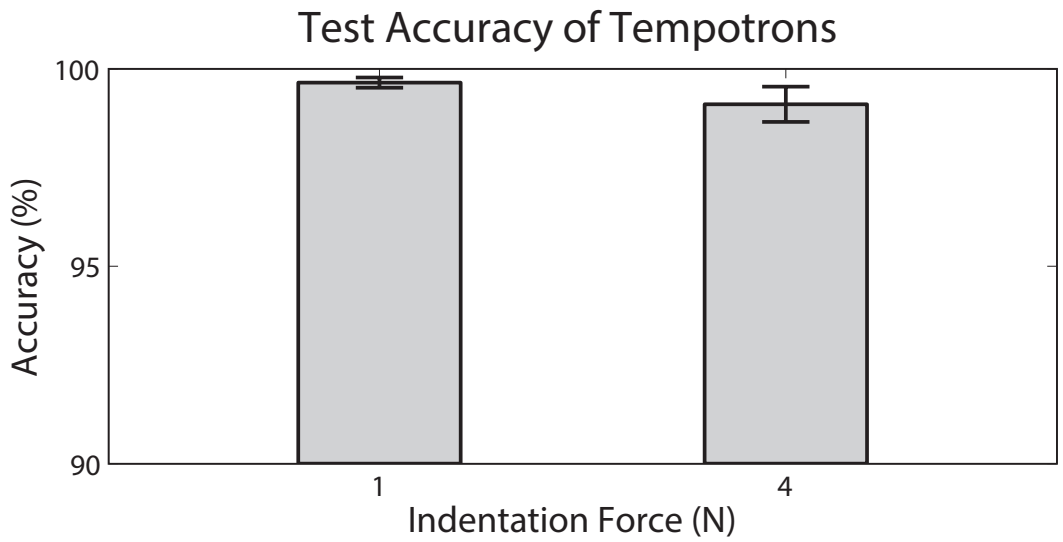


Figure 3.5: Classification accuracy of Tempotrons for 1N and 4N indentations. Error bars indicate standard deviation of mean. © 2013 IEEE.

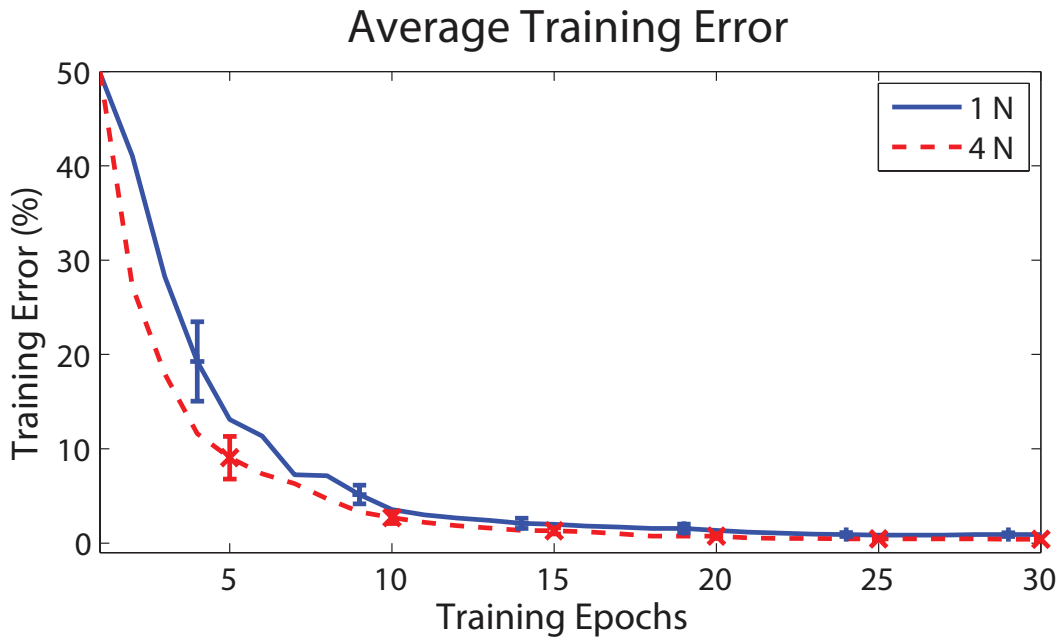


Figure 3.6: Training Error of Tempotrons. Error bars indicate standard deviation of mean. © 2013 IEEE.

data rates. The sensor's soft and compliant nature inherently acts as an intensity to latency filter, producing spatiotemporal spike patterns that are distinctive to the contact stimulus. Despite having only one bit of pressure resolution per element, we have shown that a good first approximation of the contact stimulus is still achievable. The approximation may be optimal for certain tasks such as object manipulation, where system responsiveness may be more important than having accurate pressure measurements.

The presented system relies on a learning algorithm for pattern recognition, thus avoiding the need for sensor calibration. However, given the vast variety of tactile stimuli that may be presented, the use of a supervised learning algorithm may not be ideal. Development of unsupervised learning tools for spike patterns would be an important area of research to improve the usability of this approach.

As a preliminary investigation, we note that the design of the experiment is highly simplified. For instance, the use of a fixed pressure threshold would not be practical for real-life applications, as depending on the threshold chosen, the output could be saturated before the sensor comes into contact with an object. Sensors based on piezoelectric principles may be better suited for the detection of pressure changes.

3.7 Significance

The work presented in this section serves as an early implementation of our proposed neuromorphic tactile sensing concept. The novelty lies in the spike-based representation of signals, and the use of a soft sensor array to capture spatiotemporal patterns representative of sensor deformation dynamics. Finally, the use of a single-threshold highlights the possibility of improving the scalability and reducing the latency of large scale sensor arrays by reducing pressure resolution, and is a precursor to the more advanced sensor platform presented in the next chapter.

Due to the conceptual differences between the proposed neuromorphic approach and conventional tactile sensing techniques, a direct comparison between the two methods is not practical. Table 3.1 summarizes the main differences between the two approaches. In most conventional tactile sensors, the focus is on providing accurate and reliable pressure

Table 3.1: Conceptual differences between neuromorphic and conventional approaches

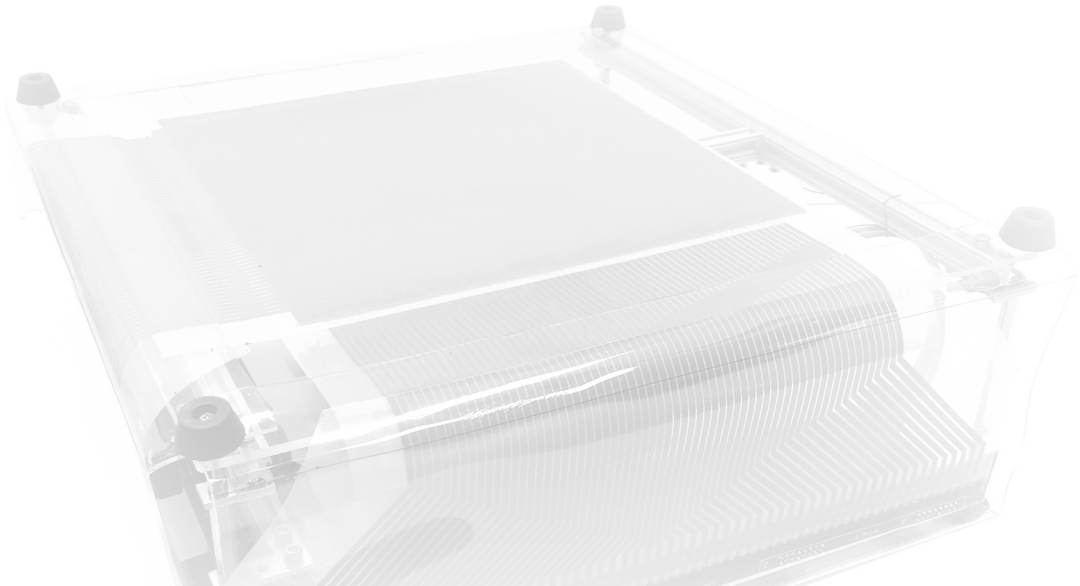
	Conventional approach	Neuromorphic approach
Transduction property	High accuracy, minimal hysteresis and drift, linear response	Accuracy, linearity, hysteresis and drift performance not critical, as long as behavior is consistent
Transducer placement	Limited elements with specific placement	Many elements randomly distributed over large area
Time constraints	Intensity values at all elements propagated in a single frame	Change and intensity information propagated on different time scales
Readout latency	Determined by Nyquist frequency of target signal	Changes propagated as rapidly as possible, intensity values can take few tens of ms
Mechanical properties	Planar, thin, flexible, stretchable	Deformable surface, multi-layered arrangement
Target Applications	Instrumentation and diagnosis	Robotics and prosthetics, real-world interaction

measurements and is thus better suited for instrumentation and industrial applications. In comparison, our neuromorphic approach is biased for scalability and minimal latency, relying instead on computational algorithms to derive estimates of contact parameters. We argue that our approach is more practical for machines that operate in real-world environments, such as robots and prosthetics, where they would experience a much wider range of contact events.

4

A Kilotaxel Kilohertz Tactile Sensor Array

Lee, W. W., Kukreja, S. L., & Thakor, N. V. (2015, October). A Kilohertz Kilotaxel Tactile Sensor Array for Investigating Spatiotemporal Features in Neuromorphic Touch. In *Biomedical Circuits and Systems Conference (BioCAS), 2015 IEEE* (pp. 1-4). IEEE.



Contents

4.1 Preface

As demonstrated in Chapter 3, spatiotemporal coding of tactile data is potentially a unique and effective signal processing strategy. However, the signal acquisition in the previous sensor array was achieved using a high-end data acquisition unit producing almost 70 MB/s of data for an 8×8 array, which is clearly not scalable. We also realized that no commercially available tactile sensor array has the ability to interface at the scale and speed comparable to biological mechanoreceptors. This necessitates the development of a new sensor platform that elucidates the advantages of spatiotemporal spike processing for complex, real-life scenarios.

This chapter describes the development of a 4096 element tactile sensor array that samples at over 5200 frames per second. Besides characterization tests, preliminary experiments were conducted to demonstrate how the high spatiotemporal resolution aids the resolution of force magnitude and direction from tactile events. The unprecedented spatiotemporal resolution demonstrates that using sparse event-based representations, it is possible for artificial tactile sensors to match the performance of biological tactile sensing in scale and speed. Subsequent chapters also make use of this platform to demonstrate the use of temporal structure in tactile learning and signal processing.

4.2 Introduction

The human hand has over 17,000 mechanoreceptors [24], with individual axons projecting to the dorsal root of the spinal cord, forming a massively parallel communication pathway. Despite the relatively slow firing rate of mechanoreceptors (about 100 Hz maximum), we are capable of responding reflexively to transient events in as little as 65 ms [38]. The ability of mechanoreceptors to respond with temporally precise action potentials is postulated to be a critical step to achieve such performance [38]. As the skin deforms upon contact, distributed patterns of stress trigger tactile afferents in particular sequences to elicit spike patterns that are spatially and temporally distinct. These patterns provide temporal precision with an accuracy of up to 0.8 ms, giving rise to a rich representation of contact parameters [38]. Neurophysiological experiments have

Table 4.1: Comparison of the Kilotaxel Kilohertz sensor array to the human hand and tactile sensor arrays of comparable scale/speed.

Sensor	Number of taxels	Sampling rate (Hz)	Output format	Ref
Gifu Hand III	859	100	8 bit pressure	[171]
Slip detection array	256	1900	12 bit pressure	[103]
FPGA data compression	640	200	Fitted ellipses of pressure distribution	[89]
CellulARSkin network	780	250	Events with 12 bit pressure	[172]
TactArray system (Pressure Profile Systems)	1024	10	12 bit pressure	[60]
Industrial I-scan system (TekScan)	1936	100	8 bit pressure	[173]
Human hand	∼17000	>1000	Action potentials	[24]
This work	**4096**	**5200**	**Events with 1 bit pressure change**	[174]

shown that spatiotemporal representations enable rapid approximation of key features such as force magnitude, direction, local curvature and edges [24, 40].

No artificial tactile sensors today can match the scale and temporal precision of tactile afferents, despite over 40 years of active research [1, 3, 175]. Sensor arrays with greater than 1000 elements rarely have a sampling rate above 100 Hz [89] (Table 4.1). A paradigm shift in methods of signal acquisition and representation would be needed to produce sensor systems with specifications comparable to mechanoreceptors. With the emergence of asynchronous imaging hardware [158–160], spatiotemporal patterns have been suggested as a concise representation in vision [176]. Without suitable hardware it is difficult to explore similar concepts for touch.

Tactile sensing differs fundamentally from vision in that the sensing elements are distributed over a much larger surface. Ideally, the sensors also have to be flexible and/or stretchable to conform to uneven surfaces. As described in Chapter 2.2.3, it is challenging to produce conductive traces that are stretchable and of high density. This was not an issue in biological systems, as nerve bundles are inherently stretchable and

are capable of exquisitely high density [86]. Almost all designs resort to multiplexing in order to reduce the number of wires down to a more manageable quantity [1]. While this may mitigate the challenges in mechanical design, multiplexing reduces readout speed as the connections are now shared [94].

Fundamentally, readout speed is constrained due to (a) multiplexing of many sensing elements to a few Analog-to-Digital Convertors (ADC), (b) moving data from ADCs to controller, (c) moving data from controllers to a central processor, (d) degradation of signals when propagated over longer distances.

To improve sensor responsiveness a common approach is to compromise on spatial resolution. Fukui et al. adopted an intelligent scanning technique that surveys a subset of the array then narrows the region for greater detail when a contact event is detected [97]. Oballe-Peinado et al. used a frame-based approach where the captured tactile image was compressed into an elliptical representation for lower bandwidth utilization [89]. Shimojo et al. designed a sensor to capture only the center and magnitude of the load to enable high temporal resolution and minimize wiring [90]. Although a sampling rate of over 1 kHz can be achieved in some cases, the reduction of spatial resolution is significant [89, 90] or the spatiotemporal structure is not preserved [97].

We take a novel approach to mitigate these constraints, and developed a tactile sensing system optimized for high speed scanning of many elements. A column-parallel readout strategy was implemented to rapidly scan the array with minimal discrete components. The circuitry was controlled by a FPGA to take advantage of its parallel processing capabilities. The configuration presented in this chapter has 4096 resistive elements and benchmarked to a sampling rate of 5.2 kHz.

The chapter is organized as follows. In Section 4.3, we describe the sensor construction and readout circuitry. Section 4.4 characterizes the sensor's response time and demonstrates the utility of high temporal resolution in contact parameter discrimination. Section 4.5 provides a discussion of the system's potential and provides a summary of the major findings of our work.

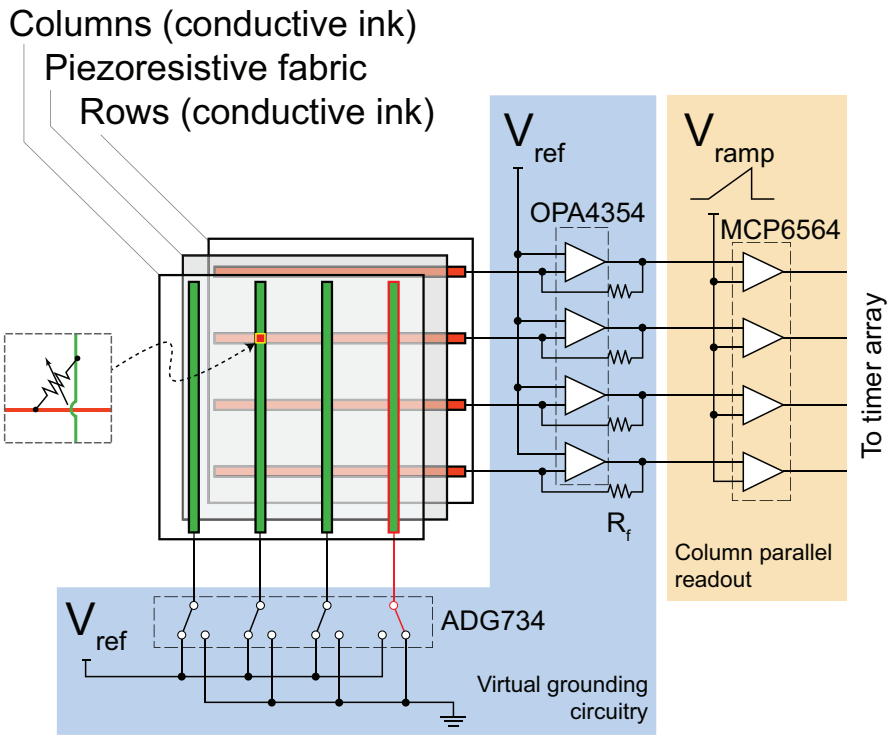


Figure 4.1: Sensor and circuit architecture. Active columns (highlighted in red) are connected to ground. All other rows and columns are held at the same voltage to minimize crosstalk. © 2015 IEEE.

4.3 System Architecture

4.3.1 Sensor design

We assembled a flexible 64 × 64 sensor array using inkjet printed conductive traces and piezoresistive fabrics. The force transducer consists of a piezoresistive fabric (LTT-SLPA, Eeonyx) sandwiched between sheets of polyethylene that are printed with conductive traces (AgIC, Tokyo, Japan). The sheets are arranged in a grid-like pattern of rows and columns, where each intersection formed a sensing element (Fig. 4.1). The active area of the sensor measured 17 × 17 cm, with an average separation of 2.67 mm between elements.

Coding by spatiotemporal spike patterns is feasible only when a sufficiently large number of mechanoreceptors are triggered upon contact. In biology, this is achieved by having high receptor density in the skin coupled by a large ratio of receptive field overlap [24]. We recreate such properties in our artificial sensor array by placing a 10 mm thick layer of polyurethane foam on top of the sensor array, resulting in a spatial

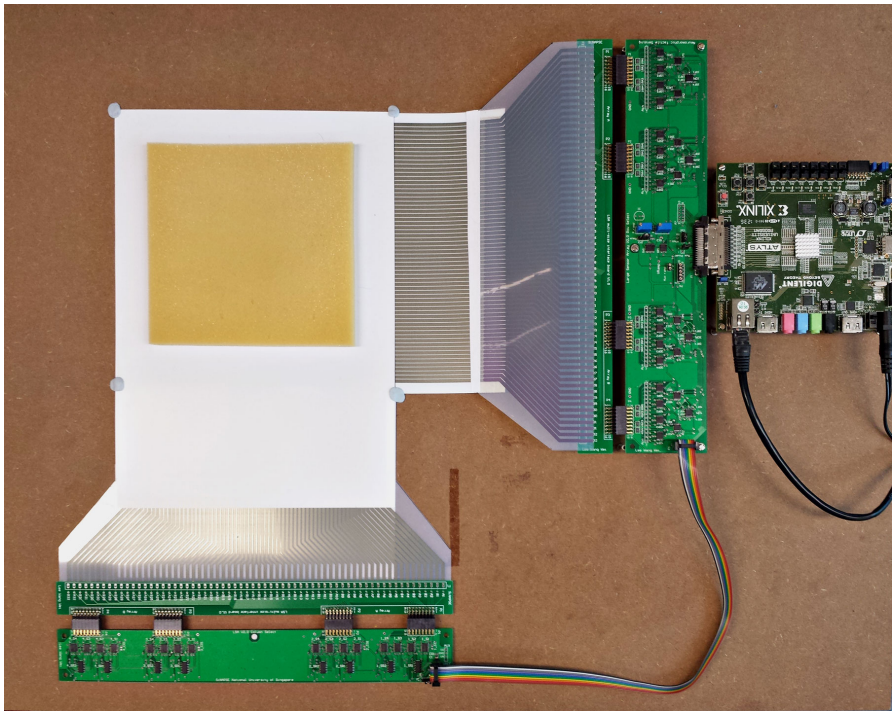


Figure 4.2: Photograph of sensor system.

smoothing effect. This allows good replication of contact force distribution over a region and facilitates the activation of a population of elements.

4.3.2 Readout circuitry

Cross-talk cancellation

Row-column multiplexing is a common approach to reduce wiring in tactile sensor arrays [1,76]. However, since there are multiple elements on the same row and column, parasitic currents may flow [178], resulting in inaccurate readouts. This is particularly significant for a large array of elements because the number of parasitic current pathways grows at an exponential rate.

We adopted a *virtual grounding* approach [178] to minimize problems associated with parasitic currents. All row conductors were held at the same voltage (V_{trans}) using transimpedance amplifiers, while column conductors were also held at V_{trans} except for the readout column, which was grounded (Fig. 4.1). This arrangement ensured that the current only flows from the rows to the readout column since all other pathways were

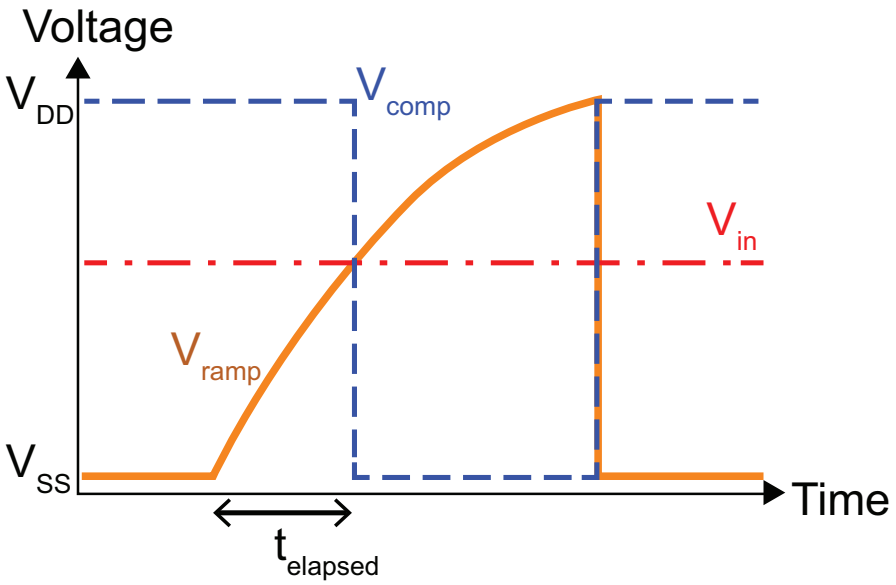


Figure 4.3: Single-slope ADC concept. V_{ramp} and V_{in} represent inputs to a binary comparator, where V_{in} is the signal to be digitized. The time interval (t_{elapsed}) between an increase in V_{ramp} to a change in digital output V_{comp} represents signal intensity. © 2015 IEEE.

virtually shorted. An additional advantage to this approach is that the entire column can be read simultaneously, thus increasing readout throughput.

Analog to digital conversion

We established A/D conversion using an array of single-slope comparators to reduce communication overhead. The circuitry consisted of an array of comparators, where one input corresponded to a ramp voltage (V_{ramp}) while the other was connected to the analog signal (V_{in}) from the sensor. The digital output of each comparator (V_{comp}) indicated whether V_{in} was higher than V_{ramp}.

Fig. 4.3 illustrates the concept of the single-slope ADC. To initiate signal conversion, a controller starts the voltage ramp on V_{ramp} while simultaneously starting an array of timers (one for each channel of V_{comp}). The time interval (t_{elapsed}) between the start of a ramp to a comparator output change is recorded for each channel. Since V_{ramp} increases monotonically during each sampling period, the duration of t_{elapsed} is proportional to the magnitude of V_{in}.

We implemented analog switches with low on-resistance to minimize error (ADG734, Analog Devices). The switches were driven by a shift register array (74HC164, NXP) to

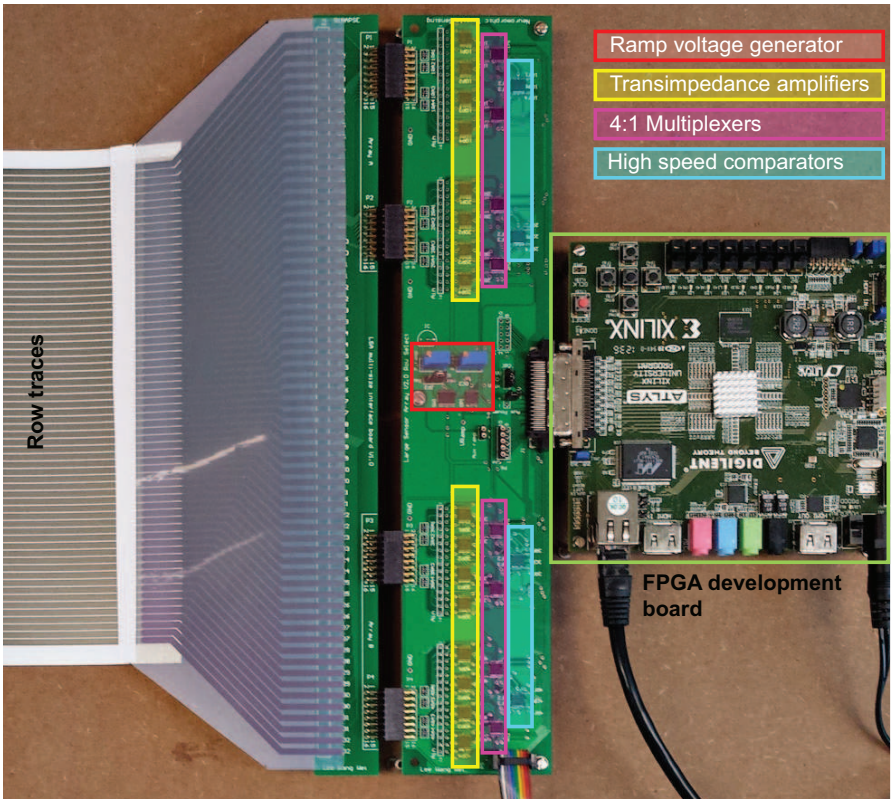
reduce the number of pins required. We used operational amplifiers (OPA4354, Texas Instruments) with feedback resistors $R_f = 47\,k\Omega$ for individual rows. A 4:1 multiplexing (ADG709, Analog Devices) of V_{in} to the comparator was applied to reduce input pin count. In addition, the comparators (MCP6564, Microchip) had a 0.1 V external hysteresis added for stability. A resistor-capacitor (RC) circuit was used to generate V_{ramp}. The timers were executed at 300 MHz and implemented on a Xilinx Spartan 6 FPGA (Atlys, Digilent) Fig. 4.4 illustrates the layout of the circuit. The schematics for the circuitry are presented in the appendix.

In addition to a significant reduction of the number of components needed and communication overhead required, a single-slope ADC approach allows for an adjustable conversion speed. This property permits the implementation of biomimetic sampling strategies. For example, a steep ramp slope may be used during gross movements for responsive feedback while a gentle ramp may be utilized during fine control, where high intensity resolution may be necessary.

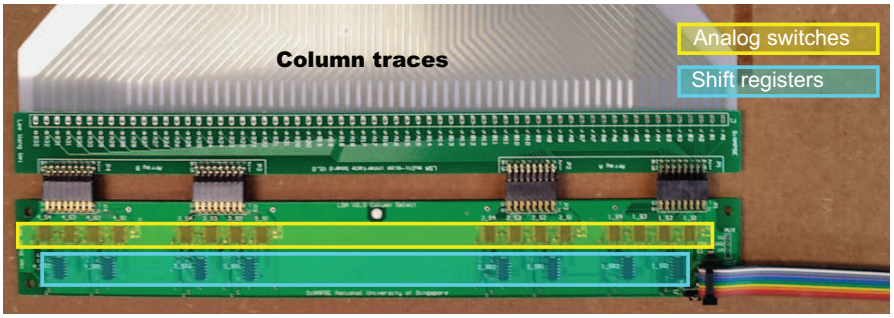
FPGA logic implementation

To minimize temporal redundancy and improve output bandwidth utilization, the FPGA was configured to output data as event packets instead of a continuous stream. The logic architecture for signal acquisition in the FPGA is described in Fig. 4.5. The value of $t_{elapsed}$ is obtained by an array of timers that execute in parallel. In addition, a local memory bank is instantiated in the same component to store the previous value of $t_{elapsed}$ for each taxel. Immediately after a new sample is obtained, a comparison with the previous sample is performed, and an output event is only generated when a significant change in $t_{elapsed}$ is detected. Time-stamps are assigned to the output events by a dedicated logic block to ensure timing precision.

A soft microprocessor (Microblaze, Xilinx) is instantiated within the FPGA hardware to simplify communication between the PC and the sampling logic. Upon receiving instructions from the PC, the microprocessor communicates with the sampling logic using fast-simplex-logic (FSL) buses to configure a bank of registers. Acquisition settings such as sampling rate and pressure sensitivity are adjusted through the registers. Output from the sampling logic to the PC is also handled by the microcontroller, where event



(a)



(b)

Figure 4.4: Printed circuit board layout for (a) row interface (b) column interface.

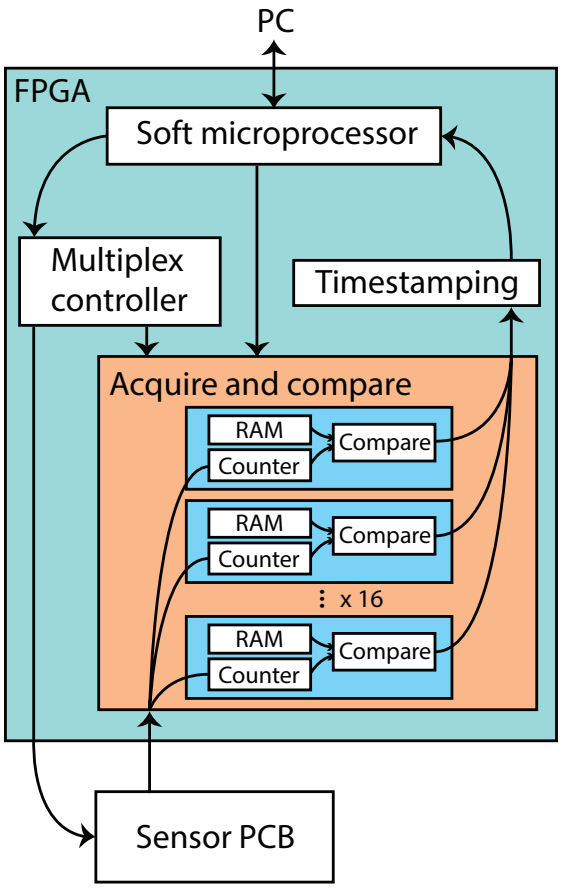


Figure 4.5: Functional block diagram of FPGA implementation.

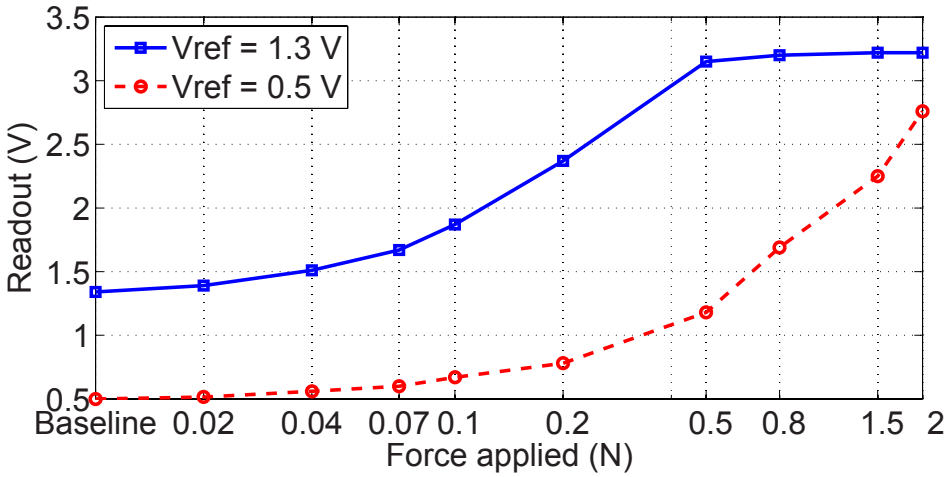


Figure 4.6: V_{in} for a single taxel response to applied loads. Two load values (V_{trans}) used (blue) 1.3 V and (red) 0.5 V. Note the semi-log scale (abscissa). © 2015 IEEE.

packets are formatted according to the asynchronous user datagram protocol (UDP) and transmitted via Ethernet.

Specific implementation details can be found in the Appendix.

4.4 Experiments

4.4.1 Characterization tests

To characterize the response of the sensor to applied loads, we delivered calibrated forces using Semmes-Weinstein monofilaments (Baseline®) on one taxel and observed the voltage response, V_{in}. Fig. 4.6 illustrated the results of this analysis. Note that the sensitivity and dynamic range of the system can be tuned by adjusting V_{trans}, R_{f} or gradient of V_{ramp}. For example, with $V_{\text{trans}} = 1.3\,V$, a force of 0.02 N can be reliably detected, at the expense of a reduced dynamic range.

Having a high-speed sampling circuitry is of little use if the transduction takes place slowly. A separate experiment was thus conducted to assess the reaction time of the piezoresistive transducer material, where we apply an impact to a pin positioned directly on top of a sensor element. Both the pin and impactor were made of metallic conductors. The impactor was connected to a 5 V voltage source and the pin connected to an oscilloscope (2 GSa/s, Agilent MSO7104B). The V_{in} channel corresponding to

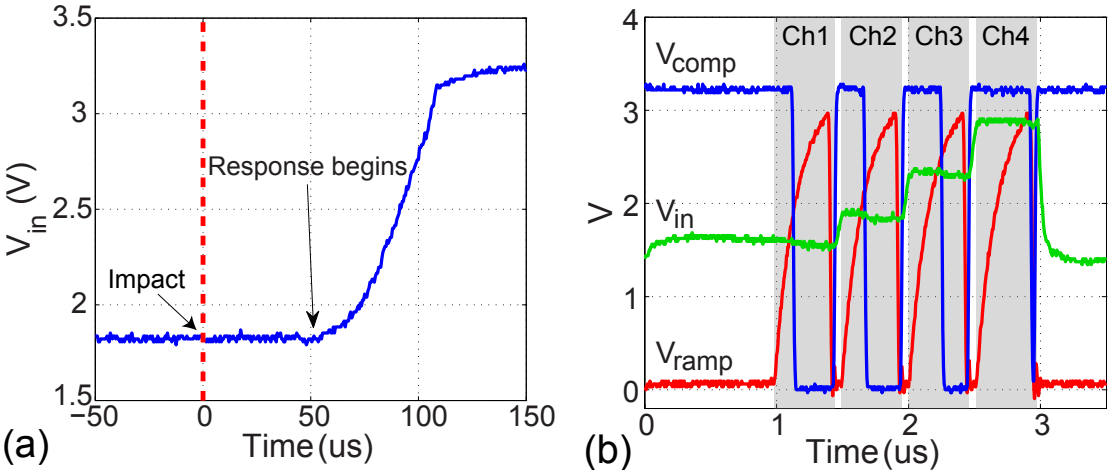


Figure 4.7: (a): Response time of sensor to impact. Dotted line: time of impact. (b): Comparator response (blue) while sampling 4 channels. Red signal corresponds to ramp voltage while the green trace is the voltage to be sampled.

the sensor element was also connected to a separate channel on the same oscilloscope. Sampling both channels simultaneously, we determined the response time as the delay from the start of the contact signal and start of change in V_{in}. Fig. 4.7a illustrates the response waveform. The sensor is shown to respond at approximately 50 μs, which was within acceptable limits for this application.

Next, we investigated response time of the circuitry while multiplexing. Fig. 4.7b shows two inputs to one of the comparators while it is multiplexed among 4 channels of V_{in}, at a sampling rate of 5.2 kHz (3 μs/column). The panel illustrates that V_{in} settles within 0.2 μs, which is within design tolerances for this application.

4.4.2 Preliminary experiment

We conducted a simple experiment to demonstrate the utility of high spatiotemporal temporal resolution provided by the system to discriminate impact forces of varying magnitude and direction. A tennis ball (⌀65 mm, 56 g) was dropped from a height of 40 cm and 80 cm onto the sensor array perpendicularly, as well as with the sensor at a 30° slant. One hundred trials were conducted for each combination of drop height and sensor slant. The duration of each trial (0.6 s) was long enough for the ball to hit the sensor only once. The sensor was sampled at 2 kHz because we heuristically found it to be a good trade-off between sampling rate and intensity resolution.

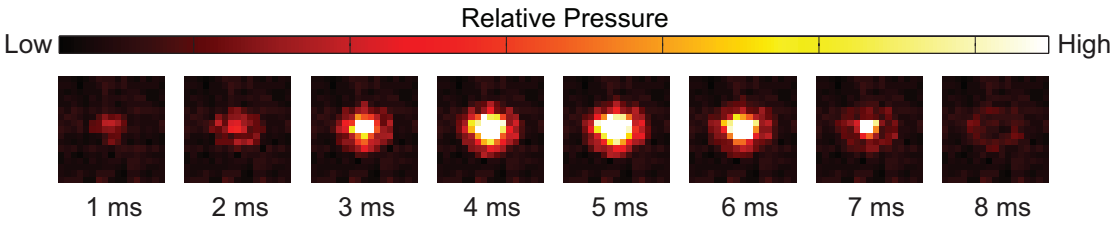


Figure 4.8: An 8 ms slice of a tactile video showing pressure distribution over a segmented 17×17 element region. The trial was of a perpendicular impact by a tennis ball as it was dropped from a height of 40 cm. Due to the readout technique employed, all pressure measurements are relative to the start of impact. © 2015 IEEE.

We extracted each impact as a region of 17×17 elements drawn around the element that reported the first event. Tactile images corresponding to a pressure distribution of the area was constructed from time of impact to end of trial. To simulate slower sampling rates, the sequence was re-sampled at a lower rate. Fig. 4.8 shows an 8 ms slice from a randomly selected trial re-sampled at 1 kHz.

We visualize the separability of the 4 different cases using an unsupervised technique, t-Stochastic Network Embedding (t-SNE) developed for the plotting of high dimensional data [179]. We used a 20 ms data segment as an input to the algorithm. The output is depicted in Fig. 4.9, where a high temporal resolution (500 Hz and above) clearly demonstrates better separability of the classes.

To further quantify separability, we trained a linear classifier (LDA) to perform the separation task. The classifier was tested via a 10-fold cross validation. Lastly, we computed the mutual information between the ground truth label (L) and response of the classifier (R), defined as

$$I(R;L) = \sum_{R,L} p(R|L) \cdot p(L) \log_2 \frac{p(R|L)}{p(R)} \tag{4.1}$$

where $p(R|L)$ is the confusion matrix of the classifier. This approach presents a more objective performance measurement since it takes into account the distribution of errors (non-diagonal values in confusion matrix), not just the number of *hits* (diagonal values) [176]. Fig. 4.10 presents the findings of this study. Although sampling rates below 100 Hz were insufficient to capture meaningful information due to the transient nature of the event, significant improvements were achieved when sampling above 100 Hz. This

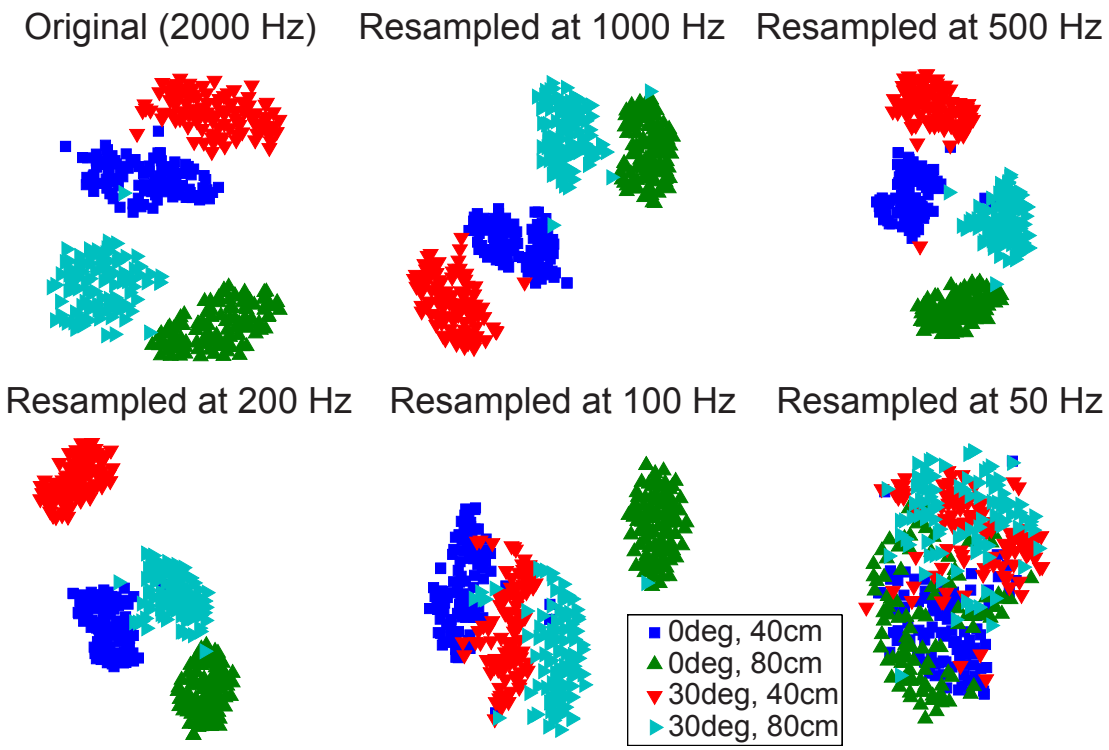


Figure 4.9: Unsupervised clustering using t-SNE. Tactile videos of the first 20 ms after impact were used as inputs. The same video was re-sampled at the respective rates to simulate lower sampling frequencies. Note how the clustering is more distinct at higher sampling rates, indicating better discriminability of the contact events. As t-SNE is a technique to project points in high-dimensional space onto a 2D surface while best preserving the relative distances between points, the axis in this plot are arbitrary.

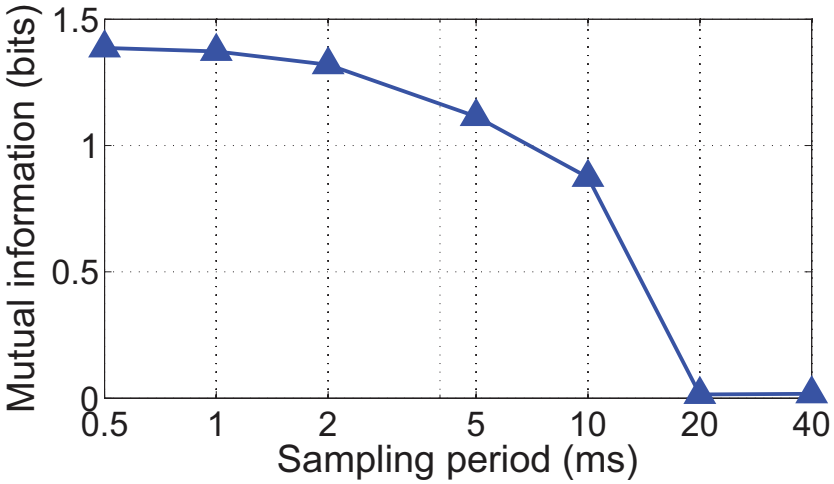


Figure 4.10: Mutual information of classifier response vs. temporal resolution of inputs. Tactile videos of 40 ms duration resampled at the respective sampling periods were used as feature vectors during classification. © 2015 IEEE.

suggests that high spatiotemporal resolution provides additional information to facilitate accurate classification, particularly during transient contact events.

4.5 Discussion and Conclusion

A tactile sensor array optimized for high speed sampling of numerous elements was presented. This system will serve as a platform to investigate the utility of time-sensitive mechanisms under experimental settings. Besides system characterization, we also conducted a realistic experiment that demonstrated the advantage of having increased temporal resolution to discriminate contact parameters of transient events.

There is mounting evidence in biology that suggests temporal encoding of neural signals has distinct advantages for fast changing events. Although the materials and techniques employed to realize our sensing device may not be revolutionary from a technical standpoint, the concept of treating tactile signals as spatiotemporal activation patterns is where the innovation lies. The availability of such a sensor will allow us to develop novel pattern recognition strategies with greater realism and convenience.

The current prototype does not have the sensor density to match the mechanoreceptors on human fingertips. However, advanced fabrication techniques has enabled the realization of sensor arrays comparable to mechanoreceptors in terms of sensitivity,

density, flexibility and stretchability [57, 76]. Many of the designs employ the resistive row-column multiplexing arrangement presented in this work, which makes our contribution relevant.

Accuracy of intensity measurements may be compromised using our approach because the A/D conversion was performed using multiple discrete components that may not be well matched. This issue can be solved by combining the circuitry onto a single chip, similar to the column-parallel ADCs that are widely used for high-speed image capture [180]. However, the goal of the thesis is not about accurate intensity measurements, but on methods to achieve responsive tactile sensing over a large area. Moreover, studies suggest that highly variable, nonlinear and hysteric coding of intensity is provided by mechanoreceptor responses and that calibration is most likely performed at a higher level in the tactile processing pathway [24]. This is in line with our objective, where low cost sensors with high spatiotemporal resolution are interfaced with neuromorphic learning architectures to produce robust and reliable tactile representations.

We note that ideally, a fully asynchronous architecture where computation is performed at individual sensing elements would enable even better power and computational efficiencies. Such an approach has been implemented in neuromorphic vision and auditory sensors as described in Chapter 2.3.4, resulting in very low latency and low power devices. As the fabrication of flexible organic transistors mature [181], a fully asynchronous sampling circuitry for tactile sensors should be achievable in the near future. As it now stands, we believe our current approach is a good compromise between scalability and ease of fabrication, serving as an interim solution for algorithmic development and analytical purposes.

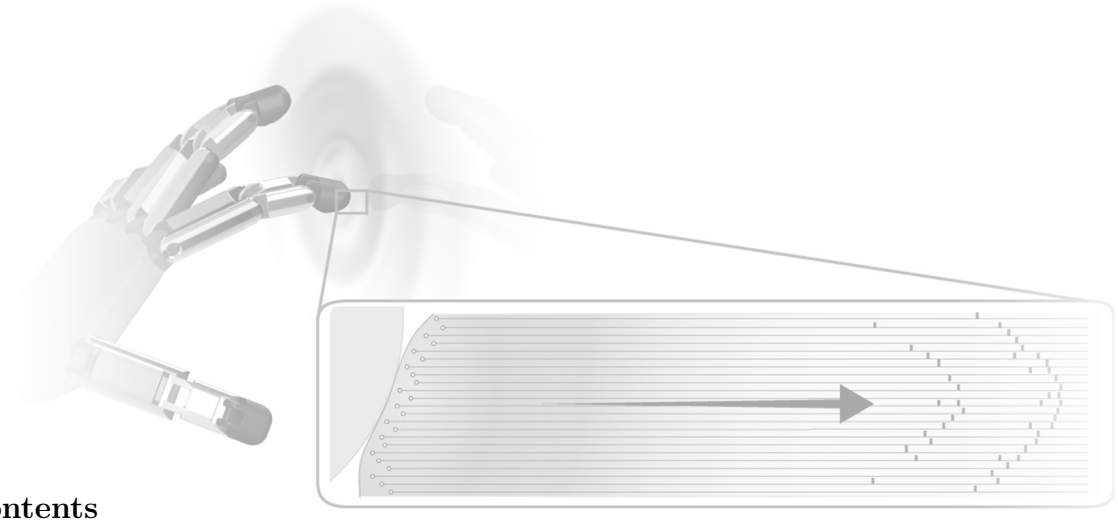
4.6 Significance

The sensor platform enables the capture of tactile signals at a comparable spatiotemporal scale as human mechanoreceptors. As illustrated in Table 4.1, no commercially available sensor array could deliver tactile information at the scale and rate needed for the physical validation of our neuromorphic approach. From a technical standpoint, we do not claim the sensor array to be groundbreaking. Indeed, the multiplexing technique

was adopted from [178], while the single slope ADC concept has been widely used in many applications. The emphasis, however, is on our signal processing technique. The use of spatiotemporal representations prioritizes the temporal precision of change detection above the accuracy pressure measurements. Using our technique, a simple ADC circuit built using discrete components is sufficient, and thus the large array could be interfaced without the use of custom chips. Ultimately, our goal is to show that using spatiotemporal structures alone, relevant information on tactile events could be extracted with reasonable accuracy but within a fraction of the current time needed, thus ensuring that the overall responsiveness and scalability of the system is maintained. This will be discussed thoroughly in Chapter 5.

5

Temporal Coding in Neuromorphic Tactile Sensing



Contents

5.1 Preface

Earlier experiments suggest that substantial information extracted from the temporal dynamics of tactile signals are useful for contact discrimination. However, the variety of stimuli used in earlier experiments are rather limited and not sufficient for making a conclusive statement. In this chapter, we develop a finite-element-model (FEM) to simulate contact stimuli with high consistency and precision, while also allowing contact parameters to be easily varied. By investigating 300 variations of contact simulated through FEM, together with 12 variations performed physically on the high speed sensor platform, this chapter presents an objective study on the utility of temporal structure for tactile sensing.

5.2 Abstract

This work presents a neuromorphic tactile encoding methodology that utilizes a temporally precise event-based representation of sensory signals. We introduce a novel concept where touch signals are characterized as binary events to denote pressure changes. This approach is amenable to a sparse signal representation and enables the extraction of relevant features or events from thousands of sensing elements with sub-millisecond temporal precision. This technique is implemented on a state-of-the-art 4096 element tactile sensor array with 5.2 kHz sampling frequency, which permits the classification of transient impact events. Using finite element simulations, we synthesized spiking sensor responses to 300 different contact conditions. Our results demonstrate that significant information is embedded in the temporal structure of signals at millisecond precision, which is lost when sampling at conventional rates of $\leq$100 Hz. In addition, precise temporal information reduces latency of contact event discrimination. Hence, a neuromorphic spatiotemporal representation of touch is well suited to rapid identification of critical contact events, making it suitable for dynamic tactile sensing in robotic and prosthetic applications.

5.3 Introduction

The increased expectation of robots to be able to interact in natural dynamic environments requires the ability to quickly detect, recognize and respond to physical contact events. Such capabilities are necessary for dexterous manipulation [182] and to ensure the safety of the user and robot [175, 183]. Ideally, the entire exterior of the robot should be sensitive and responsive to touch. These requirements highlight the need for a rapid acquisition and processing mechanism to interface with thousands of tactile sensing elements distributed across the surface of a robot.

In comparison to other sensing modalities such as vision and hearing, the difficulties in acquiring and processing data from numerous tactile sensing elements, or *taxels*, have been cited as a primary reason for the limited progress in this area [184]. The challenges are a result of taxels being spread over a large and non-uniform surface area. To minimize clutter and wiring complexity, the sampling and readout processes are often heavily time-multiplexed, leading to reduced readout data rates.

Currently, the fastest tactile sensor arrays can communicate at 2 kHz but with only 256 sensing elements [103], which is insufficient to cover the body of a reasonably sized robot (See Table 4.1). Limited communication bandwidth has been cited as a restriction on the number of sensing elements on an array. Although future developments of high speed communication protocols will increase the sampling rate of tactile sensors, it is unclear whether such a large volume of information is necessary. Increased data acquisition requires more complex algorithms and hardware, which leads to higher implementation costs and power consumption. A practical approach is to intelligently reduce the data generated from the sensing front-end, prior to transmission.

Spatial resolution is often compromised to improve response times. Shimojo et. al. developed a sensor optimized to only detect the centroid and magnitude of an applied load [90]. This enables rapid detection of contact (1 ms) but leads to severely compressed spatial resolution of a single point. Oballe et. al.'s implementation [89] compressed the tactile image into an ellipse to represent location, distribution and magnitude of a load. This work is similar to Shimojo et. al. since there is only enough resolution to represent a single point of contact. The technique employed by Fukui et. al. [97] selectively sampled

the sensor array based on a genetic algorithm, increasing the probability of detecting a contact event without increasing the sampling rate. However, this approach is ideally suited for localized contacts since scanning a large area is still slow due to a limited acquisition speed of 0.2 ms per taxel. Similarly, the sensor array presented by Schmitz et. al. [185] can only achieve a higher sampling rate of 500 Hz by averaging all sensing points. Although these techniques permit rapid detection of contact events, reduced spatial resolution does not provide accurate estimates of key contact properties such as force magnitude, direction and object curvature [7, 38]. Precise estimates of these characteristics are critical for sensory control during object-oriented actions [24].

The human somatosensory system is much more responsive than even the fastest state-of-the-art artificial tactile sensors. With over 17000 mechanoreceptors on the palmar side of each hand, humans demonstrate exquisite reflex, responding within 65 ms of a contact event [7]. Due to multiple layers of neural processing and slow conduction velocities of axons [186], there is only an estimated ∼5 ms window for data acquisition [38]. Neurophysiologists postulate that the temporal structure of action potentials from a population of mechanoreceptors may serve as a neural code that facilitates rapid sensory processing in the brain [38, 40].

Mechanoreceptors innervating the glabrous skin are classified as fast adapting (FA), or slow adapting (SA) [24]. SA receptors generally fire action potentials (spikes) at a frequency proportional to the intensity of the stimulus, while FA receptors respond mainly to changes in stimulus intensity. These two classes are further categorized as Types 1 and 2, with Type 1 receptors having a much smaller receptive field than Type 2 receptors [4]. Although there is considerable interest in imitating SA-1 afferents due to their seemingly straightforward encoding of pressure [104,105,187], researchers have also started to take an interest in the FA-1 mechanoreceptors. A unique property of FA-1 mechanoreceptors is their ability to respond to stimuli with temporal precision as high as 0.8 ms [38]. In addition to the large overlaps in receptive fields, FA-1 mechanoreceptors are often activated as a population and in a particular order as the skin deforms upon contact. This enables encoding of contact parameters in the timing and sequence of spikes. Spatiotemporal features embedded in FA-1 afferent population responses have been shown to encode edges, local curvature, force magnitude and direction [24, 40]. It

has been postulated that these spatiotemporal response patterns are highly compressed representations that enable rapid signal propagation and processing to occur [11, 38].

The utility of spatiotemporal pattern recognition is increasingly apparent in artificial tactile sensing research. Lee et. al. [188] demonstrated such an approach for discriminating between two indentation conditions and for detecting three different gait events [189]. Spatiotemporal patterns were generated by detecting the time each sensor element crossed a pre-defined pressure threshold for a given stimulus. However, the use of a fixed threshold is only practical for properly calibrated arrays. Rongala et. al. showed an improved accuracy for texture recognition using spatiotemporal spike patterns compared to using spike rates alone [105]. Subsequent experiments conducted by Oddo et. al. demonstrated that the temporal structure of signals used in peripheral nerve stimulation enhanced the ability of human subjects to discriminate textural features [41]. In both papers, an Izhikevich neuron model was used to mimic the SA-1 mechanoreceptor and its adaptive behavior. Although the authors commented on the importance of precise spike timing, the temporal precision was not quantified in detail. Nevertheless, these results suggest that important information is present in the temporal dynamics of sensor responses. These precise spatiotemporal patterns are a biomimetic representation that could enable rapid processing while using less communication bandwidth.

In this chapter, we investigate the hypothesis that spatiotemporal dynamics of a sensor array response carry critical information to enable the discrimination of contact events. In Section 5.4 we describe the finite-element-model (FEM) used in the study followed by the results obtained using synthetic data in Section 5.5. The validation of FEM analysis through physical experiments, conducted on our high speed tactile sensor array, is provided in Section 5.6. Section 5.7 describes how our neuromorphic representation of touch signals, as binary events, facilitates the expression of spatiotemporal structures with high temporal fidelity. In Section 5.8 we provide a discussion about our findings and future directions while Section 5.9 concludes the chapter.

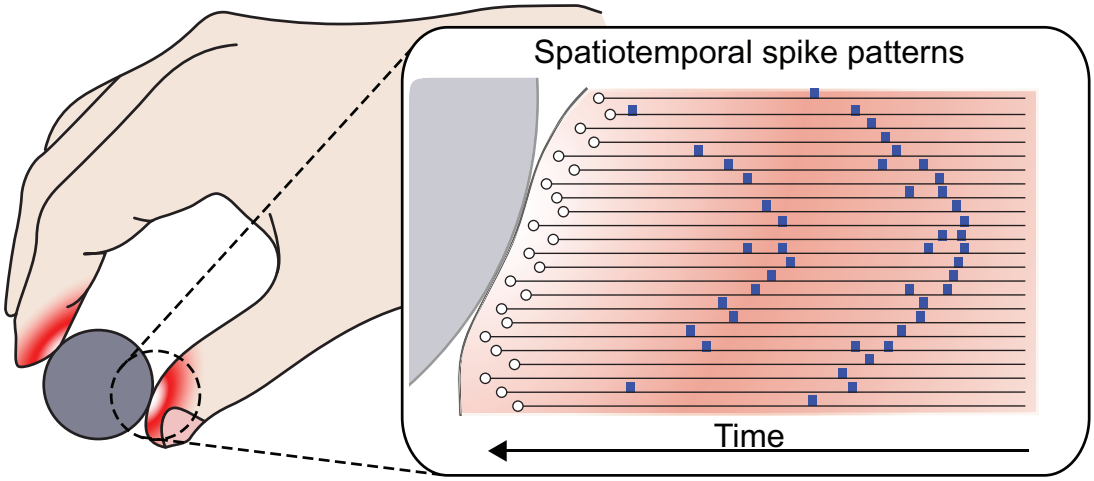


Figure 5.1: Formation of spatiotemporal spike patterns. Skin deformation during mechanical contact produces distributed patterns of pressure that triggers tactile afferents in a precise sequence to produce a spiking response pattern.

5.4 Methods

In this section, we discuss the origins of spatiotemporal features in mechanoreceptor responses. Inspiration from how biological mechanoreceptor function is used to simulate the response of a tactile sensor array using a FEM approach. This synthetic data is used to study how information content changes as temporal precision is varied for 300 different combinations of contact parameters.

5.4.1 Monte Carlo FEM simulation

During mechanical contact, skin deformation results in distributed patterns of pressure that trigger tactile afferents in a precise sequence, creating spike patterns that are spatially and temporally distinct (Fig. 5.1). The number of unique representations or dimensionality of a spatiotemporal code is dependent on the a) length of decoding time window, b) temporal precision of the spikes and c) number of mechanoreceptors involved. Experimentally, it has been observed that human fingertips are innervated with an exceptional density of mechanoreceptors of over 140 afferents/cm^2 [24]. These afferents, mainly the FA-1 and SA-1s, exhibit highly overlapping receptive fields, partly as a consequence of the soft and conformable supportive tissues beneath [24]. Combined with sub-millisecond temporal precision, spatiotemporal spiking patterns have the potential to encode a large range of stimuli within a short time window. Using this biological

mechanism as an inspiration, we designed our simulations and experiments to target soft sensors with high density.

We restricted this study to transient tactile events that can occur on a generic 5×5 mm patch of surface pressure array to ensure the problem remains computationally tractable. Since the spatial resolution of human fingertips are between 2 to 4 mm [190], it is reasonable to assume that we only have to resolve a single edge feature within this region under most interaction scenarios. Moreover, the sensor surface is assumed to be flat due to the small region of interest. This reduces the problem to one of analyzing object indentations, where an edge is depressed onto a flat surface. Taking advantage of symmetry properties to reduce computational complexity, the model was constructed in 2D. Fig 5.2a illustrates the simulation framework.

Table 5.1: Experimental variables for FEM simulation.

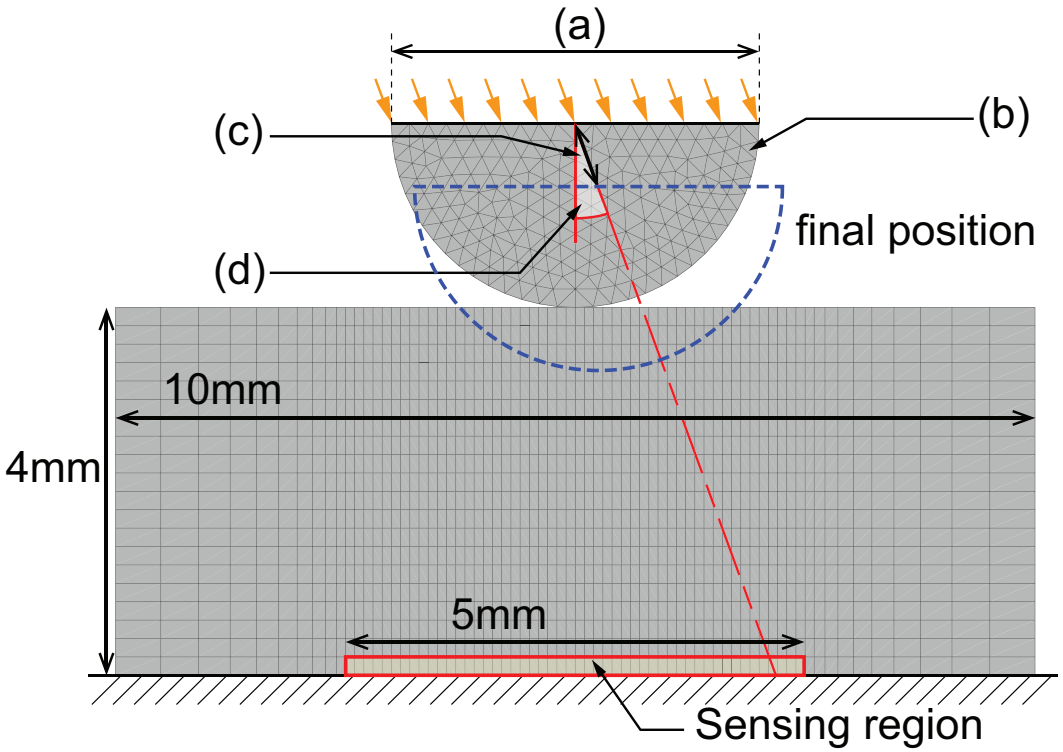
	Variable	Values (increment)	Unit
a	Indentor diameter	2 – 4 (0.5)	mm
b	Young modulus of indentor	50 – 250 (100)	kPa
c	Indentation depth	0.5 – 2 (0.5)	mm
d	Indentation angle	0 – 20 (5)	degrees

Table 5.2: Sensor material constants (3 term Yeoh hyperelastic model).

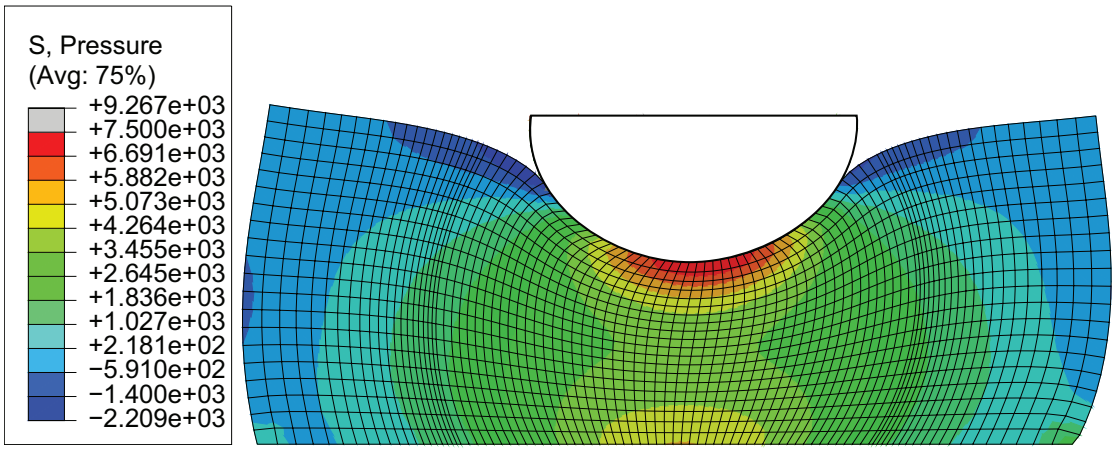
Material type	C_{10} (Pa)	C_{20} (Pa)	C_{30} (Pa)	SSE
Dragon Skin 10®	3600	25.8	−0.056	0.12

To faithfully and accurately synthesize interaction scenarios, we selected several contact parameters to vary (see Table 5.1). The indentor was modeled to have linear elastic properties, while the sensor was modeled using the parameters listed in Table 5.2. These conditions accurately simulate Dragon Skin® 10 (Smooth-on, USA) silicone rubber commonly used as a cover for soft tactile sensors [191].

For a given parameter set the result from a FEM simulation is deterministic. To assess the robustness of our approach we simulated stochastic variations due to limited sensor resolutions using a sub-sampling technique. A sub-sampling grid was constructed where the horizontal and vertical spacing of sample points corresponded to the temporal and spatial resolution of the simulated sensor array, respectively (Fig. 5.3). Each sub-



(a) FEA simulation of indentation by an edge represented by a half-circle. Experimental variables (a) – (d) are described in Table 5.1.



(b) Example of an asymmetric pressure deformation model with a 10 deg contact event. Colored contours represent pressure distributions.

Figure 5.2: FEA simulation of an indentation resulting from a contact event.

sampled output was generated by translating the sub-sampling grid within the FEM output, thus ensuring that while individual sampling points are spaced in accordance to the sampling period and spatial resolution of the simulated sensor array, the absolute time and location of the sampling points are different with each variation. The grid translations in the temporal and spatial axes were uniform random in nature to create outputs that are stochastic and suitable for classification and mutual information analysis.

The FEA experiments were conducted on ABAQUS 6.13-1. The contact event was the indentation of a half-circle onto a rectangular substrate at a constant velocity (Fig. 5.2b). Displacement control was used, with full displacement achieved in 100 ms. The friction coefficient between the indentor and substrate was fixed at 0.7. Generic, linear quadrilateral elements of type CPS4 were used for both the indentor and substrate. All outputs were recorded at 0.2 ms time steps and 0.1 mm spatial resolution, from a sensing region of 5 mm width. Data from the base sensing region was recorded and sub-sampled to 25 elements to be compatible with the density of mechanoreceptors [4]. The spline interpolation technique in MATLAB [192] was used to estimate values between discrete points of an FEM output. Five-hundred variations were generated per experimental condition, thus resulting in a dataset of 150000 samples.

5.4.2 Transduction

Pressure was used as the input parameter to the computational model since it well describes the transduction of tactile sensors [193]. We extracted pressure estimates from elements of the same depth within the sensing region and mapped them to a logarithmic scale. The logarithmic mapping approximates the response curves of mechanoreceptors [23] as well as piezoresistive transducers commonly used in artificial tactile sensors [194]. The logarithmic conversion is described as

$$\rho_t = u_t^{\alpha} \tag{5.1}$$

where u_t is the pressure at time t and α is the log compression power.

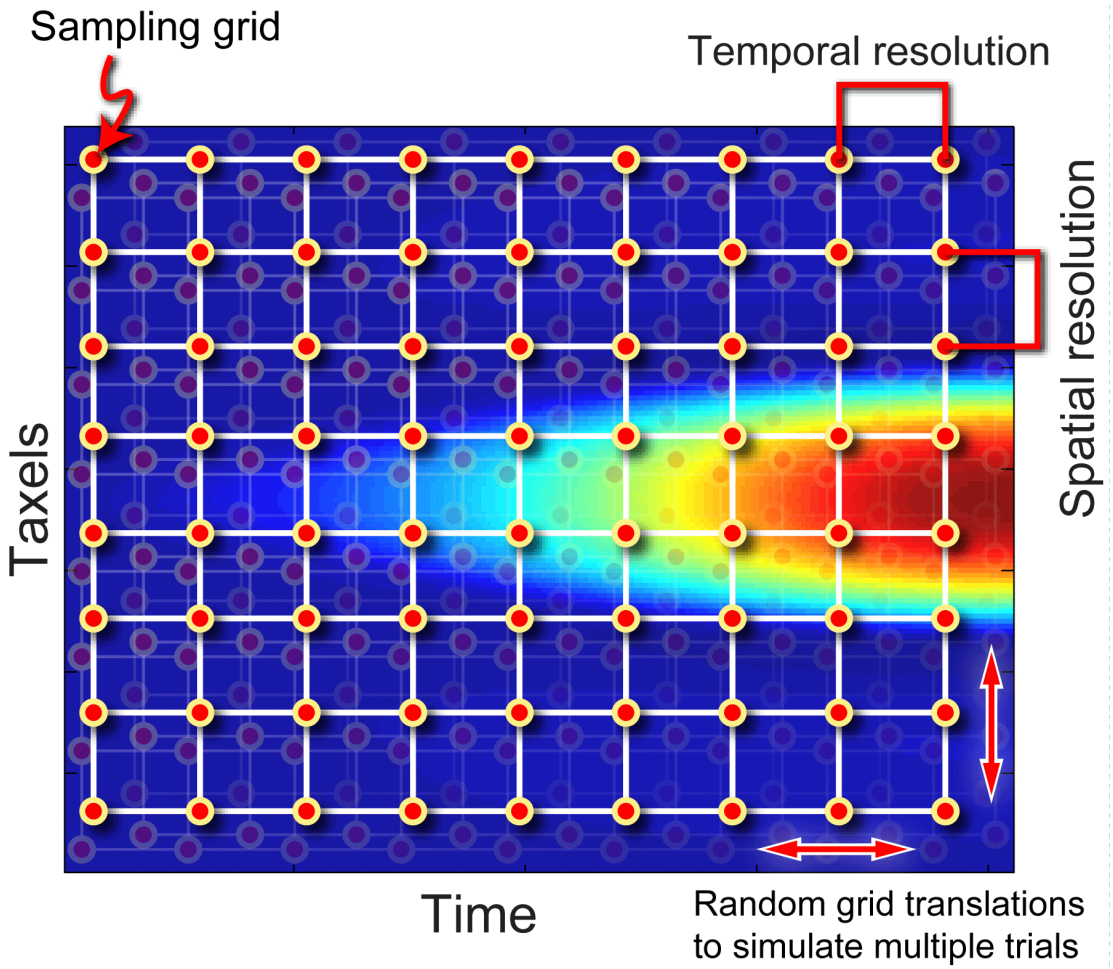


Figure 5.3: Sub-sampling procedure to obtain stochastic variations from a FEM output. Red arrows and multiple translucent grids illustrate the multiple sub-samples obtained by grid translation. This technique simulates experimental error due to limited spatiotemporal resolution of a sensor.

To analyze the information captured by spatiotemporal spike representation, a conversion from analogue pressure signals to spikes is needed. This transformation was achieved using a phenomenological FA-1 afferent model. In most transduction models of FA-1 mechanoreceptors, the first derivative of pressure is the primary input component to a leaky integrate-and-fire neuron model [195, 196]. Therefore, we defined the input current $I(t)$ to the neuron model at each time instance t as the difference in pressure

$$I(t) = \rho_t - \rho_{t-1} \tag{5.2}$$

where ρ_t is pressure at time t after logarithmic conversion (Eq. 5.1). Neuron dynamics are described by its membrane potential U, which is an integration of the input currents since the last spike t_{spk}

$$U(t) = \sum_{t=t_{spk}}^{t} I(t). \tag{5.3}$$

A spike is discharged when the membrane potential exceeds a threshold $\theta(t)$. In this work, we introduce polarity to the output spikes, with

$$s(t) = \begin{cases} 1 & \text{if } U(t) > \theta(t) \\ -1 & \text{if } U(t) < -\theta(t) \\ 0 & \text{otherwise.} \end{cases} \tag{5.4}$$

Upon spike emittance, $U(t)$ is reset to 0 mV and the threshold is increased by an amount proportional to the sampling period of the system. This property increases the difficulty of a consecutive spike and simulates a refractory period of the mechanoreceptor. The increased threshold slowly decays to a baseline value θ_0 when there are no spiking activities. The dynamics of the threshold is described as

$$\theta(t) = \begin{cases} \theta(t-1) + TA\theta_0 & \text{after a spike} \\ \theta(t-1) - B\theta_0 & \text{if } \theta(t-1) > \theta_0 \\ \theta_0 & \text{otherwise} \end{cases} \tag{5.5}$$

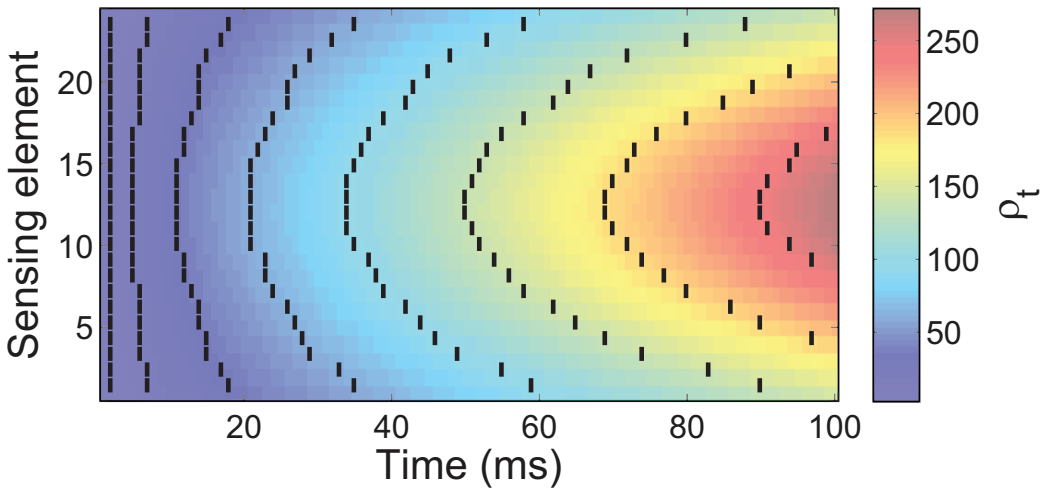


Figure 5.4: A composite illustration of spike conversion using simulated data. The background color gradient represents pressure values (ρ_t) after log compression. Solid black lines represent spike discharges by our model. Spikes are discharged only when a pressure change above $\theta(t)$ is sensed.

where A and B are constants that control the rate at which the threshold potentiation and decay occur respectively and T is the sampling period of the system in milliseconds. Fig. 5.4 illustrates spike conversion on a typical input signal.

5.4.3 Analysis

Information theory is widely used in neural systems to understand the statistical details embedded in spike trains. The information encoded in neural recordings are measured by the ability to discriminate various stimuli based on their spike train response. We used linear discriminant analysis (LDA) to classify the outputs, where inputs to the classifier are spike raster plots of size $m \times n$, representing outputs from m mechanoreceptors lasting n time steps (where $n = 100/\text{sampling period (ms)}$). Shannon's information measure [197] was applied to compute the mutual information $I(R;L)$ between the ground truth label (L) and classifier response (R):

$$I(R;L) = \sum_{R,L} p(R|L) \cdot p(L) \log_2 \frac{p(R|L)}{p(R)} \tag{5.6}$$

where $p(R|L)$ is the confusion matrix of the classifier. This approach presents a more objective performance measure since it provides information about the error distribution

(non-diagonal values in the confusion matrix), not just the number of hits (diagonal values). Results were computed using the information breakdown toolbox [198] implementing the Panzeri-Treves method [199] for bias correction.

For all experiments the constants described in Table 5.3 were used. The value of θ_0 was estimated by dividing the range of pressures by 1024, the smallest pressure change that can be resolved by a 10 bit ADC. Values for the logarithmic compression exponent of biological mechanoreceptors have been reported to range between 0.2 to 1.4 with a mean of 0.7 [10]. The value of α was computed using a grid search from 0.2 to 1.4 with an increment of 0.1. The values for A and B were obtained heuristically to ensure that the model's spike rate is biologically plausible at around 100 Hz.

Table 5.3: Constants used for simulation.

Constant	Symbol	Value
Log compression	α	0.7
Baseline threshold	θ_0	0
Threshold potentiation	A	20
Threshold decay	B	0.05
Trials per simulation	K	500

5.5 Simulation results

This section presents our findings from the FEM simulations with an emphasis on how information may be lost due to reduced temporal resolution.

5.5.1 Spatiotemporal features embed information at time scales much shorter than contact duration

The synthetic data was classified based on the combination of stimulus parameters (total of 300 classes). The results in Fig. 5.5 illustrate that mutual information decreases significantly along with the sampling rate. Although the duration of stimulus presentation is significantly longer at 100 ms, we observed a large discrepancy in mutual information between sampling with 0.5 ms and 20 ms periods. This difference in performance indicates that while a sampling period of 20 ms is within Nyquist limits for detecting the

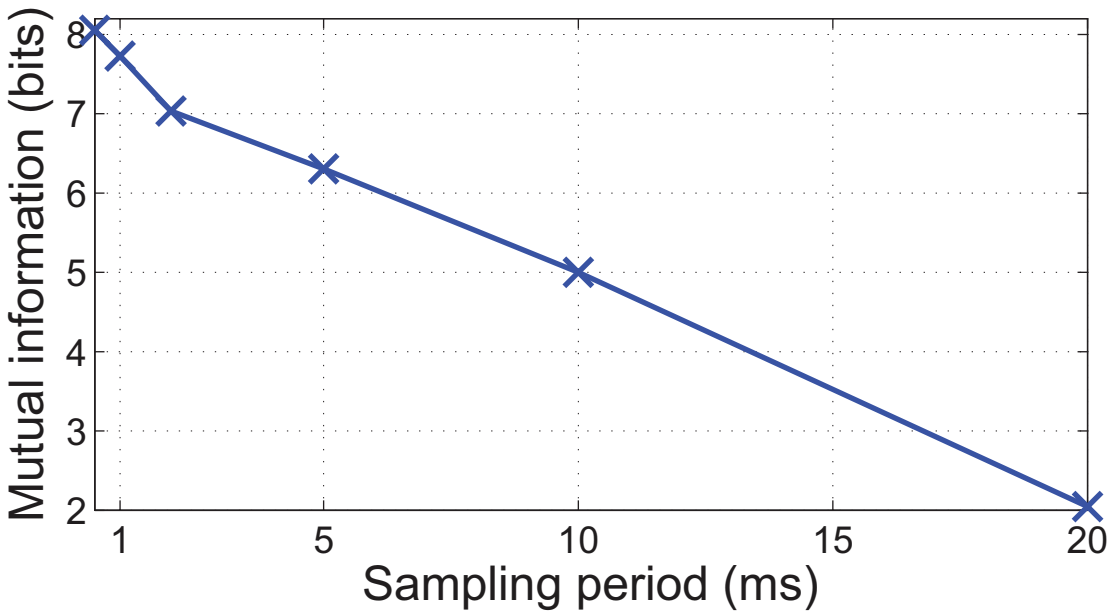


Figure 5.5: Mutual information versus sampling period for 300 simulated contact scenarios. We observed that mutual information decreases when sampling periods are increased, indicating a loss of stimulus discriminability as the temporal precision of the raw signals is reduced.

indentation event, the spatiotemporal features needed for fine discrimination is contained at time scales notably shorter than the duration of a contact event.

In a realistic setting, it is often necessary to distinguish stimuli based on a single parameter, regardless of how other properties may have changed. For instance, the hardness of objects could be identified regardless of its shape. Here we examine how the reduction in temporal precision affects the discrimination of individual contact parameters. This is accomplished by partitioning all 150000 samples into 4 classes when classifying based on slant angle/indentation depth/indentor diameter or 3 classes when classifying for hardness (Young modulus).

The results were normalized to the maximum mutual information ($\log_2$(# of classes)) for comparison. Fig. 5.6 illustrates the results of this study. As observed earlier, mutual information decreased with sampling period for most parameters. However, the effects of a decreased sampling period are most noticeable for discriminating stimuli of different indentor radii. In contrast, differences in indentation depth do not require a high sampling rate for robust discrimination.

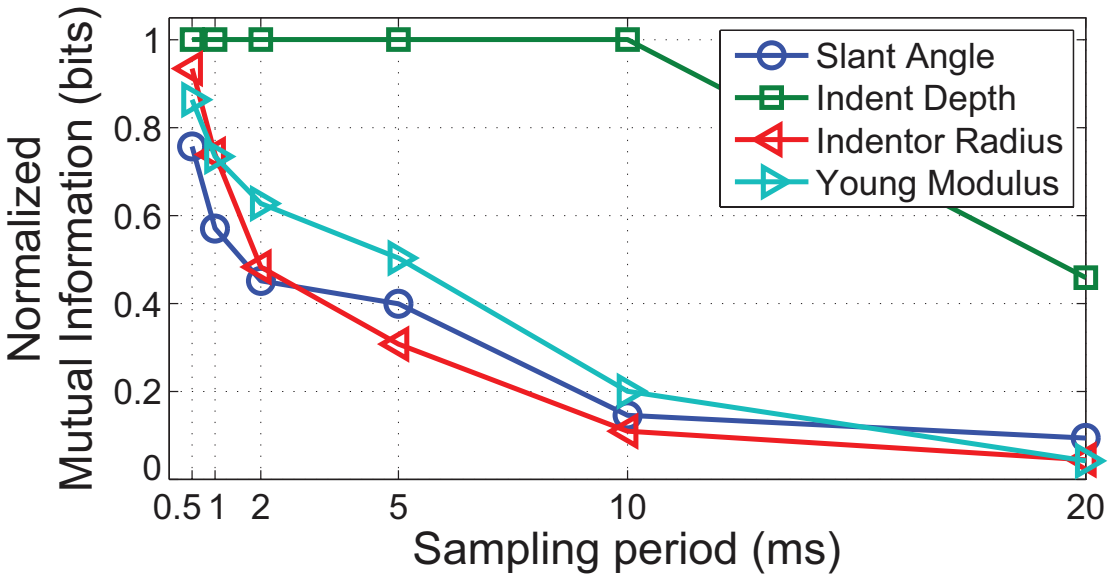


Figure 5.6: Normalized mutual information versus sampling period for each parameter varied, based on 300 simulated contact scenarios. We noticed that discrimination of indentation depth is least affected by increased sampling period, while classification of indentor radius is substantially more difficult without sufficient temporal precision. Note the logarithmic scale on the x axis.

5.5.2 Information loss due to low sampling rate cannot be recovered by increasing pressure resolution

While decreasing the temporal precision has been shown to lower mutual information, the additional communication bandwidth afforded by a lower sampling rate would allow each sample to encode pressure information in greater detail. It is thus necessary to investigate the optimal tradeoff between sampling rate and pressure information encoded, given a fixed readout data rate.

We modified the FA-1 model in Eq. 5.4, allowing for spikes that also indicate the magnitude of pressure change. The dynamics of the updated model $\tilde{s}(t)$ are described as a quantization of $U(t)$ scaled by θ_0

$$\tilde{s}(t) = \left\lfloor \frac{U(t)}{\theta_0} \right\rfloor . \tag{5.7}$$

Modifying the output as given in Eq. 5.7 preserves information about pressure changes while maintaining the spiking nature of the output. Depending on the number of output bits available, $\tilde{s}(t)$ may saturate at $2^{\text{bits}} - 1$.

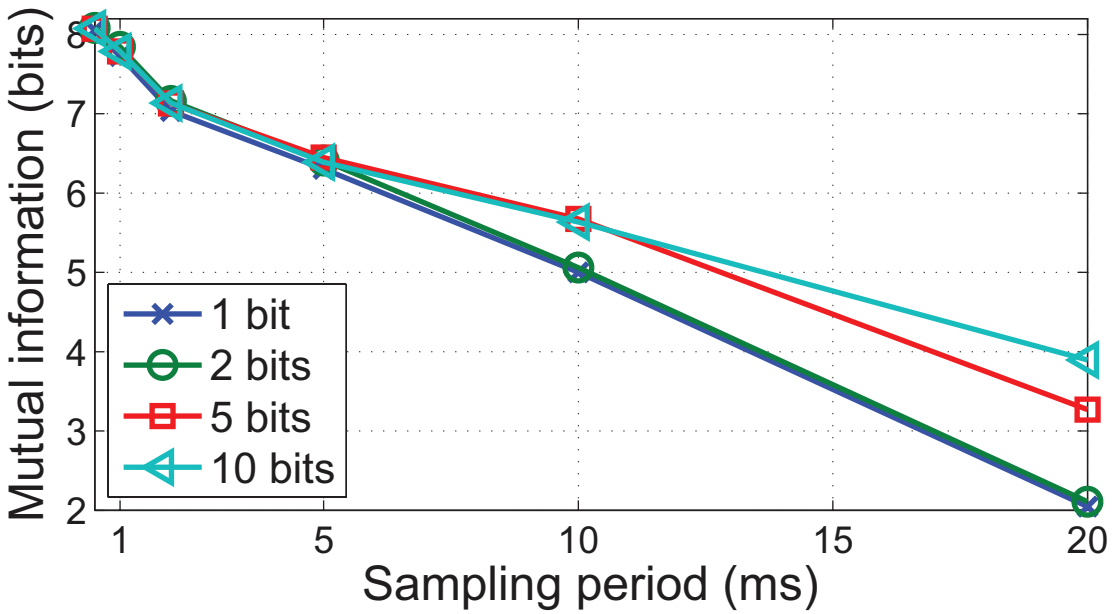


Figure 5.7: Mutual information versus sampling period, as a function of the number of bits per event. Classification was based on 300 simulated contact scenarios. Note how a higher bits-per- sample improves mutual information only for larger sampling periods.

The results exhibited in Fig. 5.7 illustrate that increasing pressure resolution leads to higher mutual information. This effect is most discernible at lower sampling rates where changes in pressure intensity between consecutive samples are more abrupt. However, the improvement is less significant at higher temporal fidelity. For example, sampling with a period of 20 ms with 10 bits of pressure resolution achieves a much lower performance compared to sampling at 2 ms period with 1 bit pressure resolution, despite producing data at the same rate. These results demonstrate that at least for dynamic edge indentation stimuli, key features for discrimination are predominantly of much higher frequency than the contact duration suggests, and it is thus prudent to allocate more resources to encode temporal dynamics than pressure intensity.

5.5.3 Precise temporal structure increases classification speed

Accurate classification is beneficial but has little advantage if higher performance is achievable only after the entire stimulus is recorded. Here we process samples truncated in time to investigate information content with respect to the duration of stimulus recorded. The results depicted in Fig. 5.8 show that additional information at fine

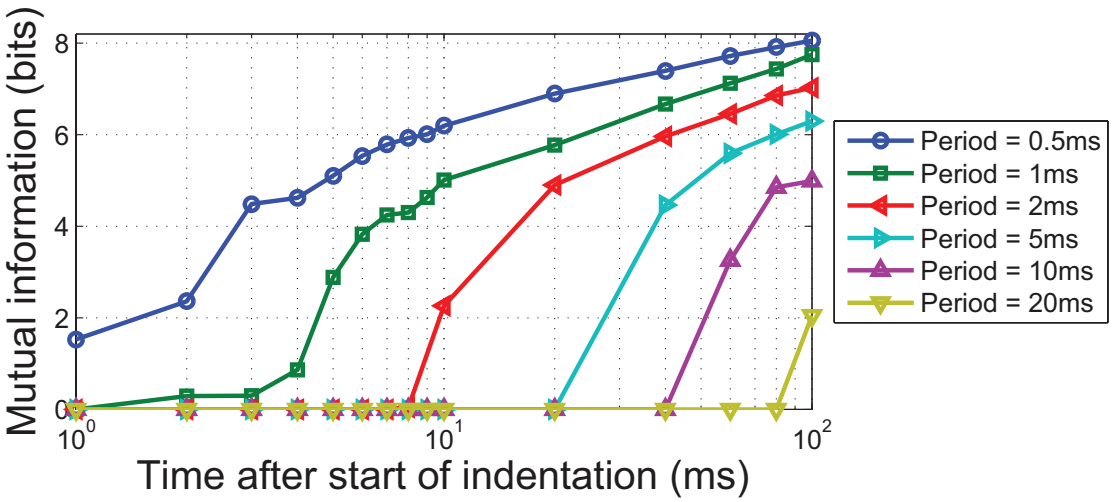


Figure 5.8: Mutual information obtained at different time-points of an indentation. Each trace corresponds to a particular sampling period used. Classification was performed on simulated contact scenarios. We observed that mutual information increases earlier for a shorter sampling period (higher temporal resolution) and reached a higher value at the end of the stimulus (100 ms). Note the logarithmic scale on the x-axis.

temporal scales exists at the early stages of indentation. For example, over 6 bits of mutual information is available in a 10 ms recording when sampling with a period of 0.5 ms, while similar performance can only be achieved in an 80 ms recording if a sampling period of 5 ms was used. This feature is an important advantage when working with event-driven sensory control algorithms, since it has the potential to enable critical contact events to be identified quickly [200].

5.6 Discrimination of impact using a high speed tactile sensor array

Results from the FEM simulations demonstrate that critical contact information is embedded in the spatiotemporal dynamics of the tactile sensor response. With high temporal resolution, discrimination of contact stimulus can be achieved accurately and quickly. In this section, we provide details for a set of physical experiments to assess the predictions from FEM simulations.

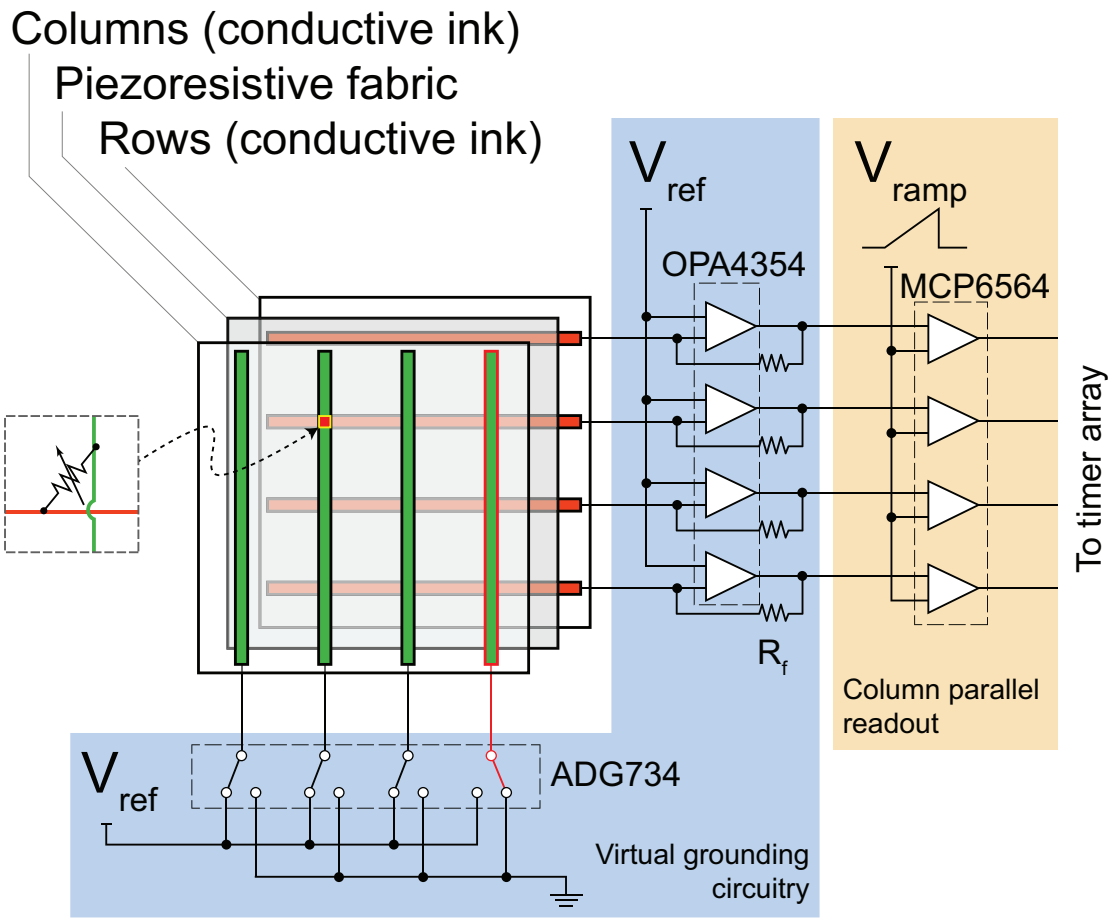


Figure 5.9: Sensor architecture and readout circuitry. Blue region: circuitry for cross-talk cancellation. Red region: Circuitry for analog to digital conversion.

5.6.1 The Kilotaxel-Kilohertz tactile sensor array

We previously reported a high speed tactile sensing platform to investigate the use of spatiotemporal coding in touch [174]. It has 4096 piezoresistive sensing elements arranged in a 64×64 grid. Using a row-parallel column readout technique a very high sampling rate of 5.2 kHz was achieved. The sensor elements were made by inkjet printing of conductive traces as rows and columns on separate sheets of polyethylene (AgIC Inc., Tokyo, Japan). The traces were then arranged orthogonally, with a piezoresistive fabric (LR-SLPA, Eeonyx, CA) sandwiched in between, where each intersection formed a sensing element (Fig. 5.9). The size of each element measured 2.29 × 2.29 mm and the entire active region covered an area of 17 × 17 cm. The sensor array was covered by a 10 mm layer of polyurethane foam to provide the mechanical compliance needed to capture spatiotemporal patterns.

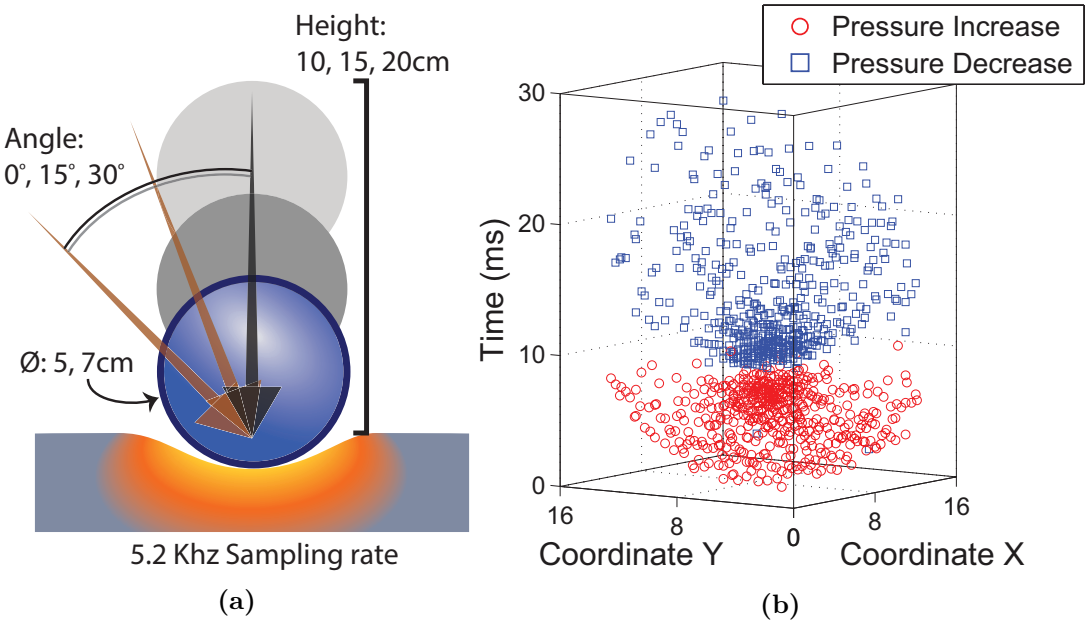


Figure 5.10: Example physical experiment. (a) Ball dropped on sensor substrate. (b) Spatiotemporal events captured by sensor array.

Sampling was controlled by a field-programmable-gate-array (FPGA) from Xilinx (XC6SLX45). A/D conversion was performed using a single-slope comparator array and multiple parallel timers (Fig. 5.9). The FPGA was programmed to produce a spike event whenever a significant change in pressure ($\succeq 0.02\,\mathrm{N}$) was detected. Each output event indicates the time, coordinates, and the magnitude of the pressure change.

5.6.2 Experiment

To validate results observed in the FEM simulations, we performed a series of indentation tests on the sensor array. Rigid 3D printed spheres of size 5 and 7 cm in diameter were released from heights of 10, 15 and 20 cm vertically above the sensing surface. The experiments were repeated with the sensor at a slant of 0, 15 and 30 degrees, resulting in a total of 18 different combinations of contact parameters. For each parameter combination 100 trials were performed. The weight of all spheres was calibrated to be similar ($36 \pm 1\,\mathrm{g}$).

We extracted a region of interest consisting of 16×16 elements centered around the averaged spatial coordinates of the elements that responded during the first sampling period upon impact. The duration of each recording was standardized to 50 ms, which

is long enough to capture the first impact but not subsequent bounces. Fig. 5.10b encapsulates data recorded from a representative impact stimulus from a 5 cm sphere dropped from a height of 10 cm with a 0° slant angle.

5.6.3 Analysis

Sampling the sensors at a high rate yields a sparse output signal with high dimensionality. Classifying the spike raster plots using LDA is computationally intensive and, hence, impractical. Therefore, we adopted a metric space approach commonly applied in neuroscience to compute mutual information [201]. Pairwise distances between trials were first computed using a modified Van Rossum distance metric [202]. Based on the calculated distances, a K-nearest neighbour clustering technique was implemented to classify the data and mutual information was computed from the classification results (See Eq. 5.6).

The Van Rossum distance metric was originally developed to measure the difference of two spike trains and can be summarized as follows [202]. Each spike is convolved with an exponential kernel to obtain a continuous signal

$$f(t) = \sum_{i=1}^{I} H(t - t_i) \exp\left(-\frac{t - t_i}{\tau}\right) \tag{5.8}$$

where:

$$H(x) = \begin{cases} 0 \text{ if } x < 0 \\ 1 \text{ if } x \geq 0 \end{cases} \tag{5.9}$$

and $t \in \mathbb{R}^+$ is time, t_i is the time of the ith spike, $H(\cdot)$ is the Heaviside step function and τ is a free parameter that affects temporal smoothing of the signal. The distance metric is defined as the Euclidean norm on the function space

$$\|\mathbf{f}_{(a)}, \mathbf{f}_{(b)}\|_2 = \sum_{t=0}^{T} \sqrt{(f_{(a)}(t) - f_{(b)}(t))^2}. \tag{5.10}$$

In this work we extended the computation to patterns involving multiple spike trains by applying a spatial kernel to the continuous signal $f(t)$. The spatial smoothing procedure

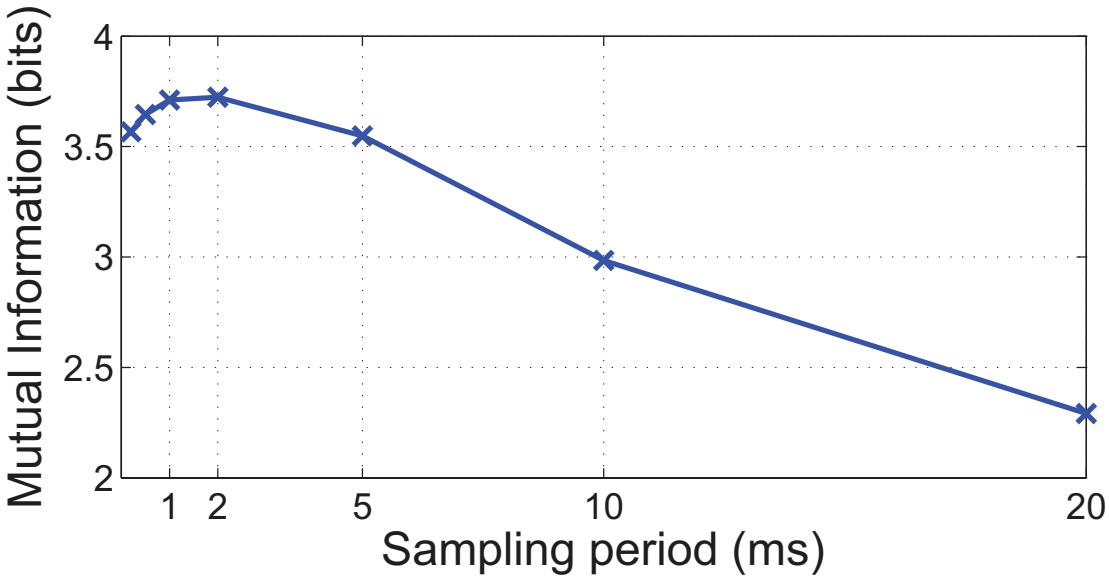


Figure 5.11: Mutual information versus sampling period, computed using experimental data from 18 classes of contact scenarios. Notice the significant decrease in discriminability when the sampling period is above 5 ms.

for spike train indexed m for a spatiotemporal pattern with N spike trains is expressed as

$$g_m(t) = f_m(t) + \sum_{n=1}^{N} \left(f_n(t) \cdot \exp\left(\frac{-d_{m,n}}{\sigma} \right) \right) \tag{5.11}$$

where $d_{m,n}$ is the physical distance between sensor elements indexed m and n and σ is the spatial smoothing parameter.

The distance between two spatiotemporal patterns indexed a and b with N spike trains is calculated as

$$\|\mathbf{g}_{(a)}, \mathbf{g}_{(b)}\|_2 = \sum_{n=1}^{N} \sqrt{(g_{n(a)}(t) - g_{n(b)}(t)}. \tag{5.12}$$

In this work, we used $\tau = 0.4$ ms and $\sigma = 2$ mm because they yield the highest mutual information after a grid search of $\tau \in [0.1, 2]$ ms at intervals of 0.1 ms and $\sigma \in [1, 5]$ mm at intervals of 1 mm.

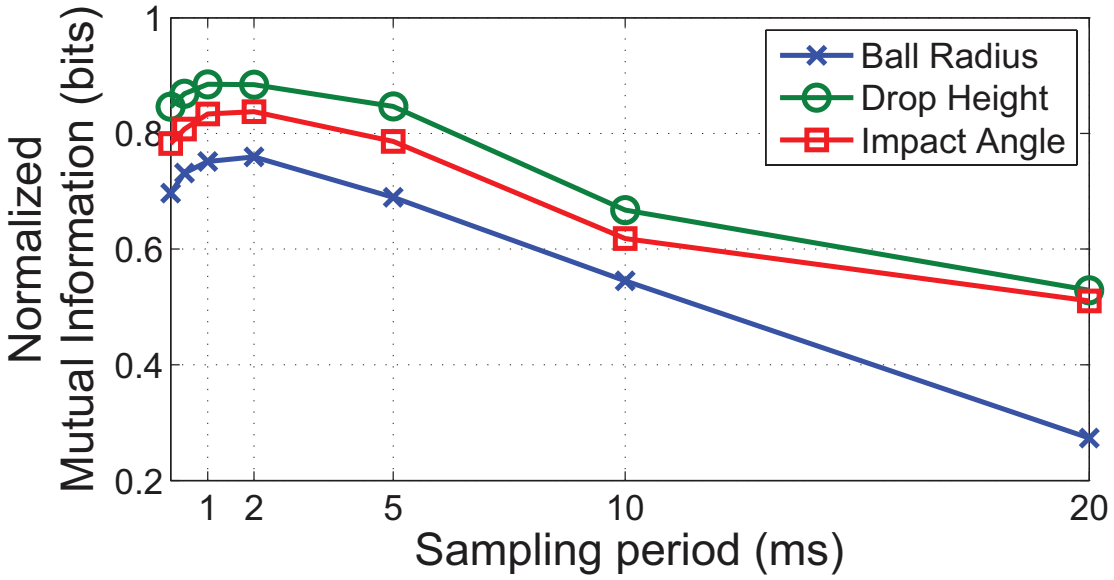


Figure 5.12: Normalized mutual information versus sampling period for specific contact parameters of the physical experiment. It is noticeably harder to distinguish impacts from different ball radii as sampling period increases.

5.6.4 Results

Maximum information was obtained when sampling at 1 ms periods (Fig. 5.11). As demonstrated with FEM simulations, mutual information generally decreases with sampling rate. However, a sampling period below 1 ms also led to a decrease in mutual information. This may be due to increased high frequency noise when sampling at faster rates, leading to a reduction in spatiotemporal pattern consistency.

Partitioning results based on contact parameter, we observed that discrimination of ball radius was most sensitive to sampling rate (Fig. 5.12), in concurrence with our FEM simulations. Since the extracted region is less than the full contact area, the size of the ball can only be distinguished based on small differences in spike times, an effect of variations in local curvature.

A high sampling rate also permits accurate discrimination to be performed with reduced latency (Fig. 5.13). In less than 10 ms after impact, mutual information almost plateaued for sampling periods less than or equal to 2 ms. Slower sampling rates took at least twice as long to achieve comparable performance.

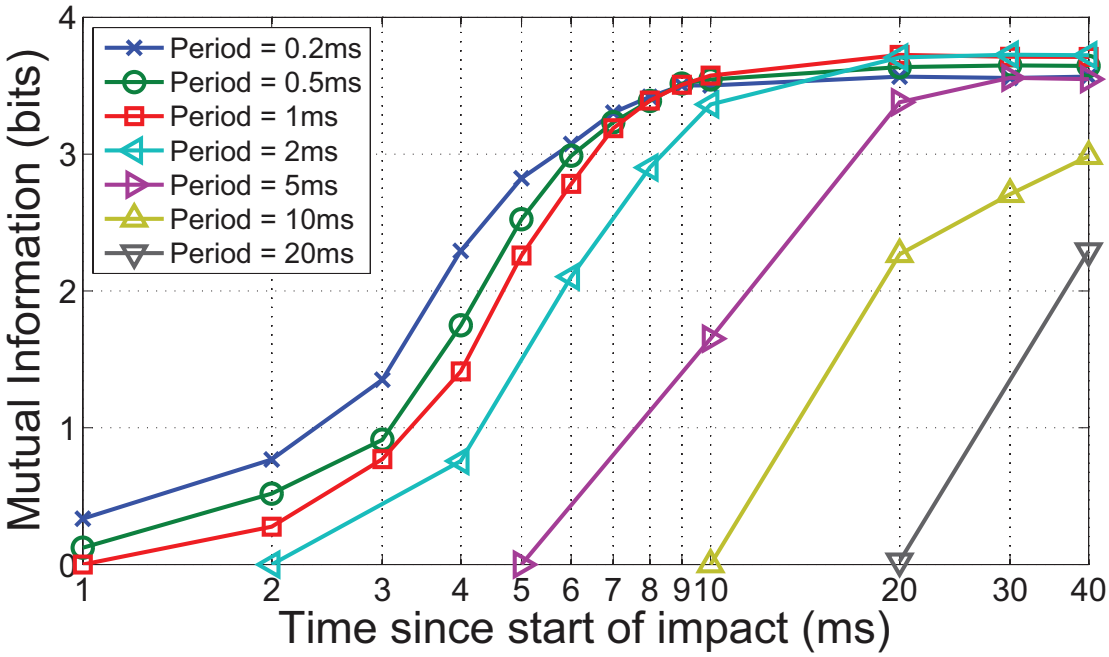


Figure 5.13: Mutual information as function of time-elapsed since start of impact stimulus. Each trace corresponds to a different sampling period used. High sampling rates allow greater mutual information in less time. Results describe the ability to distinguish among the 18 contact scenarios of the physical experiment. Note the logarithmic x-axis.

5.7 Event-based representation reduces bandwidth utilization

Sampling rates of 1 kHz and above are not practical when interfacing with thousands of sensor elements using conventional frame-based representations. Here, we describe how a bio-inspired event-based representation can reduce the amount of data generated, allowing for rapid sampling rates needed to capture spatiotemporal features. Therefore, we present a comparison of three encoding strategies.

1. Event-based change coding (EC): Event packets are generated only when a significant difference in pressure is detected. Each event packet consists of 9 bits, with 8 bits to encode the address of the element and 1 bit to indicate an increase or decrease in pressure.

2. Frame-based intensity code (FI): Pressure at each element is represented with n bits per sample. For this analysis, we used $n = 10$.

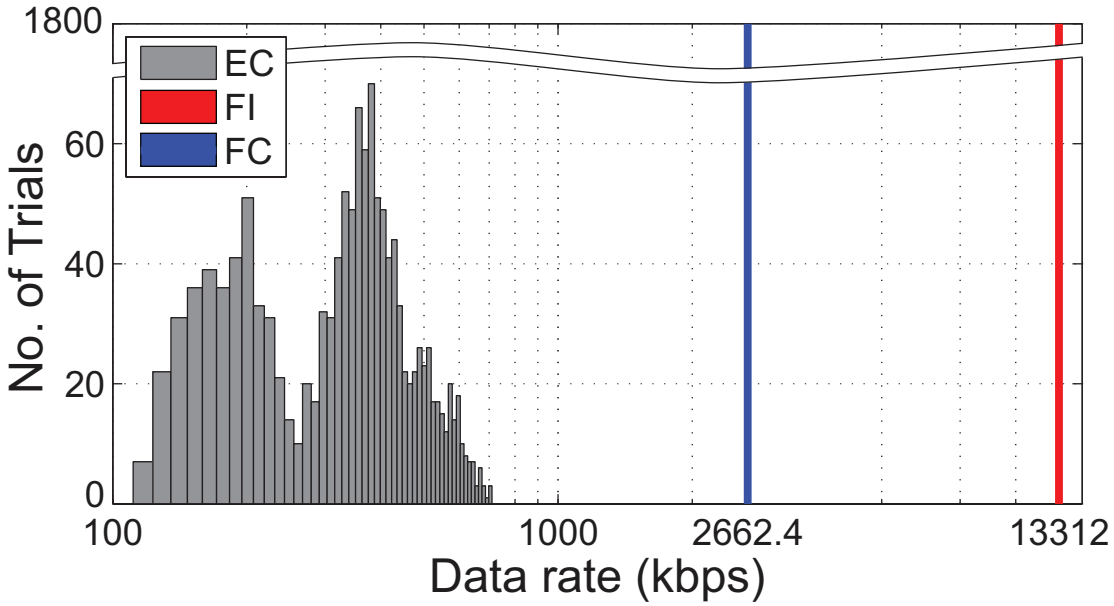


Figure 5.14: Comparison of output data-rate among the 3 coding strategies. Event-based change (EC) code is activity dependent, hence its distributed nature. Both frame-based change and intensity (FC and FI) codes produce data at a constant rate. The statistics were compiled from all 1800 recordings of the physical experiment. Note the logarithmic scale on the x-axis.

3. Frame-based change code (FC): Only 2 bits were used per element to indicate whether an increase, decrease, or no change in pressure was observed.

Note that the 8 bit address space for the EC method was selected because it is the minimum number of bits required to encode a 16×16 element region of interest. Although a larger number of sensor elements require more bits to address, it is also likely more sparse because of a lower likelihood of activating all elements at once. The duration of an impact was determined as the time between the first and last event recorded during contact. Data rates for frame-based (FI and FC) approaches were computed at 5.2 kHz frame rates to match the temporal resolution of the event-based (EC) approach.

Fig. 5.14 illustrates the distribution of data-rates for all recorded impacts. An event-based (EC) code produces data at a rate that is dependent on activity level. With an EC approach, only a maximum of 659 kilo-bits-per-second (kbps) was required to fully capture the spatiotemporal events. In contrast, the frame-based codes (FI and FC) required a constant 13.3 Mbps and 2.66 Mbps respectively.

5.8 Discussion

Neurophysiologists have emphasized the benefits of spatiotemporal spike patterns in tactile processing [38,42], although few sensor and robotic system developers have taken this approach. The lack of adoption by the robotics community is not due to an absence of enthusiasm but because tactile sensors have traditionally been designed for accurate pressure measurements at specific points [175]. A paradigm shift of methods for sensory acquisition and processing may be needed to utilize temporal structures efficiently.

Our work in this chapter focused on precise detection of spatial and temporal features of pressure changes across a large population of taxels. Through FEM analysis and physical experiments, spatiotemporal patterns of pressure changes were used to discriminate contact scenarios involving multiple combinations of stimulus parameters such as local curvature, indentation magnitude, angle and indentor hardness. We observed that signals with temporal resolution much higher than the duration of contact was needed to resolve the contact scenarios. Loss of information from reduced sampling rates could not be recovered by increasing pressure resolution. It is most probable that the key features for discrimination of dynamic contact are predominantly high frequency, and our physical experiments suggests that a tactile sensing system with sampling rates of 500 Hz and above is needed to fully exploit these features.

We advocate the use of event-based representation of pressure changes to minimize output data-rate. Implemented on our high speed sensor array, we demonstrated the ability to sample from thousands of sensing elements while preserving the temporal structure of response patterns with sub-millisecond precision. We also observed an approximately 20 time reduction in data transmitted during ball-drop experiments when compared to conventional frame-based representations. The savings incurred for high density sensors covering large surface areas would likely be much higher using event-based coding since it is unlikely that changes will be detected for all elements at every sampling period.

Realistic tactile stimuli are highly complex. We do not contend that our simulations and experiments have comparable diversity. We restricted our investigation to transient dynamic stimuli since these are some of the most challenging problems in tactile sensing.

In addition, for robots to operate safely in natural dynamic environments, the ability to react quickly to complex transient events is crucial.

A comparison of the results obtained from physical experiments to FEM simulations reveals less than predicted improvement in performance when sampling with periods below 2 ms (Fig. 5.11). This might be a consequence of stochastic error introduced by nature of the experimental trials. Sources of inconsistencies include the varying response characteristics of sensing elements and manual stimuli delivery. Both introduce large intra-class variations in the spatiotemporal patterns in comparison to the underlying deterministic FEM simulations. These inconsistencies are more noticeable when sampling at high rates and reduce discriminatory performance. In addition, the mutual information captured at lower sampling rates are likely to be positively biased because the start of the impacts were aligned with 0.2 ms precision provided by the native sampling rate of the sensor. These issues may be resolved using advanced learning algorithms that are adaptable to the characteristics of sensors and robust to experimental noise.

One of the goals of this work was to highlight the advantages of coding tactile stimuli through spatiotemporal representations, which FA-1 afferents are well suited for. Although FA-1 like representations have been demonstrated to work well under dynamic conditions, we do not diminish the importance of other mechanoreceptors; namely, the SA-1, FA-II and SA-II, and the value of accurate pressure measurements. Indeed, an efficient processing strategy could be implemented where the saliency of tactile signals are characterized rapidly based on the temporal structure to prime higher order attentional mechanisms. Processing resources can then be allocated to analyze the stimulus in greater detail as spikes from SA type mechanoreceptors are accumulated over time.

Beyond the tactile domain, neuromorphic event-based sensory processing has been implemented in vision and audition [12]. Research in spike based processing in these fields are more established, partly due to the availability of silicon retinae and cochleas [203, 204]. Dedicated processing hardware and algorithms have also been developed to interface with spatiotemporal features captured by such sensors [12]. Due to its distributed nature and challenging requirements to be mechanically flexible and stretchable,

building an inherently event-based tactile sensor array is significantly more challenging. Recent progress in material science and manufacturing technology have opened new opportunities such as using organic transistors to create digital mechanoreceptors [181]. Such innovative components and applications will hopefully increase interests and development in tactile sensing.

5.9 Conclusion

In this chapter, we have shown that temporal dynamics of tactile signals play a critical role to discriminate indentation stimuli. Using both FEM simulations and physical experiments, we demonstrated that conventional tactile sensor arrays with sampling rates of 100 Hz and below lack the temporal fidelity needed to detect and classify transient contact events. Implementing a neuromorphic event-based communication technique substantially reduces the communication bottleneck, enabling dynamic contact stimuli to be captured with exquisite spatiotemporal detail and, hence, achieving significantly higher discrimination performance. Our spatiotemporal coding strategy has the potential to play a significant role to enable responsive robotics and prosthetics equipped with full body tactile sensors.

The next chapter presents an efficient algorithm that trains spiking neurons to recognize spatiotemporal spike patterns, thus rendering this research relevant to real-life applications.

6

Learning Spike Patterns through Convex Optimization

W. W. Lee, S. L. Kukreja, N. V. Thakor, "CONE: Convex-Optimized-Synaptic Efficacies for Temporally Precise Spike Mapping" in *IEEE Transactions on Neural Networks and Learning Systems* , vol.PP, no.99, pp.1-13

Contents

6.1 Preface

Through earlier chapters, we have shown how tactile signals may be encoded through spatiotemporal spike patterns to efficiently represent dynamic touch events. This chapter compliments our works by introducing a novel technique to process and respond to the spatiotemporal representations. When compared to existing tools, we found our method significantly more efficient to train while yielding results that are more robust.

Author Contributions and Acknowledgments

Lee W. W. conceptualized and developed the algorithm, conducted the experiments, analyzed the data and generated the figures. Dr. Thakor N. V., Dr. Kukreja S. L. and Lee W. W. wrote the manuscript. The authors are grateful to Kyung-Ryoul Mun and Chan Chow Khuen for providing assistance and equipment for gait analysis.

6.2 Introduction

Neurons in most vertebrate nervous systems communicate through action potentials. These are abrupt moments where the membrane voltage of a neuron rapidly rises and falls in a consistent trajectory. As the trajectory of the membrane voltage is independent of the amount of current that induced the response, an action potential or spike is typically considered an all or nothing event [205]. It is traditionally assumed that information is conveyed through the average spiking rate of a neuron [206]. However, recent studies have found precise temporal relationships in spike patterns evoked by sensory stimuli in visual, auditory, olfactory and somatosensory pathways [11], suggesting that information is also encoded in the time of each spike. Temporal spike codes such as rank-order or first-spike latency codes are capable of propagating information much faster than rate codes because they typically require only one spike per neuron [207]. Therefore, the computational capabilities of individual neurons would be much higher than previously thought, if temporal features are taken into consideration [208].

These key insights are the fundamental basis for the field of Spiking Neural Networks (SNN). SNNs are synthetic networks of neurons that emulate the rich time-varying dynamics found in biology for increased realism [13, 145]. SNNs are shown to be

superior to rate-based neural network models for complex tasks such as recognition of visual patterns, odors, sound properties and tactile stimuli [106, 123]. Due to their biological credibility, SNNs are useful for modeling information processes in the brain and investigating various learning mechanisms [126, 127]. Engineering problems such as pattern recognition and real-time computation also benefit from the use of SNNs [129, 131].

The neurons in a SNN are linked together through synapses. Each higher order neuron may receive inputs from multiple lower order neurons through synapses of adjustable weights or efficacies. Like their biological counterparts, computational neurons models in SNNs communicate through spikes, which are discrete events in time. For SNNs with multiple layers, the ability for neurons to map specific spatiotemporal input spike patterns to temporally precise output spikes is very useful, as the spiking outputs can then be decoded by higher order neurons sensitive to spike timing [147]. The temporally precise mapping can be achieved by adjusting weights of the input synapses. However, development of efficient algorithms for computing weights remains a challenge due to the complex nature of spike trains. For instance, elementary operations such as addition, subtraction and multiplication cannot be directly performed on spike trains. Existing learning rules typically employ heuristic approaches which may produce sub-optimal results. Hence, techniques to robustly optimize synaptic weights remains an important area of research for SNNs.

Several techniques to train spiking neurons have been reported in the literature. The Tempotron [145] method trains a neuron to fire or suppress when presented with an input pattern. However, the time of the output spike carries no significance. Algorithms such as Remote Supervised Method (ReSuMe) [148], Chronotron [149], Spike Pattern Association Neuron (SPAN) [150] and Precise-Spike-Driven (PSD) [151] model neurons to output temporally coded spikes. The Chronotron (E-learning) technique uses a gradient descent approach where weights are adjusted to minimize an error function defined by a modified Victor Purpura distance metric [152]. The ReSuMe, SPAN, PSD and I-learning version of Chronotron use heuristic approaches derived from the Widrow-Hoff rule [135]. These approaches do not guarantee a globally optimal solution because of the non-convex nature of the minimization criteria. As a result, these techniques are

susceptible to being trapped at saddle points and, therefore, a feasible (globally optimal) solution may not be found even if one exists.

The recently developed Synaptic Kernel Inverse Method (SKIM) proposes using the pseudo-inverse to solve for weights of a discrete time perceptron [209]. SKIM computes weights by minimizing the error between the perceptron membrane voltage and its intended binary output. Its response is decoded by applying a hard threshold, which contingent upon the kernel used, may span several milliseconds. This continuous output requires additional processing to convert it to spikes. Although SKIM obeys convex minimization, it is not designed to produce precise spike times.

In this chapter, we introduce a novel Convex-Optimized-syNaptic-Efficacies (CONE) approach to compute the weights of a spiking neuron. Our approach differs from SKIM since the output neuron is a spiking model that permits the computation of temporally precise spikes. We show two variants of this technique as follows. (1) CONE-X: An exact formulation that guarantees a solution, if the problem is feasible. This allows CONE-X to be used to determine the existence of a solution for a given dataset. (2) CONE-R: A robust formulation that permits the computation of the best estimate when an exact solution is not possible. In addition, we introduce mechanisms to alter the weight distribution of the solution and demonstrate that this redistribution has significant influence on the accuracy of outputs in the presence of noise.

The organization of this chapter is as follows. Section 6.3 presents the SNN problem and describes the CONE formulation. Section 6.4 demonstrates the characteristics of CONE through simulation examples that are designed to elucidate how the model parameters affect the solution. In Section 6.5, CONE is applied to a practical application where the goal is to detect gait-events using an insole pressure sensor. Section 6.6 provides a discussion of our findings and Section 6.7 provides concluding remarks about our work.

6.3 Methods

The objective of CONE is to compute an optimal set of weights that permit a desired mapping function for a spiking neuron. This allows a spiking neuron to respond to

specific spatiotemporal input spike patterns with temporally precise output spikes. In this section, we provide a brief introduction to neuron dynamics followed by a detailed description of the CONE formulation.

6.3.1 Neuron dynamics

A SNN's neuron communicates through trains of action potentials (spikes). A spike train with F spikes is represented as

$$s(t) = \sum_{f=1}^{F} \delta(t - t^f) \tag{6.1}$$

where $\delta(\cdot)$ is the Dirac delta function, $t \in \mathbb{Z}$ is time and t^f is the time of the fth spike in a spike train. The input, $s_n(t)$, and target, $s_d(t)$, spike trains are defined as

$$s_n(t) = \sum_{i=1}^{I} \delta(t - t_n^i) \quad n = 1, 2, \cdots N \tag{6.2}$$

$$s_d(t) = \sum_{k=1}^{K} \delta(t - t_d^k) \tag{6.3}$$

where t_n^i is the ith spike from the nth synapse, and I refers to the total number of spikes received from the nth synapse. With an optimal set of weights, the mapping neuron will produce the target spike train $s_d(t)$ containing K spikes when presented with N input spike trains (Fig. 6.1).

In this paper, the mapping neuron is modeled using the simplified Spike Response Model (SRM_0) due to it being biologically plausible and computationally tractable [205]. The state of a SRM_0 neuron is determined by its membrane voltage U. The sub-threshold dynamics of the neuron is described by

$$U(\mathbf{w}, t) = \sum_{n=1}^{N} w_n \cdot \sum_{i=1}^{I} \alpha_n(t - t_n^i) \tag{6.4}$$

where w_n is a synaptic weight and $\mathbf{w} \in \mathbb{R}^N$ is the vector of weights. If no input spikes are received, U is at its resting potential, 0 mV. A change in U is evoked when an input

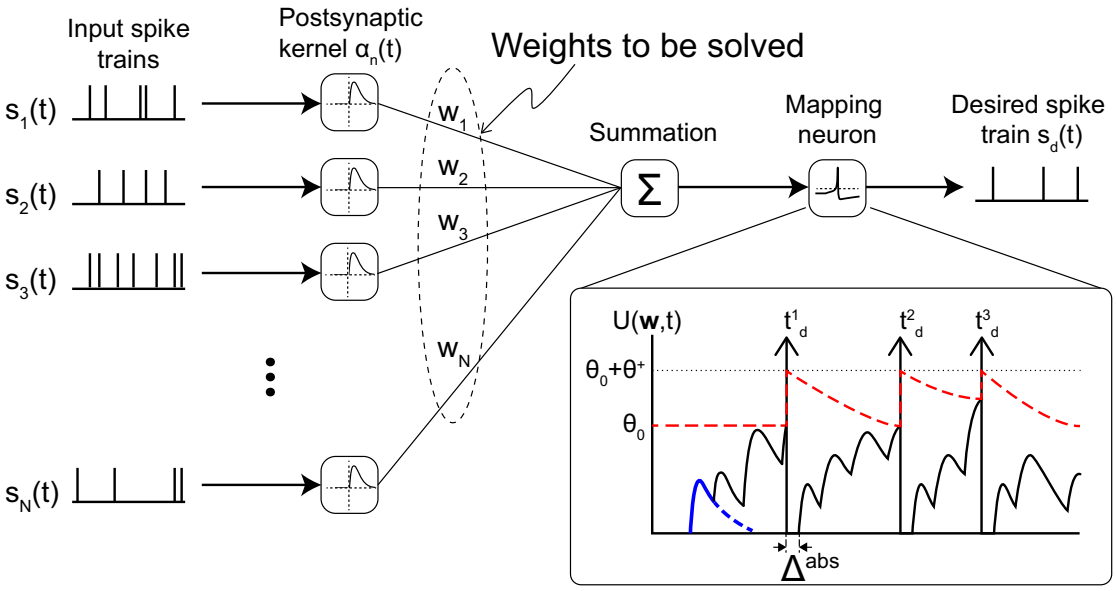


Figure 6.1: CONE is used to compute a set of weights by mapping N input spike trains to a predefined output spike train with temporal precision. Neuron dynamics are illustrated in the lower right panel. Each input spike evokes a post-synaptic potential (PSP) (thick curve) and the membrane potential is a summation of all PSPs. When the membrane potential exceeds a threshold θ_0 a spike is discharged and the neuron enters an absolute refractory period, Δ^{abs}. A relative refractory period is also implemented by a temporary increase of threshold from θ_0 to $\theta_0 + \theta^+$.

spike is received, then slowly returns to zero. For a given input synapse, s_n, a change in membrane voltage is represented by the function $\alpha_n(\cdot)$ whose magnitude is modulated by the weight, w_n.

Although alternative kernel functions are equally valid [209], we employ the α-Kernel due to its resemblance to experimentally observed postsynaptic potential (PSP) signals [205]. The α-Kernel is described as

$$\alpha_n(t - t_n^i) = \frac{t - t_n^i}{\tau} \exp\left(-\frac{t - t_n^i}{\tau}\right) \cdot H(t - t_n^i) \tag{6.5}$$

where τ is the neuron membrane time constant and $H(\cdot)$ is the Heaviside step function ($H(x) = 1$ for $x \geq 0$ and $H(x) = 0$ elsewhere).

The SRM_0 neuron has a membrane threshold voltage of θ. When $U \geq \theta$ the neuron discharges an output spike. After a spike is triggered, the neuron undergoes an absolute refractory period where U is held at its resting potential for a duration of Δ^{abs} sec. The refractory period is described as

$$\rho(t - \hat{t}_o) = \begin{cases} 0 & \text{if } 0 < t - \hat{t}_o \leq \Delta^{\text{abs}} \\ 1 & \text{otherwise} \end{cases} \tag{6.6}$$

where $\hat{t}_o$ is the time of the current output spike. Throughout the absolute refractory period all input spikes have no effect. Combining the post potential spike behavior (Eqs. 6.5 & 6.6) with the dynamics of U (Eq. 6.4) yields an expression for membrane voltage as

$$U(\mathbf{w}, t) = \rho(t - \hat{t}_o) \left(\sum_{n=1}^{N} w_n \cdot \sum_{i=1}^{I} \alpha(t - t_n^i) \right). \tag{6.7}$$

In addition, the SRM_0 neuron undergoes a relative refractory period that simulates neuronal adaptation. It is implemented using a dynamic threshold that is increased by θ^+ from its baseline value θ_0 after a spike as [205]

$$\theta(t - \hat{t}_o) = \theta_0 + \theta^+ \exp\left(-\frac{t - \hat{t}_o - \Delta^{\text{abs}}}{\tau_{\text{rec}}}\right) \cdot H(t - \hat{t}_o - \Delta^{\text{abs}}) \tag{6.8}$$

where τ_{rec} is the threshold recovery time constant. The lower right panel of Fig. 6.1 illustrates the dynamics described above.

6.3.2 CONE-X

The central goal of training a spiking neuron is that of emitting temporally precise action potentials. This is accomplished by computing an appropriate set of weights for a spiking neuron model. In this section, we provide details of how our CONE-X technique calculates optimal weights using a convex optimization approach.

For a set of weights to be feasible they have to satisfy the following constraints

$$U(\mathbf{w}, t_d^k) = \theta(t_d^k - t_d^{k-1}) \qquad k = 1 \cdots K \tag{6.9a}$$

$$U(\mathbf{w}, t') < \theta(t' - t_d^{k-1}) \qquad t' \notin \{t_d^k \leq t \leq t_d^k + \Delta^{\text{abs}}\} \tag{6.9b}$$

$$lb \preceq \mathbf{w} \preceq ub. \tag{6.9c}$$

where t' are time points outside the spiking and refractory periods and t_d^{k-1} is the time of the preceding target spike, with $t_d^0 = 0$. The constants lb and ub are the lower and upper weight bounds, respectively. The equality in Eq. 6.9a permits the membrane potential to reach a threshold at each target spike time, triggering an action potential. The inequality in Eq. 6.9b ensures the membrane potential is below a defined threshold for all time except target spike times. In addition, the boundary constraints in Eq. 6.9c are imposed to ensure that the solution space is bounded. The boundary conditions may also be used to specify the limits of the target system, such as spiking networks in hardware, or biological neurons. Weights that satisfy the conditions given in Eqs. 6.9a-6.9c enable a neuron to fire at only the target spike times. These constraints are optimized for the noise free case. Therefore, it is necessary to assess their utility and limitations to yield robust weight estimates in the presence of noise corrupted inputs.

Input spike patterns are considered to be noise corrupted if they differ from the training pattern. Deviations may be the result of (1) missing spikes; (2) additional spikes; or (3) spike time jitter. For a mapping neuron, the noise results in membrane voltage fluctuation and may lead to false positives or negatives. False positives refer to

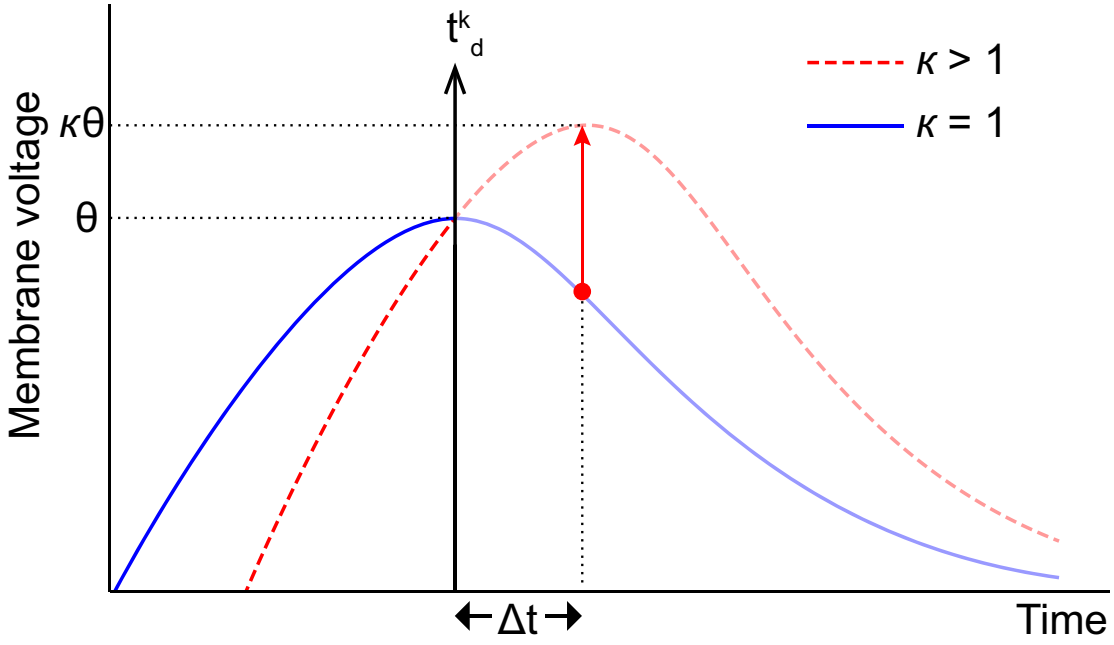


Figure 6.2: Increasing post-target membrane potential while maintaining spike time provides improved robustness to false negatives for $\kappa > 1$.

spikes discharged by a neuron at the wrong time. False negatives refer to a failure to emit spikes at the intended time. Therefore, robustness to noise is defined as the ability of our model to map temporally precise spikes in the presence of membrane voltage variation.

For perceptrons, noise robustness is quantified by a threshold stability parameter ξ [210]. For a binary classification task, perceptrons are trained to represent a positive stimuli with an output exceeding a threshold by $+\xi$. A negative stimulus is represented with an output below the threshold by $-\xi$. A large $|\xi|$ gives greater contrast between positive and negative responses, increasing robustness of the solution [210, 211].

For the CONE formulation, we reduce the likelihood of false positives by minimizing the magnitude of the neuron weights. This results in the membrane voltage being near its resting potential when it is not time to spike, ensuring small positive fluctuations due to noise do not evoke a false response. To compute optimal weights, we minimize the: (1) ℓ_2-norm, $\|\mathbf{w}\|_2$ or (2) ℓ_1-norm, $\|\mathbf{w}\|_1$. Regardless of the minimization criteria, the computed weights will evoke the neuron to discharge because of the spiking constraint (Eq. 6.9a). The effects of minimizing ℓ_1 or ℓ_2-norms are investigated in detail in Sections 6.4.2 and 6.4.7.

To reduce false negatives, perceptrons are trained with a higher membrane threshold. The higher limit causes the membrane potential to rise significantly above the threshold during training and reduces the chance of it falling below when a target stimulus is presented during testing. However, this strategy is not well-suited for CONE because the α-Kernel (Eq. 6.5) does not peak immediately when it is perturbed by an input spike. Reverting to the initial (lower) threshold after training, the membrane voltage produced by a similar input will exceed the threshold earlier than intended. This results in premature spike times. To resolve these issues, with our CONE technique, false negatives are reduced by increasing the post-target membrane voltage. This is achieved by minimizing the following objective function (see also Fig. 6.2)

$$B(\mathbf{w}) = \sum_{k=1}^{K} (U(\mathbf{w}, t_d^k + \Delta t) - \kappa\theta(t_d^k - t_d^{k-1}))^2 \qquad \kappa > 1 \tag{6.10}$$

where K is the number of target output spikes. The constant κ controls the magnitude of the post-target potential while the constant Δt defines the post-target potentiation time step. Notice that the membrane voltage does not increase beyond the threshold at $t = t_d^k + \Delta t$ because the neuron begins a refractory period. This results in increased robustness to false negatives because a larger negative change is required to prevent the neuron from spiking while maintaining the same output spike time should the same training stimulus be applied. To permit the estimation of neuron weights with these additional criterion, we formulate the problem in a convex optimization framework as

$$\text{Minimize:} \quad \hat{\lambda}_1 \frac{\|\mathbf{w}\|_1}{\|\mathbf{w_{max}}\|_1} + \hat{\lambda}_2 \frac{\|\mathbf{w}\|_2}{\|\mathbf{w_{max}}\|_2} + \hat{\lambda}_3 \frac{B(\mathbf{w})}{B_{\max}}$$

Subject to:

$$\begin{aligned} & U(\mathbf{w}, t_d^k) = \theta(t_d^k - t_d^{k-1}) \quad k = 1 \cdots K \\ & U(\mathbf{w}, t') < \theta(t' - t_d^{k-1}) \quad t' \notin \{t_d^k \le t \le t_d^k + \Delta^{\text{abs}}\} \\ & lb \preceq \mathbf{w} \preceq ub. \end{aligned} \tag{6.11}$$

where $\hat{\lambda}_1$, $\hat{\lambda}_2$ and $\hat{\lambda}_3$ are regularization parameters that "tune" the solution. A large

$\hat{\lambda}_i$ value increases the contribution of its objective function. The vectors $\mathbf{w_{min}}$ and $\mathbf{w_{max}}$ are the minimum and maximum values of the weight vectors, respectively. $B_{\max}$ indicates the largest value of the objective, depending on the limits of $\mathbf{w}$ specified. We used $B_{\max} = B(\mathbf{w_{min}})$ when $lb = -ub$. They serve to normalize the objectives and permit a comparison across different scales [212]. We refer to the formulation given in Eq. 6.11 as CONE-X.

6.3.3 CONE-R

The optimization problem in Eq.6.11 assumes the training dataset is noise free. However, when training neurons with experimental data, noise in the training dataset must be considered. For example, there may be differences between input spike patterns across multiple trials or target output spike times may not be accurately labeled. CONE-X will yield suboptimal results, or declare the problem infeasible (no solution possible) for these cases because the constraints will be inconsistent. To address this issue, we modify the equality and inequality constraints in Eq. 6.11 by replacing them with penalty terms

$$C(\mathbf{w}) = \sum_{k=1}^{K} \left\| U(\mathbf{w}, t_d^k) - \theta(t_d^k - t_d^{k-1}) \right\|_1 \tag{6.12}$$

$$D(\mathbf{w}) = \sum_{t'} max\{U(\mathbf{w}, t') - \theta(t' - t_d^{k-1}),\ 0\} \tag{6.13}$$

$$t' \notin \{t_d^k \leq t \leq t_d^k + \Delta^{\text{abs}}\}$$

where $C(\mathbf{w})$ represents the magnitude the equality constraint (Eq. 6.9a) is violated and $D(\mathbf{w})$ denotes the amount by which the inequality constraint (Eq. 6.9b) is exceeded.

This allows us to redefine the problem formulation to permit the analysis of noise corrupted data as

$$\begin{aligned}\text{Minimize:}\quad & \hat{\lambda}_1 \frac{\|\mathbf{w}\|_1}{\|\mathbf{w_{max}}\|_1} + \hat{\lambda}_2 \frac{\|\mathbf{w}\|_2}{\|\mathbf{w_{max}}\|_2} + \hat{\lambda}_3 \frac{B(\mathbf{w})}{B_{\max}} + \\ & \hat{\lambda}_4 \frac{C(\mathbf{w})}{C_{\max}} + \hat{\lambda}_5 \frac{D(\mathbf{w})}{D_{\max}} \\ \text{Subject to:}\quad & lb \preceq \mathbf{w} \preceq ub. \end{aligned} \tag{6.14}$$

where the additional regularization parameters $\hat{\lambda}_4$ and $\hat{\lambda}_5$ control the objective terms that describe constraint violations. Large values of the additional regularizers guarantee minimal violations of the constraints given in Eq. 6.9a-6.9b. $C_{\max} = C(\mathbf{w_{min}})$ and $D_{\max} = D(\mathbf{w_{max}})$ are the largest possible values of the respective objective functions and, therefore, we used these values when $lb = -ub$. We refer to the formulation in Eq. 6.14 as CONE-R.

In this paper, all the regularizations are normalized to allow for ease of comparison since they are relative. We define the regularization vectors $\boldsymbol{\lambda}_X$ and $\boldsymbol{\lambda}_R$ as

$$\boldsymbol{\lambda}_X = \begin{pmatrix} \lambda_1 & \lambda_2 & \lambda_3 \end{pmatrix} \qquad \boldsymbol{\lambda}_R = \begin{pmatrix} \lambda_1 & \lambda_2 & \lambda_3 & \lambda_4 & \lambda_5 \end{pmatrix}$$

$$\text{where} \quad \hat{\lambda}_i = \frac{\lambda_i}{\Sigma(\boldsymbol{\lambda}_{X,R})} \begin{cases} i = 1, 2, 3 & \text{for } \boldsymbol{\lambda}_X \\ i = 1, 2, \ldots, 5 & \text{for } \boldsymbol{\lambda}_R. \end{cases}$$

The vectors $\boldsymbol{\lambda}_X$ and $\boldsymbol{\lambda}_R$ represent the regularization terms used for CONE-X and CONE-R, respectively.

6.3.4 CONE-X and CONE-R Solution Surface

The CONE-X and CONE-R problem formulations are illustrated in Fig. 6.3. The panels show the solution surface of a neuron with two synapses that discharge a single spike when perturbed with one input spike per synapse. The weights form a 2D solution space with the objective function defining a convex surface. For the CONE-X formulation, the solution space is bound by hard limits defined by the inequality constraint in Eq. 6.9b and weight boundaries in Eq. 6.9c. The equality constraint Eq. 6.9a is illustrated as

a line in the solution space (Fig. 6.3a). The projection of the line onto the convex surface indicates all the feasible points while the optimal point is the one with the lowest objective along the projected line.

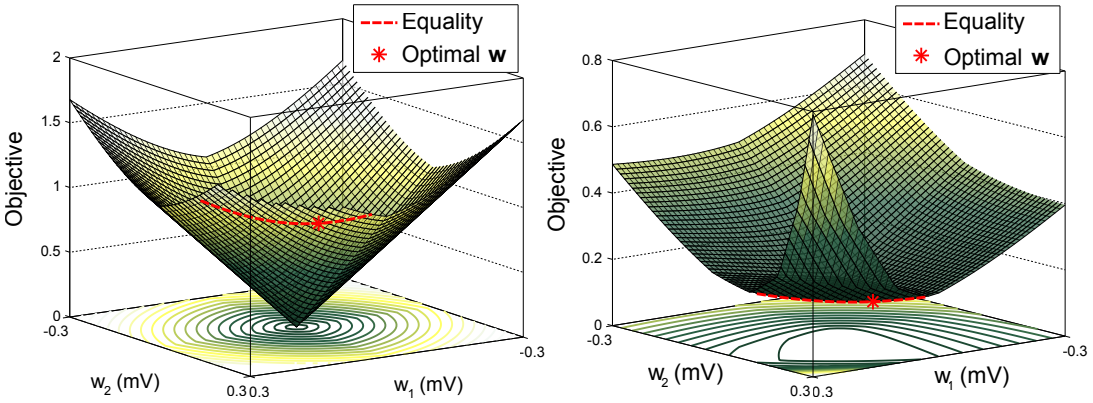
Figure 6.3b illustrates the solution surface for the same simulation but using the CONE-R formulation. Using large values of $\hat{\lambda}_4$ and $\hat{\lambda}_5$ the convex surface was reshaped such that the objective function was smallest along points of the equality constraint while points violating the inequality constraints incurred large penalties (Fig. 6.3b). Note that in a non-ideal situation noise would result in a "bumpy" solution surface. A consequence of the bumpy surface is that it only allows an (initial) solution to be in the neighborhood of the global minimum. The global minimum may be achieved using iterative procedures [212].

An important advantage of the convex formulation of CONE-X, compared to heuristic approaches, is its ability to show that no solutions exist. This is possible because the problem of determining a realizable solution is defined as convex and strictly feasible [212]. Demonstrating a solution does not exist is typically accomplished by solvers based on interior-point methods, such as Sedumi [213] and MOSEK [214], which produce a computational indicator, known as *certificate of infeasibility*, in the event of failure. Hence, solving the optimization problem using CONE-X will always produce a solution that fits the training samples exactly, if the problem is feasible. Alternatively, the CONE-R formulation is well suited for practical applications such as pattern recognition because it produces an "approximate optimal" or "best possible" solution even if the problem is infeasible (no exact solution).

6.3.5 Error metric

Accuracy of a mapping operation is quantified using the Victor Purpura (VP) spike train distance [201]. In this paper, the VP metric represents the difference between the predicted or estimated neuron output and the intended target output. The VP computes the minimum cost required to transform one train to another through a set of elementary operations, namely:

1. Spike insertion: Cost of 1 per spike inserted.



(a) CONE-X. $\boldsymbol{\lambda}_X = (2, 2, 0)$ **(b)** CONE-R. $\boldsymbol{\lambda}_R = (2, 2, 0, 10, 10)$

Figure 6.3: Solution surface for a neuron with 2 synapses. Left: CONE-X. Right: CONE-R.

2. Spike deletion: Cost of 1 per spike deleted.

3. Spike shifting: Cost of $q|\Delta t|$ is associated with moving a spike by a time interval Δt, where q is a free parameter.

For large q, the VP metric reflects the number of non-coincident spikes because it is computationally cheaper to delete and re-insert these spikes rather than shift them. For small q, the metric reflects the difference in the number of spikes between two spike trains. This allows errors due to temporal inaccuracy to be distinguished from false positives or negatives. For a detailed discussion on the VP metric the reader is referred to [201].

6.4 Simulations

In this section, we evaluate the efficacy of the CONE approaches and assess how various model parameters influence their solution. The principal concepts are illustrated using an example of mapping a spatiotemporal spike pattern to a target spike train. This is followed by an investigation of memory capacity for the neuron model using our approach. Lastly, we investigate the effects of noise corrupted inputs on our technique to show how various criterion affect the stability of its solution.

Table 6.1: Neuron parameters.

Parameter	Value
Membrane time constant τ	15 ms
Baseline threshold θ_0	0.3 mV
Maximum threshold increase θ^+	0.2 mV
Threshold recovery time constant τ_{rec}	30 ms
Absolute refractory period Δ^{abs}	3 ms
Post-target potentiation time Δt	1.5 ms
Weight bounds	$[-0.3,\ 0.3]$ mV

6.4.1 CONE-X: Mapping spike trains to a single output spike

To demonstrate the utility of CONE-X for spike pattern recognition, neuron weights were estimated to enable a neuron to fire a single spike given a spatiotemporal input pattern. In this work, we required the output spike times be precise and relative to the start time of the pattern, which was not a constraint in previous work [145]. In addition, effects of the learning parameters were investigated.

Simulation Protocol

One hundred spike trains ($N = 100$) with a single spike uniformly distributed between 0 to 100 ms were used as inputs to the neuron. The target spike was defined at $t_d^1 = 100$ ms. The neuron was modeled with parameters shown in Table 6.1. The neural parameters are typical of those used in similar studies [145, 148–151] and are of a similar order of magnitude to biological observations [215]. In this paper, all case studies were modeled using CVX [216] and solved using SDPT3 [217] in the MATLAB platform, unless stated otherwise. The SDPT3 solver was selected for its widespread availability.

Results

The membrane potential crossed the threshold at 100 ms for all values of κ (Fig. 6.4). Therefore, all sets of solutions were feasible, evoking spikes at the intended time. This example illustrates that increasing κ results in higher membrane voltage trajectories (cf. Fig. 6.2) and provides a solution that is robust against false negatives because small downward perturbations from noise is unlikely to prevent the neuron from spiking. This was achieved by potentiating the weights of synapses that deliver spikes closer to the

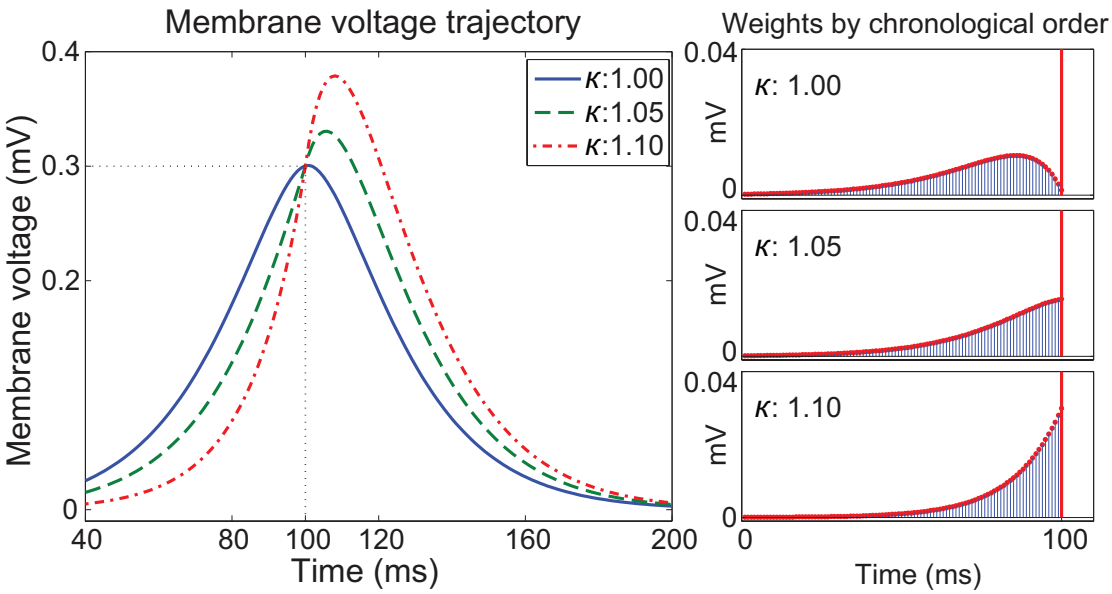


Figure 6.4: Left: Membrane potential trajectory excluding spiking behaviour. A large κ increases the post-target membrane potential. Right: Weights of relevant synapses are increased for $\boldsymbol{\lambda}_X = (0, 2, 100)$.

target spike time (Fig. 6.4). However, large weights also increase the chance of false positives since it is easier for unintended spikes to occur.

6.4.2 CONE-X: Effects of ℓ_1 and ℓ_2-norm estimation

To investigate the differences between an ℓ_1 and ℓ_2-norm solution, the experiment was repeated with different ratios of $\lambda_1 : \lambda_2$.

Results show the output spike time of the neuron remains unchanged (Fig. 6.5). However, the weight distribution is affected. The ℓ_2-norm provides a solution where all synapses contribute to the output spike. The ℓ_1-norm yields a solution where only the most significant synapses contribute. Adjusting the ratio between regularization parameters λ_1 and λ_2, the sparsity of the solution can be tuned. The sparsity of a solution has important implications for noise robustness (see §6.4.5).

In related work, synapses with the closest spike times to the target were potentiated the most [148]. However, with our CONE-X approach only the most effective synapses are potentiated. This yields a more efficient solution because the weights are of smaller magnitude for the same effect.

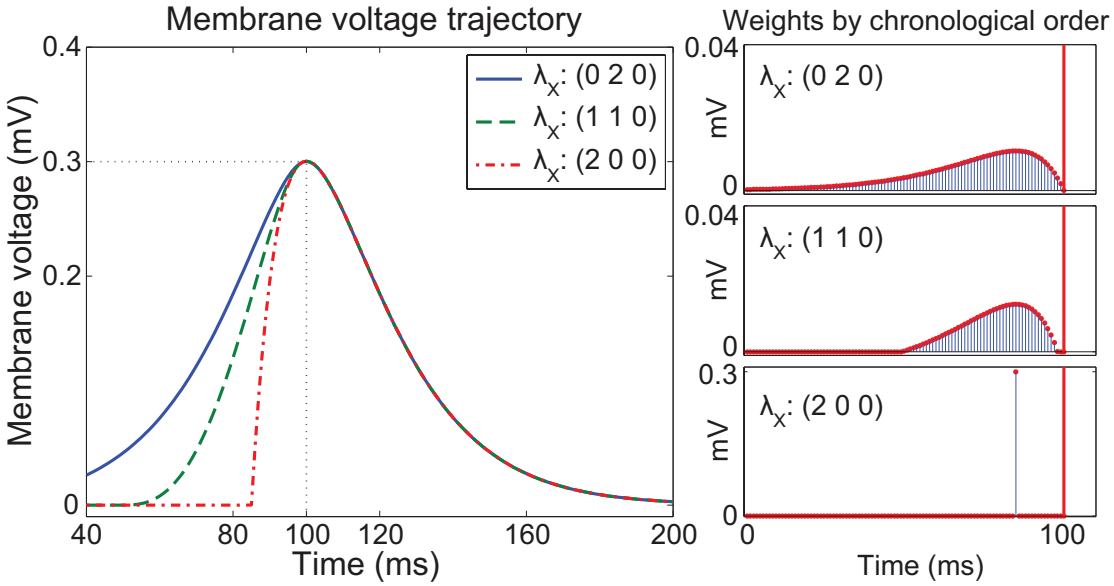


Figure 6.5: Left: Membrane potential trajectory excluding spiking behaviour. Right: Estimated weights of neurons optimized by varying $\boldsymbol{\lambda}_X$. Sparse solutions obtained for $\lambda_2 > \lambda_1$, $\kappa = 1$.

6.4.3 CONE-X: Mapping spike trains to multiple output spikes

Next, we investigate the behavior of our technique to map random Poisson spike trains to a target spike train with multiple spikes.

Simulation Protocol

Five hundred random spike trains ($N = 500$) were generated as inputs to the neuron with a Poisson distribution, 200 ms duration and a firing rate between 5 and 25 Hz. The target spike train was chosen arbitrarily to consist of 5 spikes located at $t_d^1 = 30$ ms, $t_d^2 = 40$ ms, $t_d^3 = 50$ ms, $t_d^4 = 120$ ms and $t_d^5 = 150$ ms. The target spikes were fixed across trials to allow easy comparison.

To obtain an exact fit the CONE-X formulation was used. Neuron parameters were the same as described in Table 6.1. To assess the effects of solution sparsity on the neuron membrane voltage, we used two regularization parameters sets; $\boldsymbol{\lambda}_X^1 = (0,\ 1,\ 0)$ and $\boldsymbol{\lambda}_X^2 = (1,\ 0,\ 0)$. One hundred realizations each with a randomly generated input were used to determine the effects of the estimation norm.

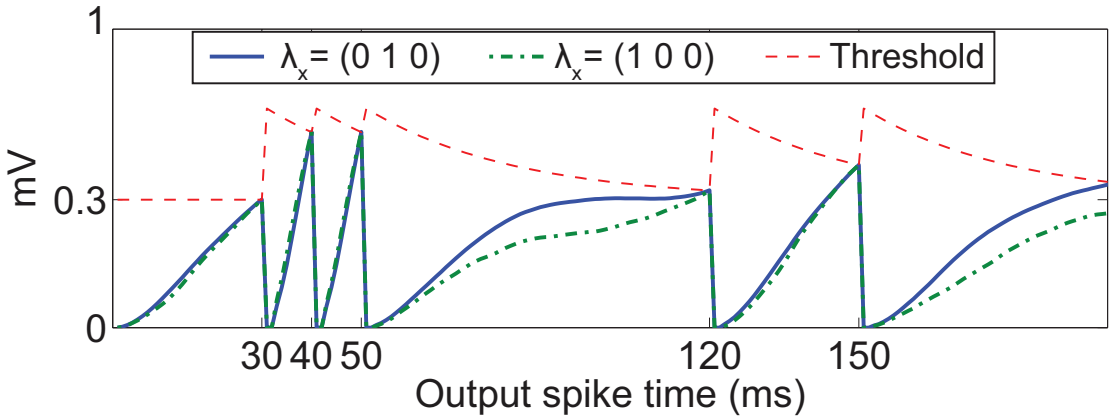


Figure 6.6: Average of 100 membrane voltage trajectories for a multiple spikes output neuron. Solid line: ℓ_2-norm estimate. Dot-dash line: ℓ_1-norm estimate. Dashed line: neuron threshold.

Results

For all realizations, the estimated weights accurately mapped the Poisson spike trains to the output target. Fig. 6.6 illustrates, during inter-output intervals, the ℓ_1 estimator offered a solution with a lower membrane voltage than the ℓ_2 estimate. A solution that offers a lower inter-output voltage trajectory may be useful for cases where there is a high level of background noise since an ℓ_1-norm is ideally suited for data with outliers. However, a sparse solution is not always the best choice (see § 6.4.5).

6.4.4 CONE-X: Memory capacity

A useful performance measure for a SNN algorithm is the amount of information a neuron can be trained to classify. To quantify performance we utilized the load factor metric, $L = p/N$. It is defined as the number of input patterns, p, a solution can successfully classify given a fixed number of input synapses, N. The goal is to determine the maximum load factor, $L_{\max} = p_{\max}/N$, which is the maximum number of patterns, $p_{\max}$, that can be classified per synapse. This procedure presents a worst case scenario where uncorrelated inputs have to be mapped onto identical outputs, setting an upper bound on CONE-X's effectiveness to precisely map spikes temporally. A memory capacity measure is commonly used to assess the effectiveness of SNN algorithms [145, 149–151].

Simulation Protocol

The spatiotemporal patterns were classified using a latency code. When presented with a pattern from class c, the trained neuron should emit a single spike at time $t_d^1 = 200c/(C+1)$, where $c = 1, 2, \ldots, C$. Each pattern consisted of N spike trains that had a single spike randomly located in the interval $[1, 2, 3, \ldots, 200]$ ms. Hence, each pattern is drawn from a set of 200^N possible patterns. The patterns were randomly and evenly assigned to the classes (i.e. p is a multiple of C). The algorithm was considered capable of managing a load, L, if it found a solution that correctly classified all p patterns consistently for 100 realizations. For example, an algorithm that can successfully process a load of 0.3 with 400 input spike trains is able to compute, for 100 consecutive random realizations, a single solution that maps 120 patterns to the same output spike train. The highest L obtained for each combination of N and C was considered the maximum load factor, $L_{\max}$.

The neuron parameters used here were the same as given in Table 6.1. The value of N was varied in steps of 10 from 50−200, steps of 50 from 200−500 and steps of 100 from 500−1000. In addition to varying N, the experiments were performed for differing number of classes, $C = 3, 5$ and 7. Due to the large computational requirements, it was not possible to perform simulations for finer increments. For this study, the problem was formulated using YALMIP [218] and solved using the Gurobi optimizer [219] because this combination was the fastest at determining the feasibility of a problem.

Results

The load factor increased rapidly until approximately $N = 150$ synapses and thereafter only incremental improvement was observed (Fig. 6.7). The maximum load factor saturated at about 0.3, independent of the number of classes. For $N < 200$, some variation of $L_{\max}$ was observed for different class sizes. This is because L can only vary discretely since p increments in multiples of C. The maximum load factor achieved by our CONE-X technique is significantly higher than those reported previously under similar conditions (0.22 and 0.02 for E-learning and I-learning Chronotrons, 0.04 for ReSuMe [149], 0.075 for SPAN [150] and 0.144 for PSD [151]). This suggests that our

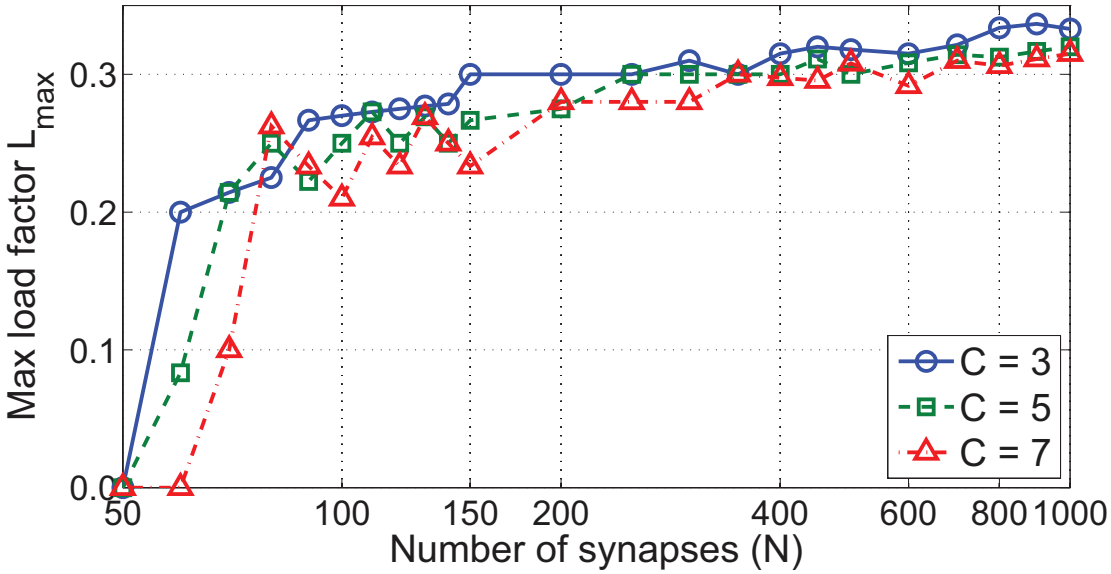


Figure 6.7: Maximum load factor for 100 realizations.

approach can classify more patterns with the same number of neurons, demonstrating the advantage of this convex optimization approach.

6.4.5 CONE-X: Noise free training and noise corrupted test data

In this section, we investigate the efficacy of CONE-X to transform waves of spatiotemporal spikes to a temporally precise pattern. In addition, we assess the accuracy of output spikes when they are corrupted by noise.

Simulation Protocol

The training set was a spatiotemporal pattern (φ) that consisted of two hundred spike trains ($N = 200$), each with a random single spike between 0-40 ms. A solution was computed with training data to enable the neuron to fire a single spike 40 ms after the start of the spatiotemporal pattern φ.

The test dataset was created by 30 repetitions of φ, with a random spacing of 50-100 ms inserted between each adjacent spatiotemporal pattern, forming a "wave-train" $\hat{\varphi}$ (See Fig. 6.8). The wave-train was corrupted by various types of noise as follows:

1. Jitter noise $X(\hat{\varphi}, x)$: Firing time of spikes in $\hat{\varphi}$ had a Gaussian zero mean process added to it. The standard deviation x ms defines the noise intensity.

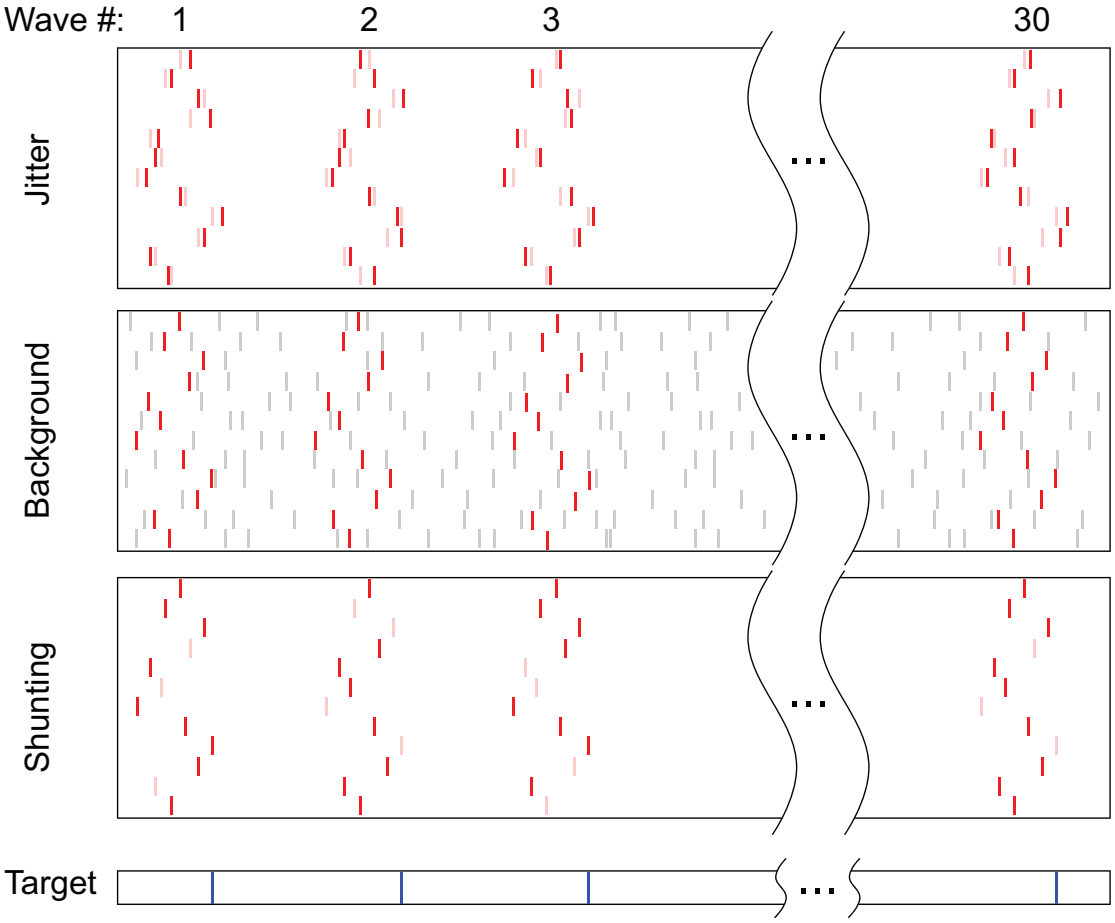


Figure 6.8: Types of noise introduced to φ. The top panel illustrates jitter noise, where resultant spike wave (solid red) is obtained by jittering spikes in φ (faint red). The second panel illustrates the addition of background noise (grey spikes) on top of φ (solid red). The third panel describes shunting noise, where spikes from φ are removed at random (faint red). A robust solution should produce the target spike train (bottom panel) despite the presence of noise.

2. Background noise $Y(\hat{\boldsymbol{\varphi}}, y)$: Spike trains with a Poisson distribution were added to $\hat{\boldsymbol{\varphi}}$. The firing rate y Hz determines the noise strength.

3. Shunting noise $Z(\hat{\boldsymbol{\varphi}}, z)$: Spikes were deleted from $\hat{\boldsymbol{\varphi}}$ in a uniform random manner. The percentage of spikes removed z % determines the noise magnitude.

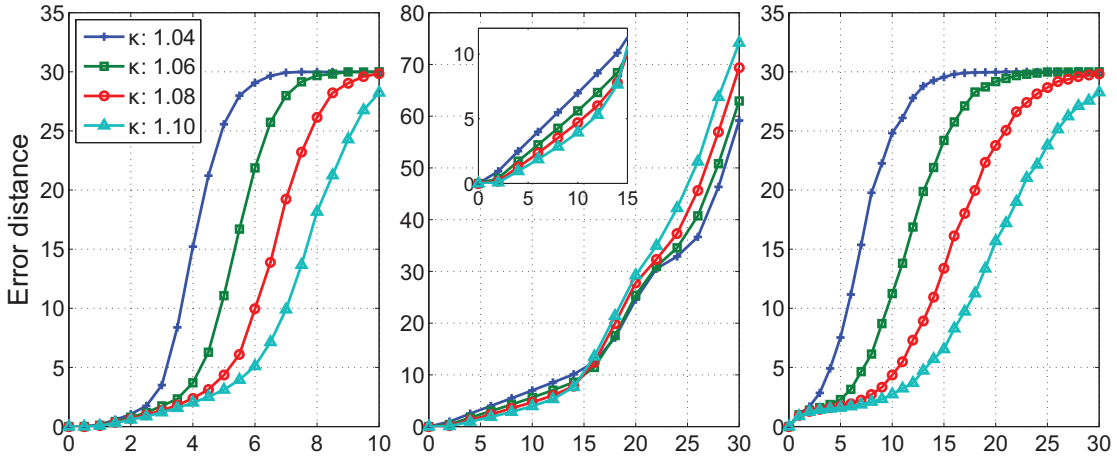
The trained neuron was presented with the noise corrupted datasets $X(\hat{\boldsymbol{\varphi}}, x)$, $Y(\hat{\boldsymbol{\varphi}}, y)$ and $Z(\hat{\boldsymbol{\varphi}}, z)$ for a range of values, $x \in [0, 10]$, $y \in [0, 30]$ and $z \in [0, 30]$. Results from 100 realizations were quantified using the VP error metric (see §6.3.5). A small $q = 0.05$ was used to create a clear distinction between errors from temporal deviations and additional/missing output spikes. The neuron parameters were the same as in Table 6.1 except $\tau = 10$ ms and $\theta^+ = 0$ mV. These parameter values were reduced to compensate for the shorter pattern durations.

Results

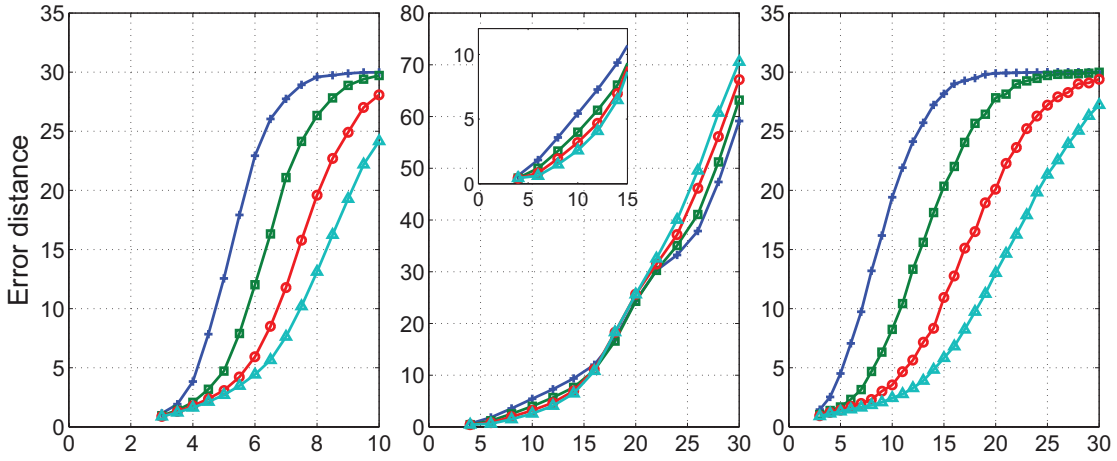
A downward shift in membrane voltage (U) was observed for data corrupted with jitter and shunting noise. This is a consequence of jitter noise disrupting the temporal structure of a pattern while shunting noise reduced the number of spikes that potentiate the membrane voltage. Increasing κ compensated for the downward shift and provided a lower error range (columns 1 & 3 of Fig. 6.9a).

When data was corrupted with background noise the optimization problem was more challenging (column 2 of Fig. 6.9a). For low frequency noise (< 15 Hz), it was possible to obtain more accurate solutions by increasing κ. This is because low frequency background noise primarily affects output time accuracy. The advantage of using a large κ is that fluctuations in membrane voltage due to noise translates to less variability in output spike time. This is a consequence of having a steeper gradient intersection between the membrane voltage and threshold (Fig. 6.2).

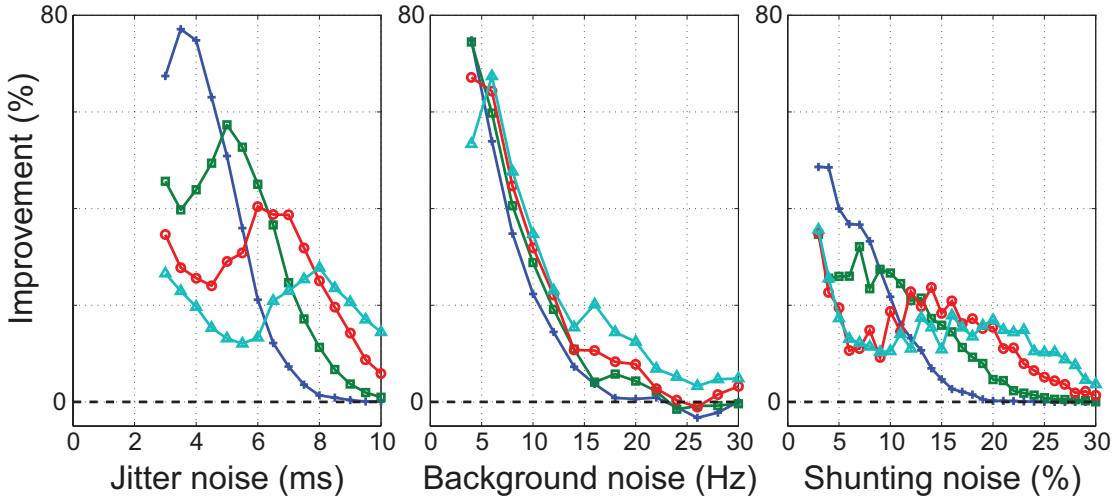
For high frequency noise (>15 Hz), small values of κ gave better solutions. A solution obtained with large κ will have larger weights (see Fig. 6.4). When there are a large number of input spikes the membrane voltage experiences a high level of potentiation. This increased potentiation leads to more false positives, illustrated by the rapid increase



(a) Model performance when trained with noise free data. Inset: magnified view of performance at lower background noise values. $\boldsymbol{\lambda}_X = (2, 2, 200)$



(b) Model performance when trained with noise corrupted data. Inset: magnified view of performance at lower background noise values. $\boldsymbol{\lambda}_R = (2, 2, 100, 100, 100)$



(c) Percentage improvement of (b) over (a).

Figure 6.9: Model performance with three types of noise corrupted data. Column 1: Jitter noise; Column 2: Background noise; Column 3: Shunting noise. Results represent mean of 100 realizations. Error distance quantified with VP error metric (see §6.3.5).

in error distance after 15 Hz. A small κ has the reverse effect and serves to reduce potentiation and decrease false positives.

6.4.6 CONE-R: Noise corrupted training and test data

In experimental applications it is not possible to have access to noise free training data. Therefore, to show applicability under laboratory conditions we repeated the previous experiments with noise added to the training data as well. For this study, we used the CONE-R technique (Eq. 6.14).

Simulation Protocol

The simulation protocol was the same as in § 6.4.5 except the training data were noise corrupted wave trains with the following characteristics, $X(\hat{\boldsymbol{\varphi}}, 3)$, $Y(\hat{\boldsymbol{\varphi}}, 3)$ and $Z(\hat{\boldsymbol{\varphi}}, 3)$. The corresponding test dataset was also corrupted using the same noise processes, varying only in noise intensity ($x \in [3, 10]$, $y \in [3, 30]$, $z \in [3, 30]$). Both the training and test datasets for each realization were taken from the same spatiotemporal pattern, φ. One hundred realizations were analyzed for each noise process.

Results

The results illustrate that using a large κ we estimated solutions that are robust to jitter and shunting noise (Fig. 6.9b). For background noise, a large κ provided good performance below 15 Hz while a smaller κ produced a better solution above 15 Hz. These results are similar to the noise free training case.

For all values of κ there was significant improvement for all three noise cases compared to noise free training data (Fig. 6.9c). In addition, these results are in good agreement with the findings of other studies, which also showed training with noise produced more robust solutions [145, 148–150].

6.4.7 CONE-R: Effects of ℓ_1 and ℓ_2-norm estimation

To assess the effects of sparsity on noise and solution robustness, the experiments in § 6.4.5 & 6.4.6 were repeated. Here we varied $\lambda_1 : \lambda_2$ ratios but fixed $\lambda_3 = \lambda_4 = \lambda_5 = 100$ and $\kappa = 1.08$. The CONE-R technique was used for weight estimation.

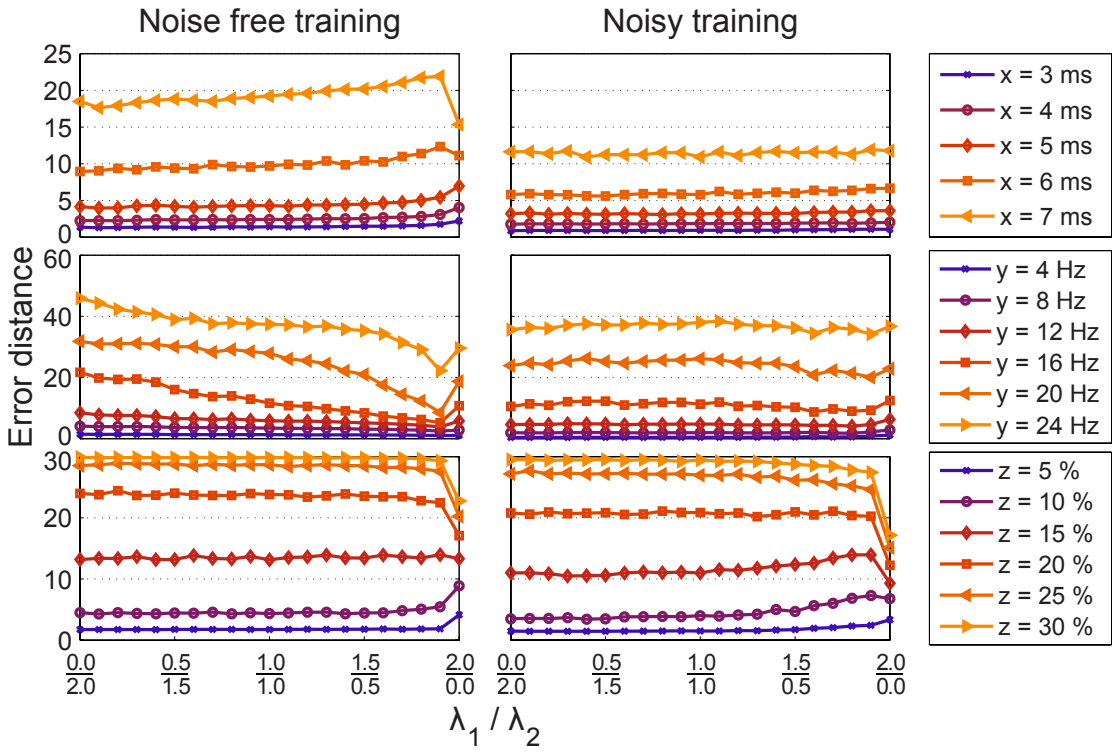


Figure 6.10: Error distance measure with varied $\lambda_1 : \lambda_2$ ratios and $\lambda_3 = \lambda_4 = \lambda_5 = 100$, $\kappa = 1.08$. Left: Noise free training data. Right: Noise corrupted training data. Top row: Jitter noise. Middle row: Background noise. Bottom row: Shunting noise. Results represent mean of 100 realizations.

When trained with noise free samples, dense solutions (pure ℓ_2 estimation) produced better performance (Fig. 6.10 top row) when tested on data with jitter noise. This is because a dense solution has more synapses contributing to produce an output spike, resulting in an averaging effect to reduce output time jitter. For additive background noise, sparse solutions (pure ℓ_1 estimation) were more robust because there were more synapses with zero weight (Fig. 6.10 middle row). For shunting noise, the results show more complex characteristics. For a high noise intensity, a sparse solution delivered better performance. For a low shunting noise strength, the reverse was true where a sparse solution yielded inferior performance (Fig. 6.10 bottom row). A sparse solution involves a small number of input spikes that contribute to an action potential because most weights are zero. Therefore, removing a small number of contributing spikes has a large effect on sparse solutions while it does not have significant consequences for a dense solution.

When trained with noise corrupted samples, the results of sparsity were less evident. The effects can be observed by comparing the weight distributions for the noise free and noise corrupted cases. When trained with noise free samples the weight distribution was multi-modal (Fig. 6.11 top row). This leads to a less selective solution because more synapses with larger weights were used to compute a final solution. However, when trained with noisy data the weight distribution was unimodal with a maximum centered around zero (Fig. 6.11 rows 2 to 4). This resulted in weights being tuned to maximize synapses that contributed most to the output and, hence, improving robustness of the solution.

We observed that the weight distributions obtained from noise corrupted training data have a close resemblance to biologically observed neurons, where the weight distribution decays monotonically from a maximum at zero weight [210]. Such a distribution has been hypothesized to be a consequence of high load factor due to noise [211]. This suggests that the CONE objective function, although not strictly biologically plausible, is a realistic formulation capable of producing biologically convincing results under similar (noisy) conditions.

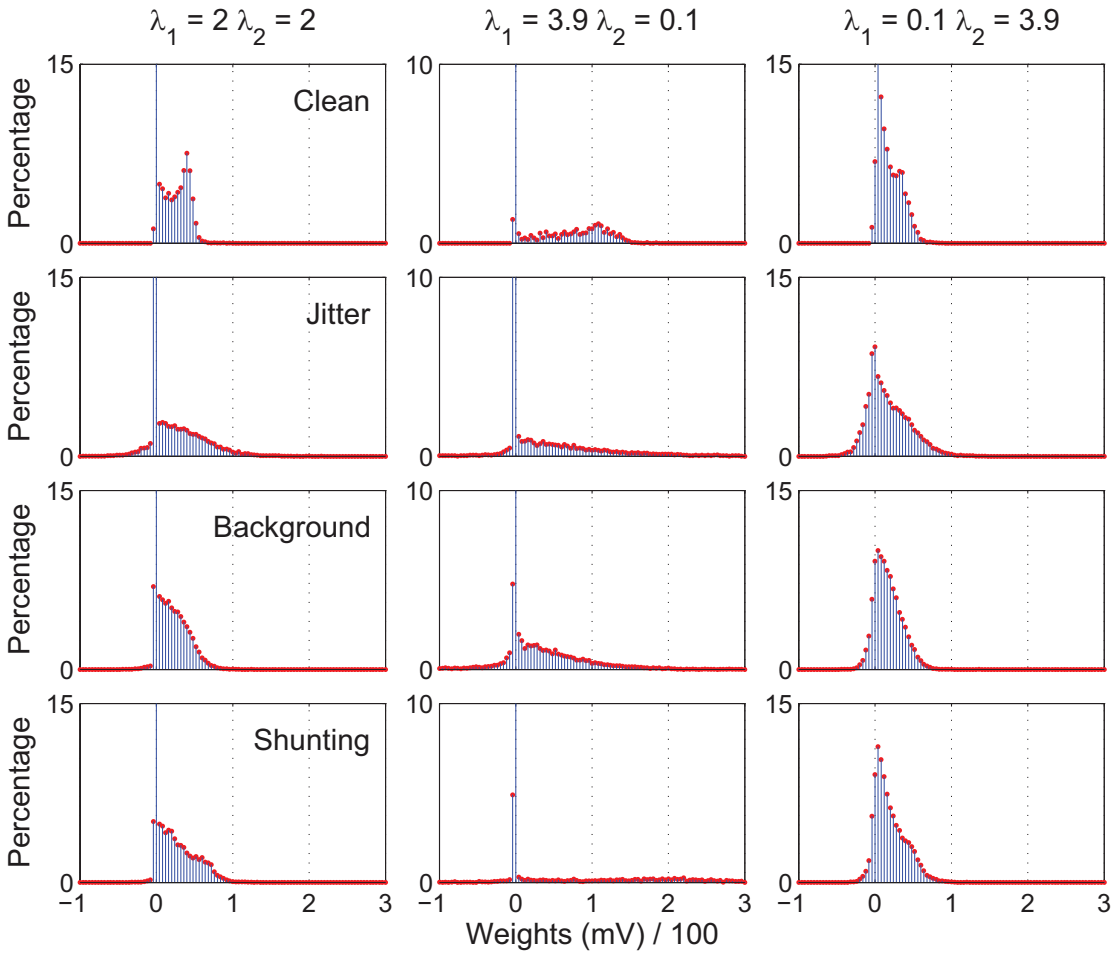


Figure 6.11: Mean weight distribution from 100 trials with $\kappa = 1.04$. Top row: Weights computed using noise free training samples. Other rows: Weights computed using noise corrupted training data.

6.5 Experimental Analysis - Gait Event Detection

Lastly, we demonstrate the practical applicability of our CONE-R technique under experimental conditions by applying it to gait-event detection. The data used for this analysis was collected from an insole pressure sensor. The objective was to train neurons to fire when a gait event occurs to illustrate application to temporally precise spike outputs.

The study of gait events is important in the fields of clinical gait analysis, biomechanics, prosthetic/exoskeleton design and robotics. Gait events are specific phenomena in time such as initial-contact and foot-off that are used as markers for gait segmentation [220], feedback to functional electrical stimulation (FES) and robotic prosthesis [221]. Such applications require near real-time robust feedback with low computational overhead and energy consumption. Event based computational platforms such as spiking neural networks have the potential to meet these criteria.

6.5.1 Experimental setup

Sensors

We developed a low cost foot pressure sensor using conductive fabric [189]. It was constructed by sandwiching a piezo-resistive fabric (NW-SLPA from Eeonyx) between two layers of conductive fabric strips (Silver coated nylon, LessEMF). The conductive fabric strips were arranged to form a grid where each intersection established a sensing element. The circuit was held together using non-conductive fabric fusible interfaces (Fig. 6.12). The sensor had 92 sensing points that was trimmed into the shape of a shoe insole and sampled at 1 kHz (USB-6356, National Instruments). The sensor did not need to be calibrated for accurate force measurements because our approach relies on temporal changes, not pressure intensity.

Spike conversion

Analog data from the sensor was converted to spikes compatible for use with CONE. We implemented the relative-spike-latency code for spike conversion because it is a rapid information propagation mechanism capable of meeting low latency requirements

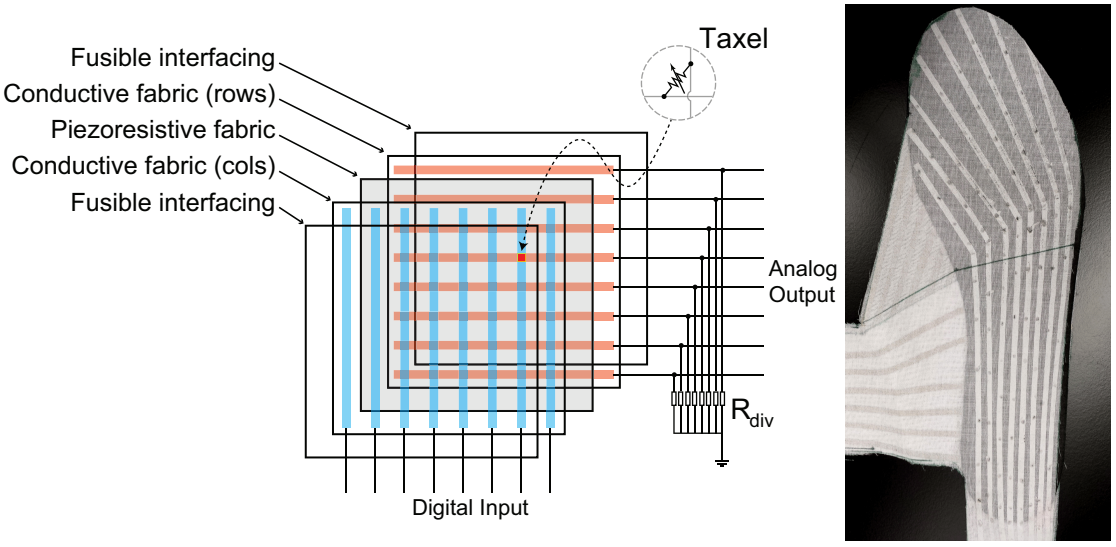


Figure 6.12: Left: Sensor architecture. Right: Fabricated shoe insole sensor.

[11]. This technique is simple yet has the potential to significantly reduced hardware requirements for sampling.

A hard threshold of 0.4 V was applied to each channel of the analog signal. The resulting signal and its inverse were used as inputs to an Izhikevich neuron model configured for phasic spiking using parameters a = 0.005, b = 0.25, c = -65, d = 6, I = 0.6 and simulated with 0.1 ms precision [222]. Outputs from the 184 neurons were used as inputs to the CONE algorithm (Fig. 6.13). Twice the number of neurons were required because the input signals and their inverse were used.

Our spike conversion procedure was motivated by recent discoveries of temporal coding in mechanoreceptors [24, 40]. Instead of computing the pressure distribution across a sensor surface, spatiotemporal patterns of pressure changes allow for sparse representations that can be transmitted and processed quickly [38]. In our model, we utilize a relative latency code [11], where spike times were represented as the time required for a single threshold to be crossed from above or below and smoothed by the Izhikevich neuron. Due to the uneven pressure distribution on the sensor during a gait cycle, spikes from each channel are transmitted with some latency relative to its neighbors, creating a spatiotemporal pattern. The use of a hard threshold permits the sensor to be sampled using a 1-bit analog to digital convertor (or a comparator). Such a circuit is inexpensive to manufacture, fast and requires a low readout bandwidth.

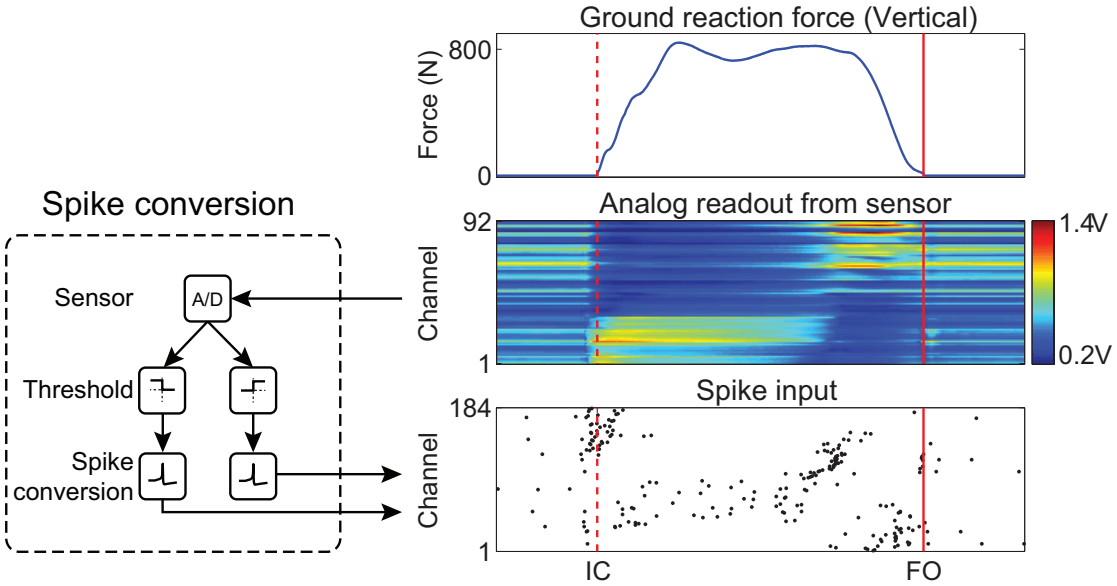


Figure 6.13: Analog signal to spike conversion.

The target signals were obtained using a force pressure plate (AMTI OR6-7-1000, Watertown, USA), which is considered the gold standard for detection of initial contact (IC) and foot off (FO) gait events [223]. Gait events were recorded as the time the vertical component of the ground reaction force crossed a calibrated threshold of 5 N on the rising edge for IC and on the falling edge for FO.

6.5.2 Walking on 10 m track

One healthy subject was given instructions to perform 90 valid walks on a 10 m track at a self-selected pace. A force pressure plate was placed at the mid-point of the track. For a walk to be considered as valid, the right foot of the subject had to land fully within the force pressure plate. The right shoe of the subject was fitted with the insole pressure sensor. Both systems were synchronized and sampled at 1,000 Hz for subsequent offline analysis. Data was segmented into individual steps starting 300 ms before IC and ending 300 ms after FO.

Two neurons were synthesized to output spikes corresponding to the IC and FO gait events. Sixty steps were used to compute the weights, while the solution was tested or cross validated on the remaining 30 steps. The neuron parameters used in this study are provided in Table 6.2. These values were close to optimal for a parameter search

Table 6.2: Neuron and optimization parameters for gait event detection

Parameter	**IC**	**FO**
Membrane time constant τ (ms)	70	190
Baseline threshold θ_0 (mV)	0.3	0.3
Threshold increase θ^+ (mV)	0.2	0.2
Recovery time constant τ_{rec} (ms)	140	380
Refractory period Δ^{abs} (ms)	3	3
Post-target potentiation time Δt (ms)	7	19
Post-target potentiation κ	1.2	1.25
$\boldsymbol{\lambda}_R$	(2.8, 1.2, 100, 100, 80)	

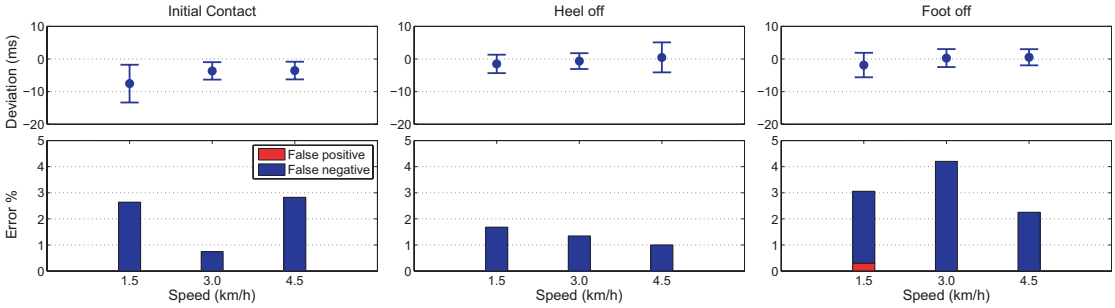


Figure 6.14: Gait event detection during continuous walking on a treadmill at 1.5, 3.0 and 4.5 km/h. Top row: Detection accuracy. Bottom row: Detection error. Results averaged from 4 cross validation trials. Error bars denote standard deviation.

space of $\tau = 10$ to 220 ms in steps of 30 ms, $\kappa = 1.05$ to 1.3 in steps of 0.05 and $(\boldsymbol{\lambda}_{R1} : \boldsymbol{\lambda}_{R2}) = (4 : 0)$ to $(0 : 4)$ in steps of 0.4. False positives/negatives were defined as additional/missing spikes within 50 ms of a gait event.

Our CONE-R technique computed gait event times for IC and FO that were 2.0 ± 6.7 ms and 8 ± 17 ms earlier, respectively, than recorded by the force plate. All events were detected accurately with no false positives/negatives. These results are comparable to other instrumented insole systems [224]. However, since our approach uses minimal pressure resolution and uncalibrated sensors, it has the potential to be a cheaper and more energy efficient system.

6.5.3 Walking on treadmill

To investigate the applicability of gait event detection in a near natural setting, the experiment was repeated with the subject walking on a treadmill at speeds of 1.5, 3.0 and 4.5 km/h for a total of 4 minutes. An Inertial-Magnetic Unit (IMU) system operating

at 100 Hz was used to provide the reference signals for IC, FO and Heel Off (HO) events [225]. In this experiment, the mapping function involved multiple output spikes because the inputs were no longer segmented into individual steps. Neuron parameters for the HO event were the same as those used for the FO (see Table 6.2). A 4-fold cross validation analysis was performed for each combination of speed and gait event.

The results show that using the CONE-R algorithm the neuron successfully detected more than 95 % of the gait events (Fig. 6.14). In most cases it had a temporal precision of 10 ms or 1 sampling period of the reference system. These results demonstrate the utility of spatiotemporal features to represent stimuli in a temporally precise and bandwidth efficient manner since this performance was achieved with minimal pressure information. Therefore, the CONE-R technique may be easily applicable in such situations.

6.5.4 Performance Comparison to SKIM

To benchmark the efficacy of CONE against a similar algorithm, we compared results on gait event detection obtained using SKIM. The weights for SKIM were computed using the alpha kernel with the same decay constants as presented in Table 6.2, and tested using the same 4-fold cross validation analysis. Selection of output threshold was based on a search of values ranging from 0.1 to 0.9 in steps of 0.1. The results represent the best detection accuracy for SKIM.

Fig. 6.15 illustrates a comparison of results obtained through SKIM and CONE. It is apparent from the comparison that SKIM produces significantly higher errors, most of them false positives, as compared to CONE. Temporal deviations of predicted gait events through SKIM were also noticeably larger. The results demonstrate that our CONE algorithm is capable of generating more accurate and robust solutions for this application.

6.6 Discussion

A novel Convex-Optimized-syNaptic-Efficacies technique to estimate weights of spiking neurons was developed. Its effectiveness was demonstrated through analysis of synthetic data and applicability to an experimental gait event detection problem.

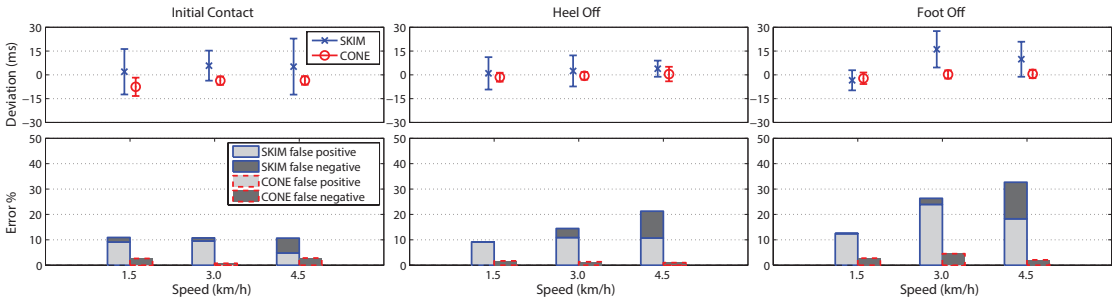


Figure 6.15: Results for the gait classification task using the SKIM algorithm (blue), as compared to CONE (red). Top row: Detection accuracy. Bottom row: Detection error. Results averaged from 4 cross validation trials. Error bars denote standard deviation.

6.6.1 CONE-X and CONE-R

The CONE technique computes neuron weights to facilitate temporally precise target spikes. Two formulations were presented: (1) CONE-X and (2) CONE-R. The CONE-X formulation has a dual role. It is able to compute a solution that meets all optimization constraints or it declares the problem infeasible. These characteristics are useful for parameter analysis of spiking neurons when inputs and outputs are known. The CONE-R formulation is well suited to encode spatiotemporal features with temporal precision because it is robust to noise corrupted training and test data. It may be useful to process experimental signals such as those obtained from neuromorphic event-based sensors [12] or biological spike recordings [123].

The CONE procedure has several desirable features. Neuron weights can be computed using standard optimization algorithms and tools because the problem is formulated in a convex optimization framework. Several standard optimization algorithms are capable of solving such problems in polynomial time [226] and are efficient for determining problem feasibility [227].

In addition, for noise free training data, CONE-X enables feasible solutions to be computed without parameter tuning. The regularization parameters, $\boldsymbol{\lambda}$, are useful to enhance solution robustness to noise. Our CONE formulation enables a user to have direct influence on the estimated weight distribution. This capability allows a user to design an algorithm that is robust to specific noise characteristics. Therefore, it has the potential to be applied to a wide variety of practical applications. The ability to

adjust sparsity of a solution also has important empirical implications for the design of spiking neural architectures. For example, a sparse solution translates to less synapses to implement in hardware, reducing the complexity and cost of a system.

The CONE-R approach has the potential to be applied to a large number of experimental situations. It may be used to synthesize weights for readout neurons of a liquid state machine [228], increasing computational capacity of the system [229]. Alternatively, CONE-R may be used to synthesize neurons to decode neural signals, which may aid in the control of neural-prostheses [230]. Also, it may be possible for such neurons to encode sensory signals as spikes for temporally precise neural stimulation to restore sensation [231].

A limitation of CONE is its inability to learn incrementally (i.e. online or adaptive analysis) as new information becomes available. This may limit its usefulness for time-varying spike association problems. However, this does not imply that CONE cannot be implemented in near real-time. A possible strategy for applying CONE to a time-varying application may be to update weights using a sliding window approach [232].

6.6.2 Suggestions on choice of parameters

We have demonstrated the effects of key parameters, λ_1, λ_2 and κ, for both the CONE-X and CONE-R formulations in §6.4.5 and §6.4.6. Here, we offer some intuitive suggestions about the other free parameters; namely, regularization constants λ_3, λ_4 and λ_5, membrane time constant τ, maximum threshold increase θ^+, threshold recovery constant τ_{rec}, absolute refractory period Δ^{abs}, post-target potentiation time Δt and the weight bounds lb and ub.

The regularization constants ensure the constraints are met (i.e. the neuron will spike only at specified times). Therefore, these parameters must have values much larger than λ_1 and λ_2. In our work, we used values 10 to 100 times larger. The values of τ, Δ^{abs}, lb and ub depend on the operational or biological specifications of the application. However, if no specific constraints are given, we recommend a value of τ that is the same order of magnitude as the length of the spatiotemporal pattern to be recognized since it influences the fading memory of the model. The value of Δ^{abs} should not be greater than the minimum inter-spike interval of the target spike train to ensure feasibility. The

weight bounds, lb, ub, can be arbitrarily small or large to expand or reduce the solution space. As for post-target potentiation step Δt, a value smaller than τ should be used to ensure feasibility. In our study, we used $\Delta t = 0.1\tau$. For threshold recovery and absolute refractory period, large θ^+ ($> \theta_0$) and τ_{rec} ($> \tau$) may help reduce false positives when a large number of input spikes are received. However, temporal accuracy for target spikes after a false positive/negative may be compromised, if the target spikes fall (do not fall) within the relative refractory period of the false positive (negative). For best temporal accuracy, we recommend using small values of τ_{rec} and θ^+ whenever possible.

6.6.3 Comparison with prior work

There are several learning algorithms for spatiotemporal pattern recognition and neuronal computation. Many of these approaches share a common configuration, they respond to multiple input spike trains by emitting spikes of their own.

The principal difference between CONE and Tempotron [145] is its ability to produce temporally precise spikes. This enables CONE's output to be compatible with subsequent layers of time sensitive neurons. In comparison to algorithms such as ReSuMe [148], Chronotron [149], SPAN [150] and PSD [151], CONE is unique because it utilizes well established optimization techniques to solve for weights. There are no learning parameters to tune. In the case of CONE-X, a solution will always be found if it exists, and it is also possible to demonstrate that no solution exists.

A comparable method to CONE is SKIM [209]. Nevertheless, SKIM produces a continuous output which, depending on the threshold chosen, may span several units in time. Moreover, careful selection of parameters (such as target spike width and magnitude) is needed with SKIM to avoid class imbalance when the target spike train is sparse; an issue discussed by the authors in [209]. CONE avoids these issues by generating solutions that do not require post-processing. In addition, the CONE formulation captures the spiking characteristic of the output neuron in more detail by accounting for refractory periods and allowing the membrane voltage to be between baseline and threshold when it is not time to spike. This leads to a larger solution space that may allow for more robust solutions. Lastly, CONE allows fine-control of solutions to improve robustness against specific noise types through parameters that

can be specified intuitively. However, we acknowledge that SKIM is generally easier to implement as there are very few parameters to tune. Moreover, SKIM does not require additional toolboxes to be installed and is thus highly attractive to users new to the field.

In addition to adjusting weights, it is also possible to utilize delays [233,234] or kernel properties [140] to synchronize the input spikes to produce temporally precise output spike trains. Delay and kernel adjustments may be easier to realize with digital hardware and are amenable to real-time implementation. Although the current update rules are typically heuristic, the general CONE formulation can be modified to be compatible to the problem, allowing the delays or kernel properties to be solved more efficiently.

6.6.4 Biological plausibility

It is difficult to relate biological processes observed in neurons to the sophisticated mathematics used for convex optimization. However, the nature of learning in neurons is not fully understood. It may be possible to heuristically achieve this optimization problem using multiple biologically plausible processes. For example, some variations of Hebbian learning, a well recognized biological process, have been shown to perform Principal Component Analysis and Independent Component Analysis [235].

We have shown weight distributions computed from noise corrupted training data resemble those observed in biology, particularly when sparsity is emphasized. In contrast, most implementations of the biologically inspired Spike Time Dependent Plasticity (STDP) approach are not capable of producing such weight distributions [210]. This finding suggests that CONE's objective of minimizing weight magnitudes is in agreement to biological requirements.

6.7 Conclusion

In this work, we proposed two novel CONE algorithms to estimate the weights of a spiking neuron that maps spatiotemporal input spike patterns to temporally precise output spikes. Their efficacies were demonstrated through simulated examples and application to experimental data for gait-event detection. Our techniques provide optimal solutions

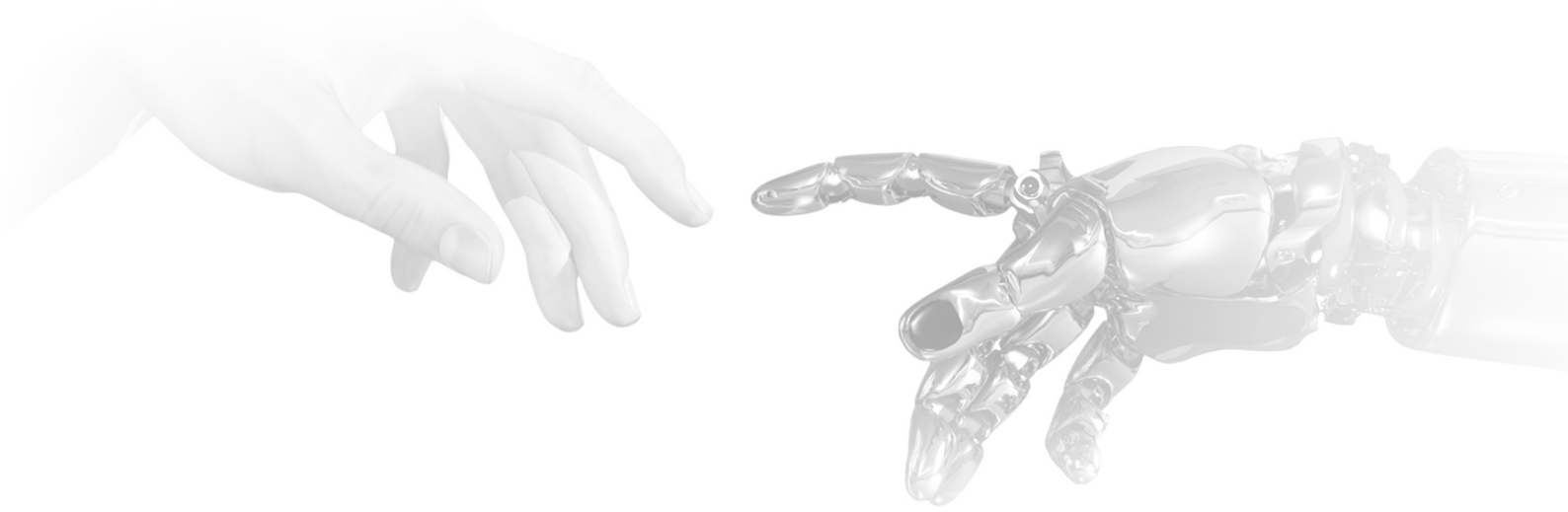
due to the convex nature of the problem formulation, which also increases memory capacity. CONE enables the user to generate solutions that are robust to specific noise profiles by influencing the weight distribution and membrane voltage trajectory. This clearly demonstrates that CONE is amenable to a variety of pattern recognition applications for complex biological processes.

7

Conclusions and Future Work

We must confess that it is in the human hand that we have the consummation of all perfection as an instrument.

Bell, 1833



7.1 Summary of contributions

The core idea presented in this dissertation represents a paradigm shift in the approach towards sensing, representing and processing of tactile signals. Instead of detailed pressure measurements at specific points, I argued for temporally precise detection of pressure changes across numerous sensing elements. Inspired by biology, I hypothesized that a spatiotemporal representation of tactile signals will be a powerful technique for efficient encoding of contact stimuli and driver of learning processes. The pursuit for a conclusive validation of my hypothesis ultimately resulted in this thesis.

I have produced numerous prototypes of tactile sensor arrays, from the early all-fabric design introduced in Chapter 3.4, to the durable insole-shaped sensor presented in Chapter 6.5, and finally the high-speed large sensor array presented in Chapter 4. A large part of the research effort was dedicated to the development of sensors and their accompanying circuitry and software because no commercially available sensor array could provide the spatial and temporal resolution sufficient to test my hypothesis. It soon became clear that developing a large scale tactile sensor array with the right mechanical characteristics and transducer properties would become a dissertation on its own. However, the impact of such a sensor would not be as apparent without theoretical justification and compatible signal processing algorithms. Sensor development was put on hold once I arrived at the Kilotaxel Kilohertz sensor array, as it provided the sensor density, sensitivity and response time that is sufficient to test my ideas.

Moving on, I combined experimental results obtained from the high speed tactile sensor array and finite element simulations to perform a comprehensive study demonstrating the advantages of capturing and representing touch as spatiotemporal patterns. In particular, I showed that dynamic contact involves features embedded at time scales much shorter than the duration of the event, justifying the need for sparse representations of high temporal precision. My proposed bio-inspired coding strategy permits rapid classification of these dynamic contact stimuli, a capability crucial for robotic manipulation and real-world interaction.

Finally, a new spiking neural classifier was developed to facilitate the application of spatiotemporal patterns of tactile signals to real-world problems. Specifically, the

CONE formulation has a much higher memory capacity compared to similar algorithms, enabling the learning of complex spike patterns while remaining robust to noise. The performance of CONE was established through simulations and demonstrated in physical experiments, where gait-events were detected with high temporal precision despite operating on uncalibrated sensors and minimal pressure resolution.

Based on the works presented in this dissertation, it is thus possible to ascertain that:

1. The spatiotemporal structure of tactile signals is highly informative for contact event discrimination, and should be preserved during sampling (Chapter 5).

2. These spatiotemporal features can be faithfully and efficiently captured by artificial tactile sensors using bio-inspired coding strategies (Chapter 4).

3. The spatiotemporal tactile patterns can be utilized for real-world applications through the use of learning algorithms implemented on spiking neural networks (Chapter 6).

7.2 Future work

A fully neuromorphic tactile sensing system is far from achieved, as the topics described in this thesis are mainly proof of concepts. There is much to be done on both the hardware and algorithm fronts to promote the benefits of neuromorphic tactile sensing.

Table 7.1 describes a proposed tactile sensing system that will best demonstrate the concept of neuromorphic tactile perception. At the transduction level, we propose to utilize a variety of transducers to increase the dimensionality of signals captured. Similar to biology, the arrangement and density of these transducers could be varied across the surface based on the type of interaction that is most likely to occur at each site. With advances in the fabrication of soft electronics, individual transducer elements should incorporate asynchronous logic in order to achieve lower power consumption and even higher temporal precision.

[1] an educated guess

Table 7.1: Comparison of biological and artificial tactile sensing to a proposed neuromorphic system.

	Biology	Artificial	Neuromorphic (proposed)
Number of taxels	∼100000[1]	∼1000	>10000
Transducer	Merkel cells, Meissner corpuscles, Pacinian corpuscles and Ruffini endings	Resistive / capacitive / piezoelectric / optical / MEMS	Multi-modal mix of transducers
Temporal precision	Up to 0.8 ms	Usually ≥10 ms	<1 ms
Output signal	Action potentials, asynchronous	Frames, synchronous	Events, ideally asynchronous
Feature extraction	Mechanoreceptors in compliant skin, somatosensory cortex	Mostly software running on microcontrollers	Transducers embedded in deformable substrate, spiking neural networks on neuromorphic hardware
Processing architecture	Distributed, massively parallel	Often centralized	Distributed, massively parallel

The physical properties of the sensor substrate also require further improvements in order to achieve the sensitivity and conformability comparable to biological skin. For instance, we observed in Section 2.1.5 that the FA-I and SA-I mechanoreceptors are situated close to the surface around the intermediate ridges. This is in stark contrast to the foam or rubber covered sensor arrays available today. Simulations have shown that the low-pass effect of a compliant cover causes aliasing, resulting in the inverse tactile problem [94] where surface interactions cannot be uniquely reconstructed based on measurements of pressure distribution gathered by sub-surface tactile sensors. However, a reliable method of producing stretchable and scalable circuitry has yet to exist, without which, it is difficult to construct a compliant sensing layer that is situated above a deformable structure. A natural next step would involve the exploration of suitable materials or sensing modality to overcome these limitations.

The multiple receptor types and their varying responses have been postulated to be a form of low level feature extraction [40]. We have presented a real-time phenomenological model of the FA-I mechanoreceptor in [236] which appears to improve the efficiency of feature representations. However, the model only simulates the first-order properties of the mechanoreceptor, and additional refinement is needed to reproduce the more

complex characteristics of the afferent. Moreover, it would be prudent to include the simulation of SA-I, SA-II and FA-II receptors to increase biological realism and signal dimensionality. An investigative effort is also needed to understand which are the more salient mechanoreceptor properties to preserve, in order to strike a balance between the complexity of implementation and biological plausibility.

It is apparent that tactile pattern recognition using spatiotemporal features is a new field, and the algorithms to process these unique representations are still in their infancy. In this thesis, we proposed to utilize spiking neural networks for signal processing, as they are inherently suited to dealing with the stochastic nature of spike trains. However, spiking neural networks simulated on conventional computing hardware will not achieve the real-time performance and low power consumption promised by neuromorphic systems. Ideally, the proposed algorithms should be implemented on multi-purpose event based computing platforms such as the SpiNNaker [155] and TrueNorth [14] to leverage on the massive parallelism and computational efficiency afforded by these systems. Ultimately, the efficacy of the neuromorphic approach will only be appreciated through the solving of real-life problems.

Bibliography

[1] R. S. Dahiya, G. Metta, M. Valle, and G. Sandini, "Tactile sensing—from humans to humanoids," *IEEE Trans. Robot.*, vol. 26, no. 1, pp. 1–20, 2010.

[2] M. I. Tiwana, S. J. Redmond, and N. H. Lovell, "A review of tactile sensing technologies with applications in biomedical engineering," *Sensor Actuat. A-Phys.*, vol. 179, pp. 17–31, 2012.

[3] H. Yousef, M. Boukallel, and K. Althoefer, "Tactile sensing for dexterous in-hand manipulation in robotics: a review," *Sensor Actuat. A-Phys.*, vol. 167, no. 2, pp. 171–187, 2011.

[4] R. S. Johansson and Å. B. Vallbo, "Tactile sensibility in the human hand: relative and absolute densities of four types of mechanoreceptive units in glabrous skin." *J. Physiol.*, vol. 286, no. 1, pp. 283–300, 1979.

[5] R. S. Johansson and G. Westling, "Roles of glabrous skin receptors and sensorimotor memory in automatic control of precision grip when lifting rougher or more slippery objects," *Exp. Brain Res.*, vol. 56, no. 3, pp. 550–564, 1984.

[6] ——, "Signals in tactile afferents from the fingers eliciting adaptive motor responses during precision grip," *Exp. Brain Res.*, vol. 66, no. 1, pp. 141–154, 1987.

[7] R. S. Johansson, C. Häger, and R. Riso, "Somatosensory control of precision grip during unpredictable pulling loads," *Exp. Brain Res.*, vol. 89, no. 1, pp. 192–203, 1992.

[8] C. Häger-Ross, K. J. Cole, and R. S. Johansson, "Grip-force responses to unanticipated object loading: load direction reveals body-and gravity-referenced intrinsic task variables," *Exp. Brain Res.*, vol. 110, no. 1, pp. 142–150, 1996.

[9] V. G. Macefield, C. Häger-Ross, and R. S. Johansson, "Control of grip force during restraint of an object held between finger and thumb: responses of cutaneous afferents from the digits," *Exp. Brain Res.*, vol. 108, no. 1, pp. 155–171, 1996.

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

[11] R. VanRullen, R. Guyonneau, and S. J. Thorpe, "Spike times make sense," *Trends Neurosci.*, vol. 28, no. 1, pp. 1–4, 2005.

[12] S.-C. Liu and T. Delbruck, "Neuromorphic sensory systems," *Curr. Opin. Neurobiol.*, vol. 20, no. 3, pp. 288–295, 2010.

[13] W. Maass, "Networks of spiking neurons: the third generation of neural network models," *Neural networks*, vol. 10, no. 9, pp. 1659–1671, 1997.

[14] P. A. Merolla, J. V. Arthur, R. Alvarez-Icaza, A. S. Cassidy, J. Sawada, F. Akopyan, B. L. Jackson, N. Imam, C. Guo, Y. Nakamura *et al.*, "A million spiking-neuron integrated circuit with a scalable communication network and interface," *Science*, vol. 345, no. 6197, pp. 668–673, 2014.

[15] L. Osborn, W. W. Lee, R. Kaliki, and N. Thakor, "Tactile feedback in upper limb prosthetic devices using flexible textile force sensors," in *Proc. IEEE RAS EMBS Int. Conf. Biomed. Robot. Biomechatron.* Sao Paulo, Brazil: IEEE, 2014, pp. 114–119.

[16] E. B. Goldstein, *Sensation and perception.* Cengage Learning, 2013.

[17] J. B. van Erp and H. A. van Veen, "Touch down: the effect of artificial touch cues on orientation in microgravity," *Neurosci. Lett.*, vol. 404, no. 1, pp. 78–82, 2006.

[18] M. Knibestöl and Å. Vallbo, "Single unit analysis of mechanoreceptor activity from the human glabrous skin," *Acta Physiol. Scand.*, vol. 80, no. 2, pp. 178–195, 1970.

[19] S. Weinstein, "Intensive and extensive aspects of tactile sensitivity as a function of body part, sex and laterality," in *1st Int'l Symp. on the Skin Senses*. Tallahassee, FL: Springfield III, Mar. 1966.

[20] R. L. Klatzky and S. J. Lederman, "Touch," in *Handbook of psychology*. Wiley Online Library, 2003.

[21] J. C. Craig and X. Baihua, "Temporal order and tactile patterns," *Perception & psychophysics*, vol. 47, no. 1, pp. 22–34, 1990.

[22] L. A. Jones and S. J. Lederman, *Human hand function*. Oxford University Press, 2006.

[23] R. Johansson, Å. Vallbo, and G. Westling, "Thresholds of mechanosensitive afferents in the human hand as measured with von frey hairs," *Brain Res.*, vol. 184, no. 2, pp. 343–351, 1980.

[24] R. S. Johansson and J. R. Flanagan, "Coding and use of tactile signals from the fingertips in object manipulation tasks," *Nat. Rev. Neurosci.*, vol. 10, no. 5, pp. 345–359, 2009.

[25] J. W. Bisley, A. W. Goodwin, and H. E. Wheat, "Slowly adapting type I afferents from the sides and end of the finger respond to stimuli on the center of the fingerpad," *J. Neurophysiol.*, vol. 84, no. 1, pp. 57–64, 2000.

[26] I. Birznieks, P. Jenmalm, A. W. Goodwin, and R. S. Johansson, "Encoding of direction of fingertip forces by human tactile afferents," *J. Neurosci.*, vol. 21, no. 20, pp. 8222–8237, 2001.

[27] P. Jenmalm, I. Birznieks, A. W. Goodwin, and R. S. Johansson, "Influence of object shape on responses of human tactile afferents under conditions characteristic of manipulation," *Eur. J. Neurosci.*, vol. 18, no. 1, pp. 164–176, 2003.

[28] C. W. Eurich and H. Schwegler, "Coarse coding: calculation of the resolution achieved by a population of large receptive field neurons," *Biol. Cybern.*, vol. 76, no. 5, pp. 357–363, 1997.

[29] K. Dandekar, B. I. Raju, and M. A. Srinivasan, "3-D finite-element models of human and monkey fingertips to investigate the mechanics of tactile sense," *J. Biomech. Eng.-T ASME*, vol. 125, no. 5, pp. 682–691, 2003.

[30] J. Z. Wu, D. E. Welcome, and R. G. Dong, "Three-dimensional finite element simulations of the mechanical response of the fingertip to static and dynamic compressions," *Comput. Methods Biomech. Biomed. Engin.*, vol. 9, no. 1, pp. 55–63, 2006.

[31] D. R. Lesniak and G. J. Gerling, "Predicting SA-I mechanoreceptor spike times with a skin-neuron model," *Math Biosci.*, vol. 220, no. 1, pp. 15–23, 2009.

[32] M. M. Merzenich, R. J. Nelson, J. H. Kaas, M. P. Stryker, W. M. Jenkins, J. M. Zook, M. S. Cynader, and A. Schoppmann, "Variability in hand surface representations in areas 3b and 1 in adult owl and squirrel monkeys," *J. Comp. Neurol.*, vol. 258, no. 2, pp. 281–296, 1987.

[33] M. M. Merzenich and W. M. Jenkins, "Reorganization of cortical representations of the hand following alterations of skin inputs induced by nerve injury, skin island transfers, and experience," *J. Hand. Ther.*, vol. 6, no. 2, pp. 89–104, 1993.

[34] F. A. Geldard and C. E. Sherrick, "The cutaneous "rabbit": A perceptual illusion," *Science*, vol. 178, no. 4057, pp. 178–179, 1972.

[35] R. W. Cholewiak, "The perception of tactile distance: Influences of body site, space, and time," *Perception*, vol. 28, no. 7, pp. 851–875, 1999.

[36] T. Seizova-Cajic and J. L. Taylor, "Somatosensory space abridged: Rapid change in tactile localization using a motion stimulus," *PLoS One*, vol. 9, no. 3, p. e90892, 2014.

[37] M. Rowe, A. Turman, G. Murray, and H. Zhang, "Parallel processing in somatosensory areas i and ii of the cerebral cortex," in *Somesthesis and the Neurobiology of the Somatosensory Cortex*. Springer, 1996, pp. 197–211.

[38] R. S. Johansson and I. Birznieks, "First spikes in ensembles of human tactile afferents code complex spatial fingertip events," *Nat. Neurosci.*, vol. 7, no. 2, pp. 170–177, 2004.

[39] R. Johansson and Å. Vallbo, "Spatial properties of the population of mechanoreceptive units in the glabrous skin of the human hand," *Brain Res.*, vol. 184, no. 2, pp. 353–366, 1980.

[40] J. A. Pruszynski and R. S. Johansson, "Edge-orientation processing in first-order tactile neurons," *Nat. Neurosci.*, vol. 17, no. 10, pp. 1404–1409, 2014.

[41] C. M. Oddo, S. Raspopovic, F. Artoni, A. Mazzoni, G. Spigler, F. Petrini, F. Giambattistelli, F. Vecchio, F. Miraglia, L. Zollo *et al.*, "Intraneural stimulation elicits discrimination of textural features by artificial fingertip in intact and amputee humans," *ELife*, vol. 5, p. e09148, 2016.

[42] M. A. Harvey, H. P. Saal, J. F. Dammann III, and S. J. Bensmaia, "Multiplexing stimulus information through rate and temporal codes in primate somatosensory cortex," *PLoS Biol.*, vol. 11, no. 5, p. e1001558, 2013.

[43] C. Escoffier, J. de Rigal, A. Rochefort, R. Vasselet, J.-L. Lévêque, and P. G. Agache, "Age-related mechanical properties of human skin: an in vivo study." *J. Invest. Dermatol.*, vol. 93, no. 3, pp. 353–357, 1989.

[44] G. J. Gerling and G. W. Thomas, "The effect of fingertip microstructures on tactile edge perception," in *1st Joint Eurohaptics Conf. Haptic Interfaces for Virtual Environment and Teleoperator Syst. World Haptics Conf.* Pisa, Italy: IEEE, Mar. 2005, pp. 63–72.

[45] S. J. Lederman and D. Pawluk, "Lessons from the study of biological touch for robotic tactile sensing," *Adv. Tactile Sens. Robot*, vol. 5, pp. 151–192, 1992.

[46] J. Scheibert, S. Leurent, A. Prevost, and G. Debregeas, "The role of fingerprints in the coding of tactile information probed with a biomimetic sensor," *Science*, vol. 323, no. 5920, pp. 1503–1506, 2009.

[47] R. S. Dahiya and M. Valle, *Robotic Tactile Sensing: Technologies and System.* Springer, 2013.

[48] N. Cauna, "Nature and functions of the papillary ridges of the digital skin," *Anat. Rec.*, vol. 119, no. 4, pp. 449–468, 1954.

[49] J. R. Phillips and K. O. Johnson, "Tactile spatial resolution. iii. a continuum mechanics model of skin predicting mechanoreceptor responses to bars, edges, and gratings," *J. Neurophysiol.*, vol. 46, no. 6, pp. 1204–1225, 1981.

[50] T. Maeno, K. Kobayashi, and N. Yamazaki, "Relationship between the structure of human finger tissue and the location of tactile receptors." *JSME Int. J.*, vol. 41, no. 1, pp. 94–100, 1998.

[51] E. R. Kandel, J. H. Schwartz, T. M. Jessell *et al.*, *Principles of neural science.* McGraw-Hill New York, 2000, vol. 4.

[52] B. Siciliano and O. Khatib, *Springer handbook of robotics.* Springer, 2008.

[53] H. Liu, P. Meusel, and G. Hirzinger, "A tactile sensing system for the DLR three-finger robot hand," in *Int. Symp. Measurement and Control in Robot.*, Slovakia, Jun. 1995, pp. 91–96.

[54] M. A. Diftler, C. Culbert, R. Ambrose, R. Platt Jr, and W. Bluethmann, "Evolution of the NASA/DARPA robonaut control system," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 2. Taipei, Taiwan: IEEE, Sep. 2003, pp. 2543–2548.

[55] D. Bloor, K. Donnelly, P. Hands, P. Laughlin, and D. Lussey, "A metal–polymer composite with unusual properties," *J. Phys. D Appl. Phys.*, vol. 38, no. 16, p. 2851, 2005.

[56] R. Walkler, "Developments in dextrous hands for advanced robotic applications," in *Proc. 6th Biannual World Autom. Congr.*, Seville, Spain, 2004, pp. 123–128.

[57] V. Maheshwari and R. F. Saraf, "High-resolution thin-film device to sense texture by touch," *Science*, vol. 312, no. 5779, pp. 1501–1504, 2006.

[58] H.-H. Chou, A. Nguyen, A. Chortos, J. W. To, C. Lu, J. Mei, T. Kurosawa, W.-G. Bae, J. B.-H. Tok, and Z. Bao, "A chameleon-inspired stretchable electronic skin with interactive colour changing controlled by tactile sensing," *Nat. Comm.*, vol. 6, no. 8011, 2015.

[59] B. L. Gray and R. S. Fearing, "A surface micromachined microtactile sensor array," in *Int. Conf. Robot. Autom.*, vol. 1. Minneapolis, MN: IEEE, Apr. 1996, pp. 1–6.

[60] Pressure Profile Systems. (2016) Capacitive tactile sensors in robotics. [Online]. Available: http://www.pressureprofile.com/oem-robotics/

[61] Analog Devices. (2009) AD7147: Captouch® programmable controller for single-electrode capacitance sensors. [Online]. Available: http://www.analog.com/static/imported-files/Data_Sheets/AD7147.pdf

[62] C. Chorley, C. Melhuish, T. Pipe, and J. Rossiter, "Development of a tactile sensor based on biologically inspired edge encoding," in *Int. Conf. Adv. Robot.* Munich, Germany: IEEE, Jun. 2009, pp. 1–6.

[63] R. Li and E. H. Adelson, "Sensing and recognizing surface textures using a Gelsight sensor," in *Conf. Comput. Vision and Pattern Recognition.* Portland, OR: IEEE, Jun. 2013, pp. 1241–1247.

[64] Y. Ohmura, Y. Kuniyoshi, and A. Nagakubo, "Conformable and scalable tactile sensor skin for curved surfaces," in *Proc. IEEE Int. Conf. Robot. Autom.* Orlando, FL: IEEE, May 2006, pp. 1348–1353.

[65] Y. Ohmura and Y. Kuniyoshi, "Humanoid robot which can lift a 30kg box by whole body contact and tactile feedback," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots and Syst.,*. San Diego, CA: IEEE, Oct. 2007, pp. 1136–1141.

[66] A. Shirinov and W. Schomburg, "Pressure sensor from a PVDF film," *Sensor Actuat. A-Phys.*, vol. 142, no. 1, pp. 48–55, 2008.

[67] S. Jin, "Tactile sensor comprising nanowires and method for making the same," USA Patent 6,286,226, 2001.

[68] W. Wu, X. Wen, and Z. L. Wang, "Taxel-addressable matrix of vertical-nanowire piezotronic transistors for active and adaptive tactile imaging," *Science*, vol. 340, no. 6135, pp. 952–957, 2013.

[69] J. A. Fishel, "Design and use of a biomimetic tactile microvibration sensor with human-like sensitivity and its application in texture discrimination using bayesian exploration," PhD Thesis, Dept. Biomed. Eng., Univ. Southern California, Los Angeles, 2012.

[70] Y. Tenzer, L. P. Jentoft, and R. D. Howe, "Inexpensive and easily customized tactile array sensors using MEMS barometers chips," *IEEE Robot. Autom. Mag*, vol. 21, no. 3, pp. 89–95, 2014.

[71] RightHand Robotics. (2016) Takktile sensors. [Online]. Available: http://www.labs.righthandrobotics.com/takktile-sensors

[72] Y. Kato, T. Mukai, T. Hayakawa, and T. Shibata, "Tactile sensor without wire and sensing element in the tactile region based on EIT method," in *IEEE Sens. J.* IEEE, 2007, pp. 792–795.

[73] D. Cheneler, M. Ward, and C. Anthony, "Bio-hybrid tactile sensor for the study of the role of mechanoreceptors in human tactile perception," *Microelectron. Eng.*, vol. 97, pp. 297–300, 2012.

[74] H. Muhammad, N. Hunt, R. Shelton, L. Grover, M. Ward, C. Oddo, C. Recchiuto, and L. Beccai, "Incorporation of novel MEMS tactile sensors into tissue engineered skin," in *4th Int. Conf. Bioinform. Biomed. Eng.* Chengdu, China: IEEE, Jun. 2010, pp. 1–4.

[75] E. Buselli, A. M. Smith, L. M. Grover, A. Levi, R. Allman, V. Mattoli, A. Menciassi, and L. Beccai, "Development and characterization of a bio-hybrid

skin-like stretchable electrode," *Microelectron. Eng.*, vol. 88, no. 8, pp. 1676–1680, 2011.

[76] M. Kaltenbrunner, T. Sekitani, J. Reeder, T. Yokota, K. Kuribara, T. Tokuhara, M. Drack, R. Schwödiauer, I. Graz, S. Bauer-Gogonea *et al.*, "An ultra-lightweight design for imperceptible plastic electronics," *Nature*, vol. 499, no. 7459, pp. 458–463, 2013.

[77] J. Roh, A. Freed, Y. Mann, and D. Wessel, "Robust and reliable fabric, piezoresistive multitouch sensing surfaces for musical controllers," in *Proc. Int. Conf. New Interfaces for Musical Expression*, Oslo, Norway, Jun. 2011, pp. 393–398.

[78] K. Takei, T. Takahashi, J. C. Ho, H. Ko, A. G. Gillies, P. W. Leu, R. S. Fearing, and A. Javey, "Nanowire active-matrix circuitry for low-voltage macroscale artificial skin," *Nat. Mater.*, vol. 9, no. 10, pp. 821–826, 2010.

[79] J. Engel, J. Chen, Z. Fan, and C. Liu, "Polymer micromachined multimodal tactile sensors," *Sensor Actuat. A-Phys.*, vol. 117, no. 1, pp. 50–61, 2005.

[80] J. Park, M. Kim, Y. Lee, H. S. Lee, and H. Ko, "Fingertip skin–inspired microstructured ferroelectric skins discriminate static/dynamic pressure and temperature stimuli," *Sci. Adv.*, vol. 1, no. 9, p. e1500661, 2015.

[81] J. Kim, M. Lee, H. J. Shim, R. Ghaffari, H. R. Cho, D. Son, Y. H. Jung, M. Soh, C. Choi, S. Jung, K. Chu, D. Jeon, S.-T. Lee, J. H. Kim, S. H. Choi, T. Hyeon, and D.-H. Kim, "Stretchable silicon nanoribbon electronics for skin prosthesis," *Nat. Comm.*, vol. 5, p. 5747, Dec. 2014.

[82] J. A. Rogers, T. Someya, and Y. Huang, "Materials and mechanics for stretchable electronics," *Science*, vol. 327, no. 5973, pp. 1603–1607, 2010.

[83] T. Sekitani, H. Nakajima, H. Maeda, T. Fukushima, T. Aida, K. Hata, and T. Someya, "Stretchable active-matrix organic light-emitting diode display using printable elastic conductors," *Nat. Mater.*, vol. 8, no. 6, pp. 494–499, 2009.

[84] DuPont. (2016) Making smart clothing easier to design, manufacture, wash and wear. [Online]. Available: http://www.dupont.com/products-and-services/electronic-electrical-materials/printed-electronics/products/stetchable-inks-for-wearable-electronics.html

[85] J. M. Nassar, M. D. Cordero, A. T. Kutbee, M. A. Karimi, G. A. T. Sevilla, A. M. Hussain, A. Shamim, and M. M. Hussain, "Paper skin multisensory platform for simultaneous environmental monitoring," *Adv. Mater.*, vol. 1, no. 1, pp. 1–14, February 2016.

[86] R. A. Nixon and A. Yuan, *Cytoskeleton of the nervous system.* Springer, 2011.

[87] M. Inaba, Y. Hoshino, K. Nagasaka, T. Ninomiya, S. Kagami, and H. Inoue, "A full-body tactile sensor suit using electrically conductive fabric and strings," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots and Syst.,*, vol. 2. Osaka, Japan: IEEE, Nov. 1996, pp. 450–457.

[88] T. Someya, Y. Kato, T. Sekitani, S. Iba, Y. Noguchi, Y. Murase, H. Kawaguchi, and T. Sakurai, "Conformable, flexible, large-area networks of pressure and thermal sensors with organic transistor active matrixes," *P. Natl. Acad. Sci. USA*, vol. 102, no. 35, pp. 12 321–12 325, 2005.

[89] O. Oballe-Peinado, J. A. Hidalgo-Lopez, J. A. Sanchez-Duran, J. Castellanos-Ramos, and F. Vidal-Verdú, "Architecture of a tactile sensor suite for artificial hands based on FPGAs," in *Proc. IEEE RAS EMBS Int. Conf. Biomed. Robot. Biomechatron.* Rome, Italy: IEEE, Jun. 2012, pp. 112–117.

[90] M. Shimojo, T. Araki, A. Ming, and M. Ishikawa, "A high-speed mesh of tactile sensors fitting arbitrary surfaces," *IEEE Sens. J.*, vol. 10, no. 4, pp. 822–830, 2010.

[91] M. Nilsson, "Tactile sensors and other distributed sensors with minimal wiring complexity," *IEEE-ASME T. Mech.*, vol. 5, no. 3, pp. 253–257, 2000.

[92] H. Shinoda and H. Oasa, "Wireless tactile sensing element using stress-sensitive resonator," *IEEE/ASME Trans. Mechatron.*, vol. 5, no. 3, pp. 258–265, 2000.

[93] H. Shinoda, N. Asamura, M. Hakozaki, and X. Wang, "Two-dimensional signal transmission technology for robotics," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 3. Taipei, Taiwan: IEEE, Sep. 2003, pp. 3207–3212.

[94] M. H. Lee and H. R. Nicholls, "Review article tactile sensing for mechatronics, a state of the art survey," *Mechatronics*, vol. 9, no. 1, pp. 1–31, 1999.

[95] G. Cannata, M. Maggiali, G. Metta, and G. Sandini, "An embedded artificial skin for humanoid robots," in *IEEE Int. Conf. Multisensor Fusion and Integration for Intell. Syst.* Seoul, Korea: IEEE, Aug. 2008, pp. 434–438.

[96] P. Mittendorfer and G. Cheng, "Integrating discrete force cells into multi-modal artificial skin," in *IEEE-RAS Int. Conf. Humanoid Robots.* Osaka, Japan: IEEE, Nov. 2012, pp. 847–852.

[97] W. Fukui, F. Kobayashi, F. Kojima, H. Nakamoto, N. Imamura, T. Maeda, and H. Shirasawa, "High-speed tactile sensing for array-type tactile sensor and object manipulation based on tactile information," *J. Robotic.*, vol. 2011, 2011.

[98] S. Decherchi, P. Gastaldo, R. S. Dahiya, M. Valle, and R. Zunino, "Tactile-data classification of contact materials using computational intelligence," *IEEE Trans. Robot.*, vol. 27, no. 3, pp. 635–639, 2011.

[99] A. Jimenez, A. Soembagijo, D. Reynaerts, H. Van Brussel, R. Ceres, and J. Pons, "Featureless classification of tactile contacts in a gripper using neural networks," *Sens. Actuators, A*, vol. 62, no. 1, pp. 488–491, 1997.

[100] S. Aoyagi, T. Tanaka, and M. Minami, "Recognition of contact state of four layers arrayed type tactile sensor by using neural network," in *IEEE Int. Conf. Inform. Acquisition.* Shandong, China: IEEE, Aug. 2006, pp. 393–397.

[101] M. Johnsson and C. Balkenius, "Sense of touch in robots with self-organizing maps," *IEEE Trans. Robot.*, vol. 27, no. 3, pp. 498–507, 2011.

[102] N. Jamali and C. Sammut, "Majority voting: material classification by tactile sensing using surface texture," *IEEE Trans. Robot.*, vol. 27, no. 3, pp. 508–521, 2011.

[103] C. Schürmann, M. Schöpfer, R. Haschke, and H. Ritter, "A high-speed tactile sensor for slip detection," in *Towards Service Robots for Everyday Environments.* Springer, 2012, pp. 403–415.

[104] G. Spigler, C. M. Oddo, and M. C. Carrozza, "Soft-neuromorphic artificial touch for applications in neuro-robotics," in *Proc. IEEE RAS EMBS Int. Conf. Biomed. Robot. Biomechatron.* Rome, Italy: IEEE, Jun. 2012, pp. 1913–1918.

[105] U. B. Rongala, A. Mazzoni, and C. M. Oddo, "Neuromorphic artificial touch for categorization of naturalistic textures." *IEEE Trans. Neural Netw. Learn. Syst.*, vol. PP, September 2015.

[106] L. L. Bologna, J. Pinoteau, R. Brasselet, M. Maggiali, and A. Arleo, "Encoding/decoding of first and second order tactile afferents in a neurorobotic application," *J. Physiology-Paris*, vol. 105, no. 1, pp. 25–35, 2011.

[107] S. McGregor, D. Polani, and K. Dautenhahn, "Generation of tactile maps for artificial skin," *PLoS One*, vol. 6, no. 11, p. e26561, 2011.

[108] P. Mittendorfer and G. Cheng, "3D surface reconstruction for robotic body parts with artificial skins," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots and Syst.,*. Portugal: IEEE, Oct. 2012, pp. 4505–4510.

[109] S. B. Furber, D. R. Lester, L. A. Plana, J. D. Garside, E. Painkras, S. Temple, and A. D. Brown, "Overview of the SpiNNaker system architecture," *IEEE Trans. Comput.*, vol. 62, no. 12, pp. 2454–2467, 2013.

[110] P. A. Merolla, J. V. Arthur, B. E. Shi, and K. A. Boahen, "Expandable networks for neuromorphic chips," *IEEE Trans. Circuits Syst.*, vol. 54, no. 2, pp. 301–311, 2007.

[111] J.-s. Seo, B. Brezzo, Y. Liu, B. D. Parker, S. K. Esser, R. K. Montoye, B. Rajendran, J. A. Tierno, L. Chang, D. S. Modha *et al.*, "A 45nm CMOS neuromorphic chip with a scalable architecture for learning in networks of spiking neurons," in *Custom Integrated Circuits Conf. (CICC).* San Jose, CA: IEEE, Sep. 2011, pp. 1–4.

[112] S. J. Thorpe and M. Imbert, "Biological constraints on connectionist modelling," in *Connectionism in Perspective*, R. Pfeifer, Ed. North-Holland, 1989, pp. 63–92.

[113] M. Meister and M. J. Berry II, "The neural code of the retina," *Neuron*, vol. 22, no. 3, pp. 435–450, 1999.

[114] T. Lu, L. Liang, and X. Wang, "Temporal and rate representations of time-varying signals in the auditory cortex of awake primates," *Nat. Neurosci.*, vol. 4, no. 11, pp. 1131–1138, 2001.

[115] J. Perez-Orive, O. Mazor, G. C. Turner, S. Cassenaer, R. I. Wilson, and G. Laurent, "Oscillations and sparsening of odor representations in the mushroom body," *Science*, vol. 297, no. 5580, pp. 359–365, 2002.

[116] D. A. Butts, C. Weng, J. Jin, C.-I. Yeh, N. A. Lesica, J.-M. Alonso, and G. B. Stanley, "Temporal precision in the neural code and the timescales of natural vision," *Nature*, vol. 449, no. 7158, pp. 92–95, 2007.

[117] L. A. Jeffress, "A place theory of sound localization." *J. Comp. Physiol. Psych.*, vol. 41, no. 1, p. 35, 1948.

[118] S. Thorpe and J. Gautrais, "Rank order coding," in *Comput. Neurosci.* Springer, 1998, pp. 113–118.

[119] G. Portelli, J. M. Barrett, G. Hilgen, T. Masquelier, A. Maccione, S. Di Marco, L. Berdondini, P. Kornprobst, and E. Sernagor, "Rank order coding: a retinal information decoding strategy revealed by large-scale multielectrode array retinal recordings," *eneuro*, vol. 3, no. 3, 2016.

[120] J. Hopfield, "Pattern recognition computation using action potential timing for stimulus representation," *Nature*, vol. 376, no. 6535, pp. 33–36, 1995.

[121] A. Delorme and S. J. Thorpe, "Face identification using one spike per neuron: resistance to image degradations," *Neural Networks*, vol. 14, no. 6, pp. 795–803, 2001.

[122] C. D. Brody and J. Hopfield, "Simple networks for spike-timing-based computation, with application to olfactory processing," *Neuron*, vol. 37, no. 5, pp. 843–852, 2003.

[123] R. Gütig, T. Gollisch, H. Sompolinsky, and M. Meister, "Computing complex visual features with retinal spike times," *PLoS One*, vol. 8, no. 1, p. e53063, 2013.

[124] O. Booij *et al.*, "A gradient descent rule for spiking neurons emitting multiple spikes," *Inform. Process. Lett.*, vol. 95, no. 6, pp. 552–558, 2005.

[125] W. Maass, T. Natschläger, and H. Markram, "Computational models for generic cortical microcircuits," in *Computational Neuroscience: A Comprehensive Approach*, J. Feng, Ed. Chapman & Hall/CRC, Boca Raton, 2004, ch. 18, pp. 575–605.

[126] B. Nessler, M. Pfeiffer, and W. Maass, "STDP enables spiking neurons to detect hidden causes of their inputs." in *Neural Inform. Process. Syst.*, Vancouver, BC, Dec. 2009, pp. 1357–1365.

[127] R. Legenstein, D. Pecevski, and W. Maass, "A learning theory for reward-modulated spike-timing-dependent plasticity with application to biofeedback," *PLoS Comp. Biol.*, vol. 4, no. 10, p. e1000180, 2008.

[128] J. Iglesias and A. E. Villa, "Emergence of preferred firing sequences in large spiking neural networks during simulated neuronal development," *Int. J. Neural Syst.*, vol. 18, no. 04, pp. 267–277, 2008.

[129] B. Schrauwen, M. D'Haene, D. Verstraeten, and J. V. Campenhout, "Compact hardware liquid state machines on FPGA for real-time speech recognition," *Neural Networks*, vol. 21, no. 2, pp. 511–523, 2008.

[130] H. Burgsteiner, M. Kröll, A. Leopold, and G. Steinbauer, "Movement prediction from real-world images using a liquid state machine," *Appl. Intell.*, vol. 26, no. 2, pp. 99–109, 2007.

[131] K. L. Rice, M. A. Bhuiyan, T. M. Taha, C. N. Vutsinas, and M. C. Smith, "FPGA implementation of Izhikevich spiking neural networks for character recognition,"

in *Int. Conf. on Reconfigurable Computing and FPGAs*. Quintana Roo, Mexico: IEEE, Dec. 2009, pp. 451–456.

[132] R. Serrano-Gotarredona, M. Oster, P. Lichtsteiner, A. Linares-Barranco, R. Paz-Vicente, F. Gómez-Rodríguez, L. Camuñas-Mesa, R. Berner, M. Rivas-Pérez, T. Delbruck *et al.*, "CAVIAR: A 45k neuron, 5M synapse, 12G connects/s AER hardware sensory–processing–learning–actuating system for high-speed visual object recognition and tracking," *IEEE Trans. Neural Netw.*, vol. 20, no. 9, pp. 1417–1438, 2009.

[133] J. A. Pérez-Carrasco, C. Serrano, B. Acha, T. Serrano-Gotarredona, and B. Linares-Barranco, "Spike-based convolutional network for real-time processing," in *20th Int. Conf. on Pattern Recognition*. Istanbul, Turkey: IEEE, Aug. 2010, pp. 3085–3088.

[134] J. A. Pérez-Carrasco, B. Acha, C. Serrano, L. Camuñas-Mesa, T. Serrano-Gotarredona, and B. Linares-Barranco, "Fast vision through frameless event-based sensing and convolutional processing: Application to texture recognition," *IEEE Trans. Neural Netw.*, vol. 21, no. 4, pp. 609–620, 2010.

[135] B. Widrow and M. A. Lehr, "30 years of adaptive neural networks: perceptron, madaline, and backpropagation," *Proc. IEEE*, vol. 78, no. 9, pp. 1415–1442, 1990.

[136] S. Song, K. D. Miller, and L. F. Abbott, "Competitive hebbian learning through spike-timing-dependent synaptic plasticity," *Nat. Neurosci.*, vol. 3, no. 9, pp. 919–926, 2000.

[137] D. E. Feldman, "The spike-timing dependence of plasticity," *Neuron*, vol. 75, no. 4, pp. 556–571, 2012.

[138] S. Hussain, A. Basu, M. Wang, and T. J. Hamilton, "Deltron: Neuromorphic architectures for delay based learning," in *IEEE Asia Pacific Conf. on Circuits and Syst.* Kaoshiung, Taiwan: IEEE, Dec. 2012, pp. 304–307.

[139] G. Orchard, C. Meyer, R. Etienne-Cummings, C. Posch, N. Thakor, and R. Benosman, "HFirst: a temporal approach to object recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 10, pp. 2028–2040, 2015.

[140] S. Afshar, L. George, J. Tapson, A. A. van Schaik, and T. J. Hamilton, "Racing to learn: statistical inference and learning in a single spiking neuron with adaptive kernels," *Front. Neurosci.*, vol. 8, no. 377, pp. 1–18, Nov. 2014.

[141] T. Masquelier, R. Guyonneau, and S. J. Thorpe, "Spike timing dependent plasticity finds the start of repeating patterns in continuous spike trains," *PLoS One*, vol. 3, no. 1, p. e1377, 2008.

[142] T. Masquelier and S. J. Thorpe, "Unsupervised learning of visual features through spike timing dependent plasticity," *PLoS Comput. Biol.*, vol. 3, no. 2, p. e31, 2007.

[143] E. M. Izhikevich, "Solving the distal reward problem through linkage of STDP and dopamine signaling," *Cereb. Cortex*, vol. 17, no. 10, pp. 2443–2452, 2007.

[144] R. V. Florian, "Reinforcement learning through modulation of spike-timing-dependent synaptic plasticity," *Neural Comput.*, vol. 19, no. 6, pp. 1468–1502, 2007.

[145] R. Gütig and H. Sompolinsky, "The tempotron: a neuron that learns spike timing–based decisions," *Nat. Neurosci.*, vol. 9, no. 3, pp. 420–428, 2006.

[146] H. Tang, Q. Yu, and K. C. Tan, "Learning real-world stimuli by single-spike coding and tempotron rule," in *Int. Joint Conf. Neural Network*. Brisband, Australia: IEEE, Jun. 2012, pp. 1–6.

[147] R. Gütig, "To spike, or when to spike?" *Curr. Opin. Neurobiol.*, vol. 25, pp. 134–139, Apr. 2014.

[148] F. Ponulak and A. Kasiński, "Supervised learning in spiking neural networks with ReSuMe: sequence learning, classification, and spike shifting," *Neural Comput.*, vol. 22, no. 2, pp. 467–510, 2010.

[149] R. V. Florian, "The chronotron: A neuron that learns to fire temporally precise spike patterns," *PLoS One*, vol. 7, no. 8, p. e40233, 2012.

[150] A. Mohemmed, S. Schliebs, S. Matsuda, and N. Kasabov, "SPAN: Spike pattern association neuron for learning spatio-temporal spike patterns," *Int. J. Neural Syst.*, vol. 22, no. 4, 2012.

[151] Q. Yu, H. Tang, K. C. Tan, and H. Li, "Precise-spike-driven synaptic plasticity: Learning hetero-association of spatiotemporal spike patterns," *PLoS One*, vol. 8, no. 11, p. e78318, 2013.

[152] J. D. Victor and K. P. Purpura, "Nature and precision of temporal coding in visual cortex: a metric-space analysis," *J. Neurophysiol.*, vol. 76, no. 2, pp. 1310–1326, 1996.

[153] K. S. Cole and H. J. Curtis, "Electrical impedance of nerve during activity," *Nature*, vol. 142, pp. 209–210, 1938.

[154] C. Mead and M. Ismail, *Analog VLSI implementation of neural systems*. Springer Science & Business Media, 2012, vol. 80.

[155] M. M. Khan, D. R. Lester, L. A. Plana, A. Rast, X. Jin, E. Painkras, and S. B. Furber, "SpiNNaker: mapping neural networks onto a massively-parallel chip multiprocessor," in *IEEE Int. Joint Conf. Neural Netw.* Hong Kong, China: IEEE, Jun. 2008, pp. 2849–2856.

[156] T. Sharp, C. Patterson, and S. Furber, "Distributed configuration of massively-parallel simulation on SpiNNaker neuromorphic hardware," in *IEEE Int. Joint Conf. Neural Netw.* San Jose, CA: IEEE, Aug. 2011, pp. 1099–1105.

[157] K. Minkovich, N. Srinivasa, J. M. Cruz-Albrecht, Y. Cho, and A. Nogin, "Programming time-multiplexed reconfigurable hardware using a scalable neuromorphic compiler," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 23, no. 6, pp. 889–901, 2012.

[158] P. Lichtsteiner, C. Posch, and T. Delbruck, "A 128×128 120 dB 15μs latency asynchronous temporal contrast vision sensor," *IEEE J. Solid-St Circ*, vol. 43, no. 2, pp. 566–576, 2008.

[159] C. Posch, D. Matolin, and R. Wohlgenannt, "A QVGA 143 dB dynamic range frame-free PWM image sensor with lossless pixel-level video compression and time-domain CDS," *IEEE J. Solid-St Circ.*, vol. 46, no. 1, pp. 259–275, 2011.

[160] C. Brandli, R. Berner, M. Yang, S.-C. Liu, and T. Delbruck, "A 240× 180 130 dB 3 μs latency global shutter spatiotemporal vision sensor," *IEEE J. Solid-St Circ.*, vol. 49, no. 10, pp. 2333–2341, 2014.

[161] A. Vanarse, A. Osseiran, and A. Rassau, "A review of current neuromorphic approaches for vision, auditory, and olfactory sensors," *Front. Neurosci.*, vol. 10, no. 115, pp. 1–6, 2016.

[162] S.-C. Liu, A. van Schaik, B. A. Minch, and T. Delbruck, "Asynchronous binaural spatial audition sensor with 2 64 4 channel output," *IEEE Trans. Biomed. Circuits Syst.*, vol. 8, no. 4, pp. 453–464, 2014.

[163] K. T. Ng, F. Boussaid, and A. Bermak, "A CMOS single-chip gas recognition circuit for metal oxide gas sensor arrays," *IEEE Trans. Circ. Syst.*, vol. 58, no. 7, pp. 1569–1580, 2011.

[164] H.-Y. Hsieh and K.-T. Tang, "VLSI implementation of a bio-inspired olfactory spiking neural network," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 23, no. 7, pp. 1065–1073, 2012.

[165] M. Bernabei, K. C. Persaud, S. Pantalei, E. Zampetti, and R. Beccherelli, "Large-scale chemical sensor array testing biological olfaction concepts," *IEEE Sens. J.*, vol. 12, no. 11, pp. 3174–3183, 2012.

[166] R. Douglas, M. Mahowald, and C. Mead, "Neuromorphic analogue VLSI," *Annu. Rev. Neurosci.*, vol. 18, pp. 255–281, 1995.

[167] R. Silver, K. Boahen, S. Grillner, N. Kopell, and K. L. Olsen, "Neurotech for neuroscience: unifying concepts, organizing principles, and emerging tools," *J. Neurosci.*, vol. 27, no. 44, pp. 11 807–11 819, 2007.

[168] E. Fragnière, "A 100-channel analog CMOS auditory filter bank for speech recognition," in *IEEE Int. Digest of Technical Papers. Solid-St. Circ. Conf.*, vol. 1. San Francisco, CA: IEEE, Feb. 2005, pp. 140–589.

[169] R. Kwok, "Neuroprosthetics: Once more, with feeling." *Nature*, vol. 497, no. 7448, pp. 176–178, 2013.

[170] E. M. Izhikevich *et al.*, "Simple model of spiking neurons," *IEEE Trans. Neural Netw.*, vol. 14, no. 6, pp. 1569–1572, 2003.

[171] T. Mouri, H. Kawasaki, K. Yoshikawa, J. Takai, and S. Ito, "Anthropomorphic robot hand: Gifu hand III," in *Proc. Int. Conf. Control Autom. Syst.*, Seoul, Korea, Oct. 2002, pp. 1288–1293.

[172] F. Bergner, P. Mittendorfer, E. Dean-Leon, and G. Cheng, "Event-based signaling for reducing required data rates and processing power in a large-scale artificial robotic skin," in *IEEE/RSJ Int. Conf. Intell. Robot. Syst.* Hamburg, Germany: IEEE, Sep. 2015, pp. 2124–2129.

[173] TekScan. (2016) I-Scan system. [Online]. Available: https://www.tekscan.com/products-solutions/systems/i-scan-system

[174] W. W. Lee, S. L. Kukreja, and N. V. Thakor, "A kilohertz kilotaxel tactile sensor array for investigating spatiotemporal features in neuromorphic touch," in *IEEE Biomed. Circuits Syst. Conf.* Atlanta, GA: IEEE, Oct 2015, pp. 1–4.

[175] R. S. Dahiya, P. Mittendorfer, M. Valle, G. Cheng, and V. J. Lumelsky, "Directions toward effective utilization of tactile skin: A review," *IEEE Sens. J.*, vol. 13, no. 11, pp. 4121–4138, 2013.

[176] H. Akolkar, C. Meyer, Z. Clady, O. Marre, C. Bartolozzi, S. Panzeri, and R. Benosman, "What can neuromorphic event-driven precise timing add to spike-based pattern recognition?" *Neural Comput.*, vol. 27, no. 3, pp. 561–593, 2015.

[177] W. Fukui, F. Kobayashi, F. Kojima, H. Nakamoto, N. Imamura, T. Maeda, and H. Shirasawa, "High-Speed Tactile Sensing for Array-Type Tactile Sensor and Object Manipulation Based on Tactile Information," *J. Robot.*, vol. 2011, pp. 1–9, 2011.

[178] T. D'Alessio, "Measurement errors in the scanning of piezoresistive sensors arrays," *Sensor Actuat. A-Phys.*, vol. 72, no. 1, pp. 71–76, 1999.

[179] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 85, pp. 2579–2605, 2008.

[180] M. El-Desouki, M. Jamal Deen, Q. Fang, L. Liu, F. Tse, and D. Armstrong, "CMOS image sensors for high speed applications," *Sensors*, vol. 9, no. 1, pp. 430–444, 2009.

[181] B. C.-K. Tee, A. Chortos, A. Berndt, A. K. Nguyen, A. Tom, A. McGuire, Z. C. Lin, K. Tien, W.-G. Bae, H. Wang *et al.*, "A skin-inspired organic digital mechanoreceptor," *Science*, vol. 350, no. 6258, pp. 313–316, 2015.

[182] R. S. Johansson, "Dynamic use of tactile afferent signals in control of dexterous manipulation," in *Sensorimotor Control of Movement and Posture.* Springer, 2002, pp. 397–410.

[183] V. J. Lumelsky, *Sensing, intelligence, motion: how robots and humans move in an unstructured world.* John Wiley & Sons, 2005.

[184] M. H. Lee, "Tactile sensing: new directions, new challenges," *Int. J. Robot. Res.*, vol. 19, no. 7, pp. 636–643, 2000.

[185] A. Schmitz, P. Maiolino, M. Maggiali, L. Natale, G. Cannata, and G. Metta, "Methods and technologies for the implementation of large-scale robot tactile sensors," *IEEE Trans. Robot.*, vol. 27, no. 3, pp. 389–400, 2011.

[186] F. Buchthal and A. Rosenfalck, "Evoked action potentials and conduction velocity in human sensory nerves," *Brain Res.*, vol. 3, no. 1, pp. v–122, 1966.

[187] A. Russell, R. Armiger, R. Vogelstein, S. Bensmaia, and R. Etienne-Cummings, "Real-time implementation of biofidelic SA1 model for tactile feedback," in *Conf. Proc. IEEE Eng. Med. Biol. Soc.* Minneapolis, MN: IEEE, Sep. 2009, pp. 185–188.

[188] W. W. Lee, J. Cabibihan, and N. Thakor, "Bio-mimetic strategies for tactile sensing," in *Proc. IEEE Sens.* Baltimore, MD: IEEE, Nov. 2013, pp. 1–4.

[189] W. W. Lee, H. Yu, and N. V. Thakor, "Gait event detection through neuromorphic spike sequence learning," in *Proc. IEEE RAS EMBS Int. Conf. Biomed. Robot. Biomechatron.* Sao Paulo, Brazil: IEEE, Aug. 2014, pp. 899–904.

[190] K. O. Johnson and J. R. Phillips, "Tactile spatial resolution. i. two-point discrimination, gap detection, grating resolution, and letter recognition," *J. Neurophysiol.*, vol. 46, no. 6, pp. 1177–1192, 1981.

[191] N. Elango and A. Faudzi, "A review article: investigations on soft materials for soft robot manipulations," *Int. J. Adv. Manuf. Tech.*, vol. 80, no. 5-8, pp. 1027–1037, 2015.

[192] MATLAB, *version 7.10.0 (R2010a).* Natick, Massachusetts: The MathWorks Inc., 2012a.

[193] J.-J. Cabibihan, S. S. Chauhan, and S. Suresh, "Effects of the artificial skin's thickness on the subsurface pressure profiles of flat, curved, and braille surfaces," *IEEE Sens. J.*, vol. 14, no. 7, pp. 2118–2128, 2014.

[194] L. Paredes-Madrid, L. Emmi, E. Garcia, and P. G. De Santos, "Detailed study of amplitude nonlinearity in piezoresistive force sensors," *Sensors*, vol. 11, no. 9, pp. 8836–8854, 2011.

[195] S. Bensmaıa, "A transduction model of the meissner corpuscle," *Math. Biosci.*, vol. 176, no. 2, pp. 203–217, 2002.

[196] S. S. Kim, A. P. Sripati, and S. J. Bensmaia, "Predicting the timing of spikes evoked by tactile stimulation of the hand," *J. Neurophysiol.*, vol. 104, no. 3, pp. 1484–1496, 2010.

[197] C. E. Shannon, "A mathematical theory of communication," *Mobile Comput. Comm. Rev*, vol. 5, no. 1, pp. 3–55, 2001.

[198] C. Magri, K. Whittingstall, V. Singh, N. K. Logothetis, and S. Panzeri, "A toolbox for the fast information analysis of multiple-site LFP, EEG and spike train recordings," *BMC Neurosci.*, vol. 10, no. 1, p. 1, 2009.

[199] S. Panzeri and A. Treves, "Analytical estimates of limited sampling biases in different information measures," *Network-Comp. Neural*, vol. 7, no. 1, pp. 87–107, 1996.

[200] R. S. Johansson and K. J. Cole, "Sensory-motor coordination during grasping and manipulative actions," *Curr. Opin. Neurobiol.*, vol. 2, no. 6, pp. 815–823, 1992.

[201] J. D. Victor and K. P. Purpura, "Metric-space analysis of spike trains: theory, algorithms and application," *Network-Comp. Neural*, vol. 8, no. 2, pp. 127–164, 1997.

[202] M. C. van Rossum, "A novel spike distance," *Neural Comput.*, vol. 13, no. 4, pp. 751–763, 2001.

[203] T. Delbrück, B. Linares-Barranco, E. Culurciello, and C. Posch, "Activity-driven, event-based vision sensors," in *IEEE Int. Symp. Circuits Syst.* Paris, France: IEEE, May 2010, pp. 2426–2429.

[204] V. Chan, S.-C. Liu, and A. Van Schaik, "AER EAR: A matched silicon cochlea pair with address event representation interface," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 54, no. 1, pp. 48–59, 2007.

[205] W. Gerstner and W. M. Kistler, *Spiking neuron models: Single neurons, populations, plasticity*. Cambridge university press, 2002.

[206] M. N. Shadlen and W. T. Newsome, "Noise, neural codes and cortical organization," *Curr. Opin. Neurobiol.*, vol. 4, no. 4, pp. 569–579, 1994.

[207] S. Thorpe, A. Delorme, and R. Van Rullen, "Spike-based strategies for rapid processing," *Neural networks*, vol. 14, no. 6, pp. 715–725, 2001.

[208] W. Maass and H. Markram, "On the computational power of circuits of spiking neurons," *J. Comput. Syst. Sci.*, vol. 69, no. 4, pp. 593–616, 2004.

[209] J. C. Tapson, G. K. Cohen, S. Afshar, K. M. Stiefel, Y. Buskila, R. M. Wang, T. J. Hamilton, and A. van Schaik, "Synthesis of neural networks for spatio-temporal spike pattern recognition and processing," *Front. Neurosci.*, vol. 7, no. 153, pp. 1—21, 2013.

[210] N. Brunel, V. Hakim, P. Isope, J.-P. Nadal, and B. Barbour, "Optimal information storage and the distribution of synaptic weights: perceptron versus purkinje cell," *Neuron*, vol. 43, no. 5, pp. 745–757, 2004.

[211] B. Barbour, N. Brunel, V. Hakim, and J.-P. Nadal, "What can we learn from synaptic weight distributions?" *Trends Neurosci.*, vol. 30, no. 12, pp. 622–629, 2007.

[212] S. P. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[213] J. F. Sturm, "Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones," *Optim. Method Softw.*, vol. 11, no. 1-4, pp. 625–653, 1999.

[214] Mosek, ApS. (2010, Sep.) The MOSEK optimization software. [Online]. Available: http://www.mosek.com

[215] G. La Camera, A. Rauch, H.-R. Lüscher, W. Senn, and S. Fusi, "Minimal models of adapted neuronal response to in vivo–like input currents," *Neural Comput.*, vol. 16, no. 10, pp. 2101–2124, 2004.

[216] M. Grant and S. Boyd, "CVX: MATLAB software for disciplined convex programming, version 2.1," http://cvxr.com/cvx, Mar. 2014.

[217] K.-C. Toh, M. J. Todd, and R. H. Tütüncü, "SDPT3 – a MATLAB software package for semidefinite programming, version 1.3," *Optim. Method Softw.*, vol. 11, no. 1-4, pp. 545–581, 1999.

[218] J. Lofberg, "YALMIP: A toolbox for modeling and optimization in MATLAB," in *IEEE Int. Symp. on Comput. Aided Control Syst. Design.* Taipei, Taiwan: IEEE, Sep. 2004, pp. 284–289.

[219] Gurobi Optimization, Inc. (2014) Gurobi optimizer reference manual. [Online]. Available: http://www.gurobi.com

[220] J. Perry, J. R. Davids *et al.*, "Gait analysis: normal and pathological function," *J. Pediatr. Orthoped.*, vol. 12, no. 6, p. 815, 1992.

[221] S. Ghoussayni, P. Catalfamo, D. Moser, and D. Ewins, "Experience in the use of a single gyroscope as a sensor for FES foot drop correction systems," in *Proc. 9th Ann. Conf. Int. FES Soc.*, Seoul, Korea, Sep. 2004, pp. 398–400.

[222] E. M. Izhikevich, "Which model to use for cortical spiking neurons?" *IEEE Trans. Neural Networks*, vol. 15, no. 5, pp. 1063–1070, 2004.

[223] A. Hreljac and R. N. Marshall, "Algorithms to determine event timing during normal walking using kinematic data," *J. Biomech.*, vol. 33, no. 6, pp. 783–786, 2000.

[224] S. De Rossi, S. Crea, M. Donati, P. Rebersek, D. Novak, N. Vitiello, T. Lenzi, J. Podobnik, M. Munih, and M. Carrozza, "Gait segmentation using bipedal foot pressure patterns," in *Proc. IEEE RAS EMBS Int. Conf. Biomed. Robot. Biomechatron.* Rome, Italy: IEEE, Jun. 2012, pp. 361–366.

[225] X. Meng, H. Yu, and M. P. Tham, "Gait phase detection in able-bodied subjects and dementia patients," in *35th IEEE Annu. Int. Conf. Eng. Med. Biol. Soc.* Osaka, Japan: IEEE, Jul. 2013, pp. 4907–4910.

[226] N. Karmarkar, "A new polynomial-time algorithm for linear programming," in *Proc. of the 16th Ann. ACM Symp. on Theory of Computing.* Washington, DC: ACM, Apr. 1984, pp. 302–311.

[227] G. B. Dantzig, A. Orden, P. Wolfe *et al.*, "The generalized simplex method for minimizing a linear form under linear inequality restraints," *Pac. J. Math.*, vol. 5, no. 2, pp. 183–195, 1955.

[228] W. Maass, T. Natschläger, and H. Markram, "Real-time computing without stable states: A new framework for neural computation based on perturbations," *Neural Comput.*, vol. 14, no. 11, pp. 2531–2560, 2002.

[229] W. Maass, P. Joshi, and E. D. Sontag, "Computational aspects of feedback in neural circuits," *PLoS Comp. Biol.*, vol. 3, no. 1, p. e165, 2007.

[230] J. Dethier, P. Nuyujukian, S. I. Ryu, K. V. Shenoy, and K. Boahen, "Design and validation of a real-time spiking-neural-network decoder for brain–machine interfaces," *J. Neural. Eng.*, vol. 10, no. 3, p. 036008, 2013.

[231] R. D. Airan, K. R. Thompson, L. E. Fenno, H. Bernstein, and K. Deisseroth, "Temporally precise in vivo control of intracellular signalling," *Nature*, vol. 458, no. 7241, pp. 1025–1029, 2009.

[232] J. Mattingley and S. Boyd, "Automatic code generation for real-time convex optimization," in *Convex optimization in signal process. and commun.*, D. Palomar and Y. Eldar, Eds. Cambridge University Press, 2009, pp. 1–41.

[233] R. M. Wang, T. J. Hamilton, J. C. Tapson, and A. van Schaik, "A mixed-signal implementation of a polychronous spiking neural network with delay adaptation," *Front. Neurosci.*, vol. 8, no. 51, pp. 118–133, 2014.

[234] S. Hussain, A. Basu, R. M. Wang, and T. J. Hamilton, "Delay learning architectures for memory and classification," *Neurocomputing*, vol. 138, pp. 14–26, 2014.

[235] E. Oja, "Neural networks, principal components, and subspaces," *Int. J. Neural Syst.*, vol. 1, no. 01, pp. 61–68, 1989.

[236] W. W. Lee, C. H. Yeow, H. Ren, S. L. Kukreja, and N. V. Thakor, "FPGA implementation of a FA-1 mechanoreceptor model for efficient representation of tactile features," in *Proc. IEEE RAS EMBS Int. Conf. Biomed. Robot. Biomechatron.* Singapore: IEEE, Jun. 2016, pp. 243–246.

Appendix 1

LARGE SENSOR ARRAY REFERENCE MANUAL

Lee Wang Wei

REVISION 1.0

(Internal distribution only)

NATIONAL UNIVERSITY OF SINGAPORE

2016

Contents

1 Overview

The large sensor array (LSA) is a prototype tactile sensor array for evaluating temporally precise tactile sensing over large surfaces. It has 4096 sensing elements and is tested to achieve 5208 Hz sampling rate. It is driven by a Xilinx Spartan-6 FPGA located on the Digilent Atlys development board.

This manual provides a detailed description of the system, from the fabrication of the sensor to the PC interface. It assumes that the reader has basic experience with the following software tools (version used during development indicated in parenthesis):

- Altium Designer (2013)
- Xilinx ISE, EDK and SDK (v14.7)
- Visual Studio (2010)
- Matlab (2012b)

In addition, knowledge on the following programming languages will be required:

- VHDL - for the logic implementation on FPGA
- C++ - for code on the soft microprocessor
- C# - for PC interface code
- Matlab - for data analysis

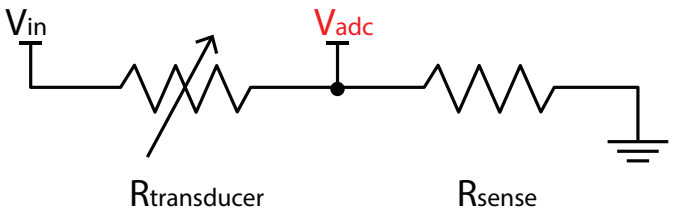


Figure 1.1: Potential divider circuit for single transducer readout. The lower the $R_{transducer}$ value, the larger the V_{adc} reading.

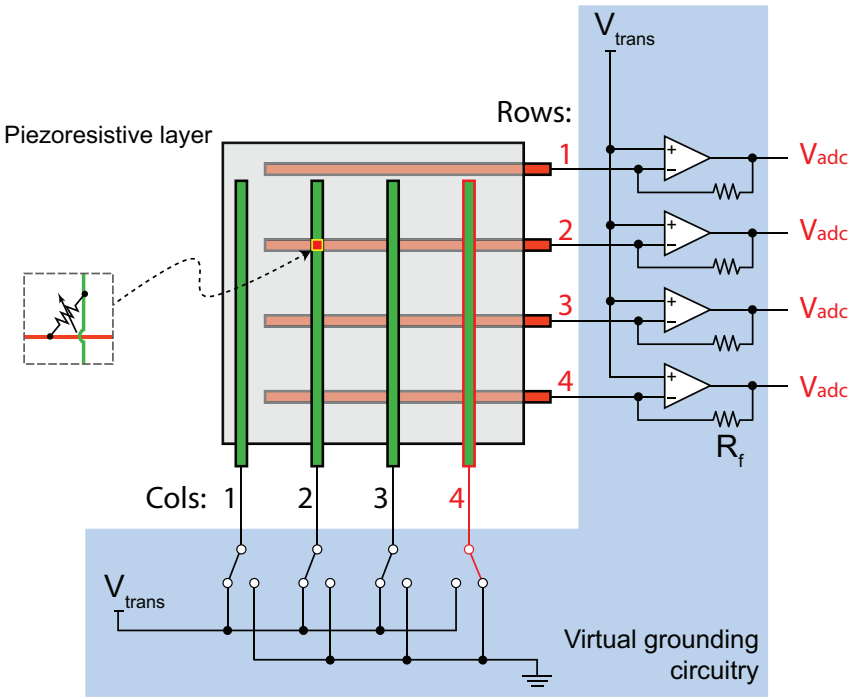


Figure 1.2: Virtual grounding for multiplexed readout of an array of piezoresistive transducers. In this configuration, intersections (1,4), (2,4), (3,4), (4,4) can be read out simultaneously.

1.1 Theory of operation

1.1.1 Multiplexing

The sensor operates by piezoresistive transduction. A single piezoresistive transducer can be interfaced using the potential divider circuit (Fig. 1.1).

As pressure is applied, the resistance across the transducer element decreases. This is the basic working principle of the LSA. However, the circuit described in Fig. 1.1 requires one ADC pin per transducer, which is not scalable for large arrays. Instead, a *virtual grounding* approach was used to multiplex the readout from an array of transducer elements (Fig. 1.2).

Using the layout described in Fig. 1.2, all row and conductor traces are maintained at the same voltage (V_{trans}), except for the column 4 which is grounded (highlighted in red). This ensures that current from rows 1, 2, 3 and 4 can only flow to column 4 via the intersections (1,4), (2,4), (3,4) and (4,4). These intersections are the transduction elements to be read. Current will not flow through any other intersection since there is no potential difference across them, thus eliminating cross-talk.

1.1.2 Analog to Digital Conversion

An additional advantage to using the layout described in Fig. 1.2 is that all rows can be read simultaneously. The LSA implements a single-slope ADC to achieve row-parallel readout while reducing the components and physical connections needed. Fig. 1.3 describes the single-slope ADC concept.

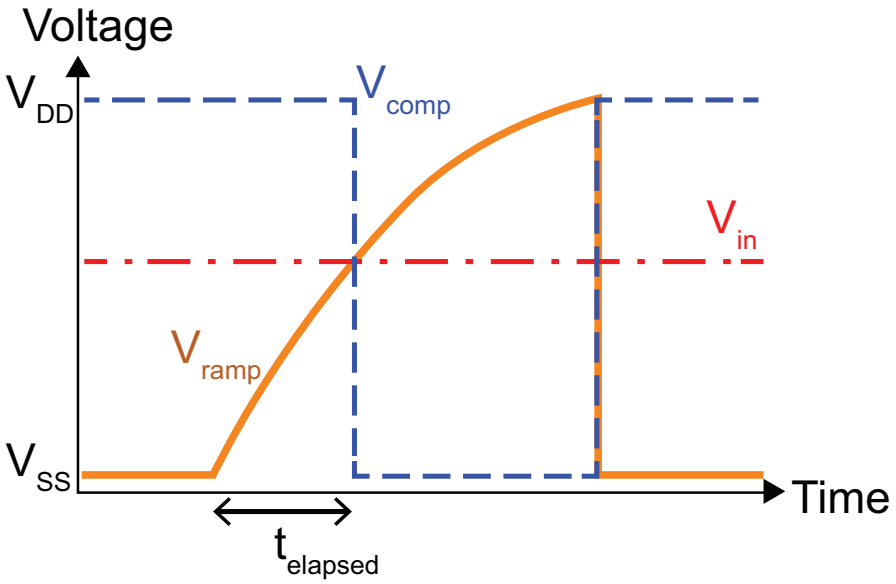


Figure 1.3: Single slope ADC concept. The controller first starts the voltage ramp V_{ramp} while simultaneously starting a timer. The timer stops when the digital signal V_{comp} transitions from high to low. The time $t_{elapsed}$ will be proportional to the level of analogue signal V_{in}.

2 Sensor and circuitry

2.1 Steps to making the sensor

1. Print the traces

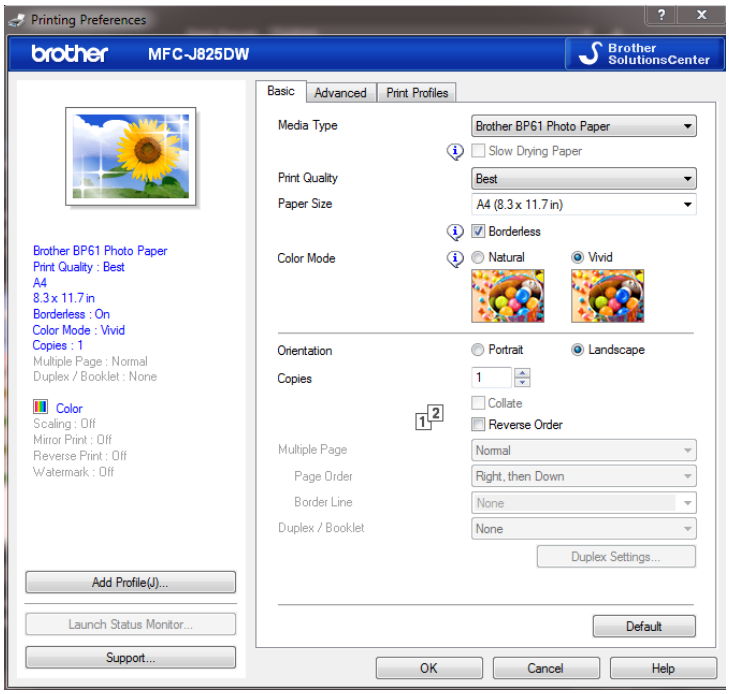


Figure 2.1: Printer settings for printing conductive circuits

Print the following designs on suitable media (either photo paper or PET film) from AgIC. Two copies of `PrintedSensor_V4.pdf` and one copy of `PrintedSensor_V4_adaptor.pdf` is needed. The printer model is Brother MFC-J825DW with settings described in Fig. 2.1.

2. Attach trace to adapter

Cut the adapter printout in half to create the two adapter traces. Next, connect the trace circuit and adapter circuits using 3M 9703 conductive double sided tape.

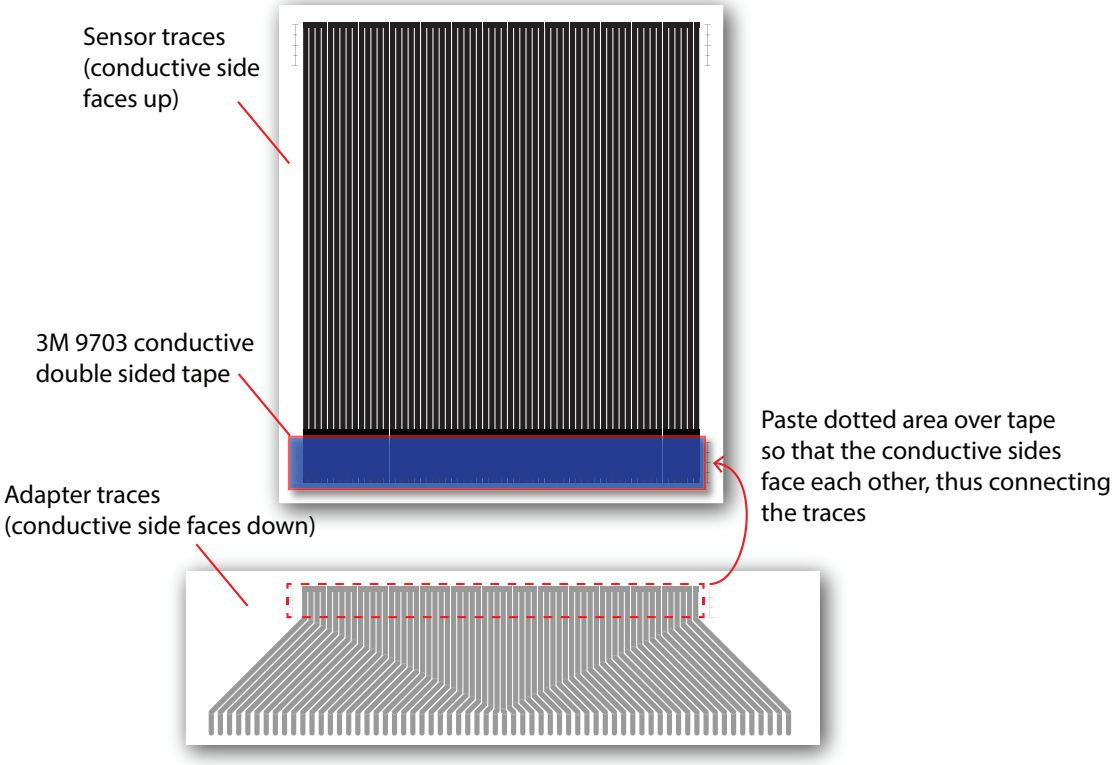


Figure 2.2: Connecting the traces to the adapter.

3. Sandwich the transducer layer between traces

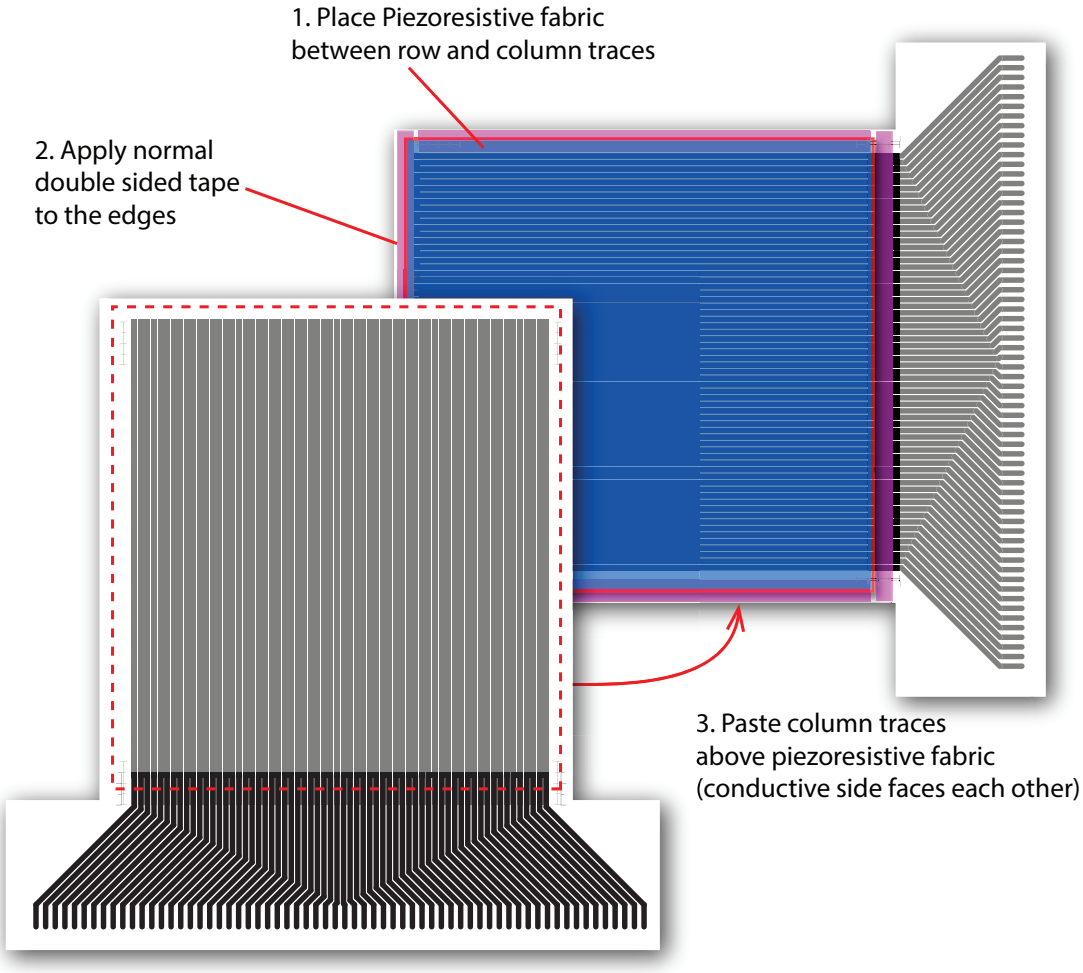


Figure 2.3: Sandwich the transducer between layer of conductive traces to form the sensor matrix.

4. Attach ends to PCB board

The sensor is now complete.

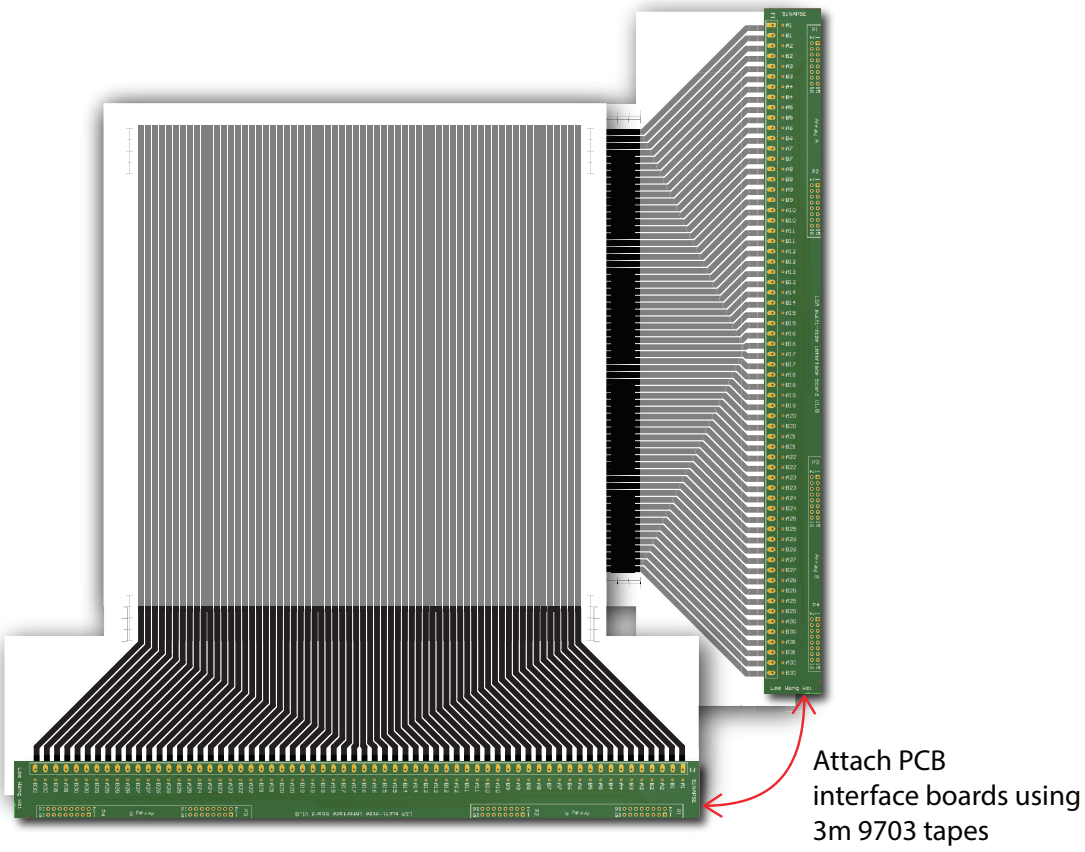


Figure 2.4: Using the 3M 9703 double sided tape again, attach the adapter traces to the interface PCBs.

2.1.1 Points to take note

The connections formed using the 3M 9703 tape may not hold well over time. It is necessary to press down on these joints from time to time. In addition, the adapter traces were made in order to accommodate the oversized interface PCBs. A new PCB design should eliminate the need for these traces.

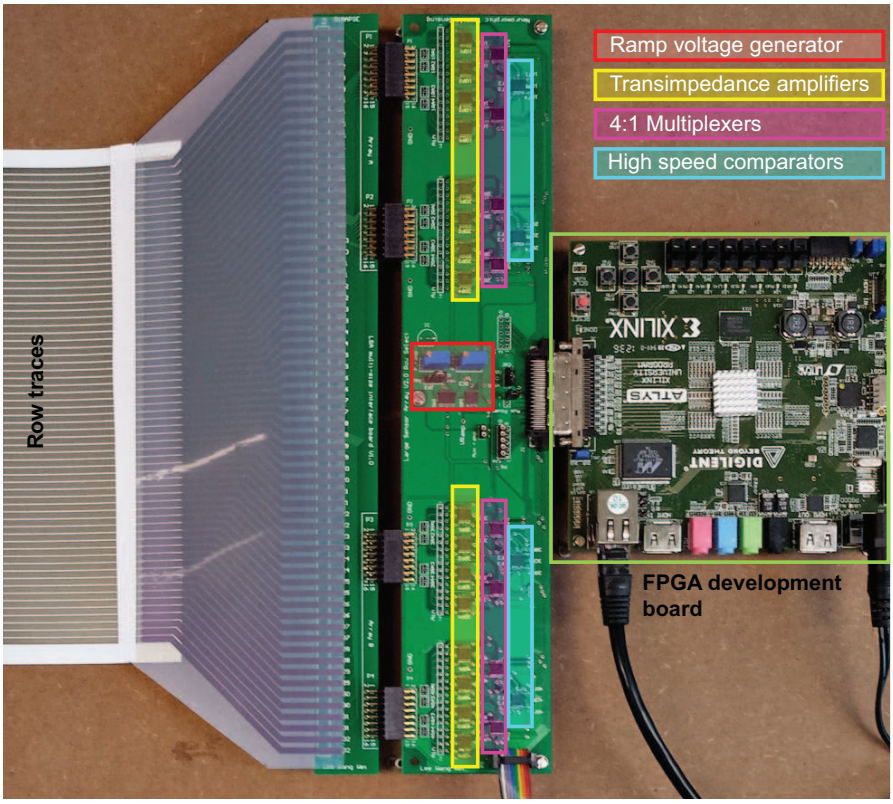
The transducer material (LR-SLPA) is no longer in production. It is a very thin fabric and has excellent dynamic range. However, due to its thinness, there is a tendency for traces to be shorted. This can be minimized by ensuring that the fabric is not stretched during step 3. However, if a trace does short, it will require the entire fabric to be replaced. An alternative material would be LTT-SLPA. However, much larger gains are required as it is less sensitive.

2.2 Circuit

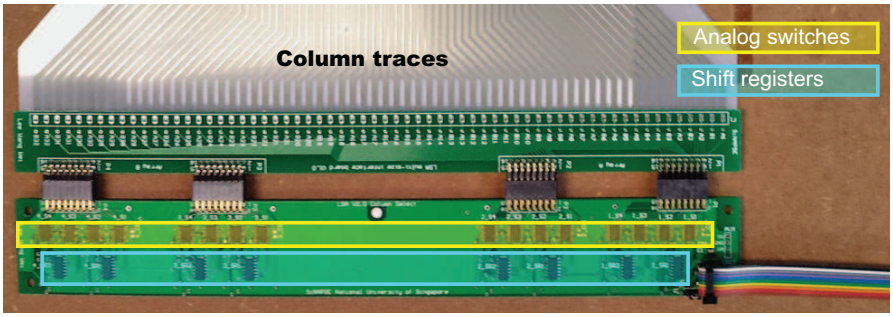
2.2.1 Bill of materials

Table 2.1: Bill of materials for row select board

Item	Designator	Quantity
Capacitor 0.1uF	0C1, 0C2, 0C3, 0C4, 1C1, 1C2, 1C3, 1C4, 1C5, 1C6, 1C7, 2C1, 2C2, 2C3, 2C4, 2C5, 2C6, 2C7, 3C1, 3C2, 3C3, 3C4, 3C5, 3C6, 3C7, 4C1, 4C2, 4C3, 4C4, 4C5, 4C6, 4C7	32
Capacitor 10uF	0C01, 0C5	2
Capacitor 100uF	0C05	1
Capacitor 1nF	0C6	1
Capacitor 15pF	1C001, 1C002, 1C003, 1C004, 1C005, 1C006, 1C007, 1C008, 1C009, 1C010, 1C011, 1C012, 1C013, 1C014, 1C015, 1C016, 2C001, 2C002, 2C003, 2C004, 2C005, 2C006, 2C007, 2C008, 2C009, 2C010, 2C011, 2C012, 2C013, 2C014, 2C015, 2C016, 3C001, 3C002, 3C003, 3C004, 3C005, 3C006, 3C007, 3C008, 3C009, 3C010, 3C011, 3C012, 3C013, 3C014, 3C015, 3C016, 4C001, 4C002, 4C003, 4C004, 4C005, 4C006, 4C007, 4C008, 4C009, 4C010, 4C011, 4C012, 4C013, 4C014, 4C015, 4C016	64
Capacitor 0.01uF	1C08, 1C09, 1C10, 1C11, 1C12, 2C08, 2C09, 2C10, 2C11, 2C12, 3C08, 3C09, 3C10, 3C11, 3C12, 4C08, 4C09, 4C10, 4C11, 4C12	20
Quad Comparator MCP6564	1CP1, 2CP1, 3CP1, 4CP1	4
4:1 Multiplexer ADG709	1M1, 1M2, 2M1, 2M2, 3M1, 3M2, 4M1, 4M2	8
Quad op-amp OPA4354	1OP1, 1OP2, 1OP3, 1OP4, 2OP1, 2OP2, 2OP3, 2OP4, 3OP1, 3OP2, 3OP3, 3OP4, 4OP1, 4OP2, 4OP3, 4OP4, OP1	17
Quad Resistor Array	1RA1, 1RA2, 2RA1, 2RA2, 3RA1, 3RA2, 4RA1, 4RA2	8
Res3	1RF1, 1RF2, 1RF3, 1RF4, 1RI1, 1RI2, 1RI3, 1RI4, 2RF1, 2RF2, 2RF3, 2RF4, 2RI1, 2RI2, 2RI3, 2RI4, 3RF1, 3RF2, 3RF3, 3RF4, 3RI1, 3RI2, 3RI3, 3RI4, 4RF1, 4RF2, 4RF3, 4RF4, 4RI1, 4RI2, 4RI3, 4RI4, R1	33
Header 2 pin	Aux Power, Aux ramp	2
Capacitor 22pF	CR	1
Current source LM334Z	IC	1
Connector VHDCI	J1	1
Header 3 pin	JP2, JP3	2
Header 8X2	P1, P2, P3, P4	4
Header 5X2	P5, P6, P7	3
Potentiometer	P_ramp, P_ref	2
Quad switch ADG734	SW1	1



(a) ADC layout and components.



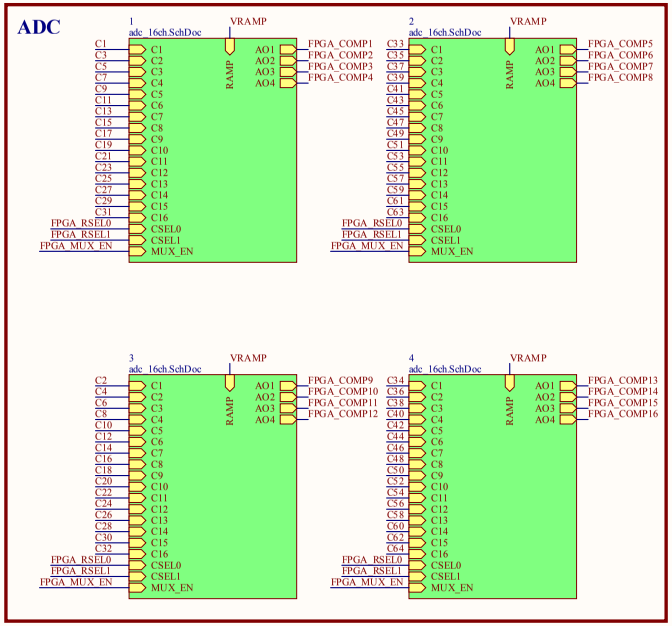
(b) Multiplexer board layout and components.

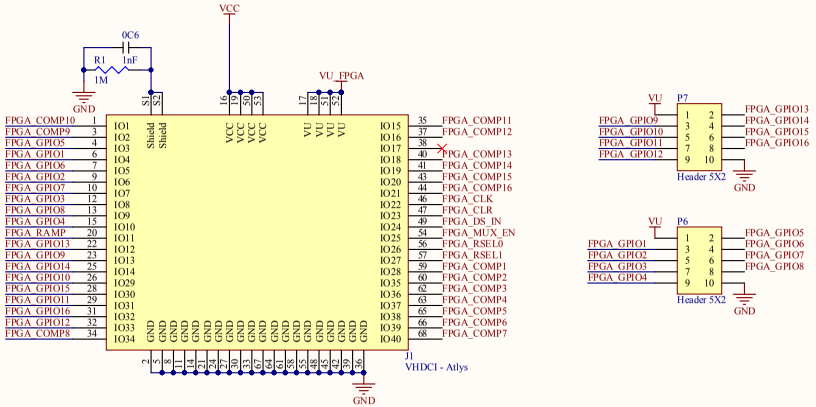
Figure 2.5: Printed circuit board layout.

2.2.2 Schematic

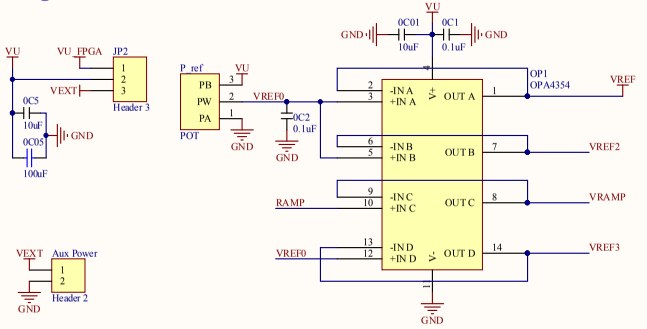
Schematic for Kilotaxel Kilohertz sensor array - Analog board

ADC

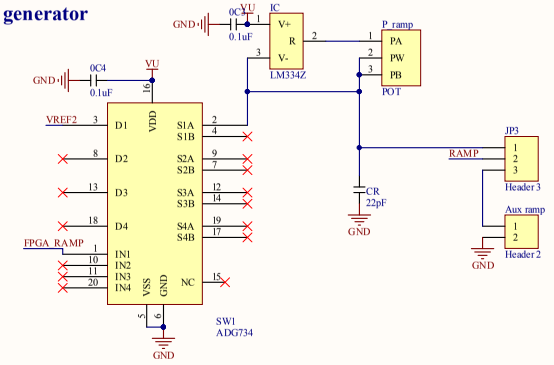




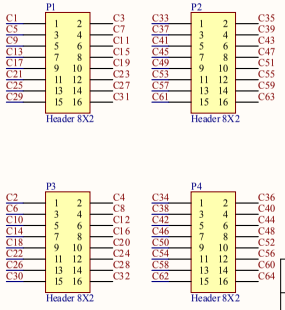
Power regulation



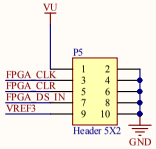
Ramp generator



Connectors to sensor rows



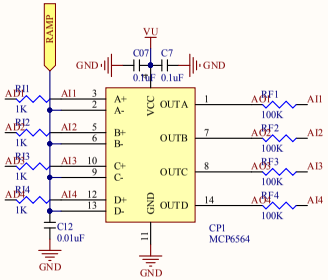
Connectors to MUX board



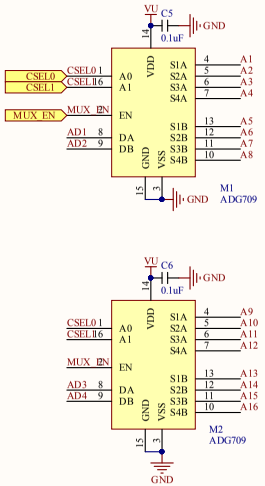
Title		
Size: A3	Number	Revision
Date: 09-Aug-16		Sheet of
File: C:\Users\...\VHDCI.SchDoc		Drawn By:

Analog to digital conversion

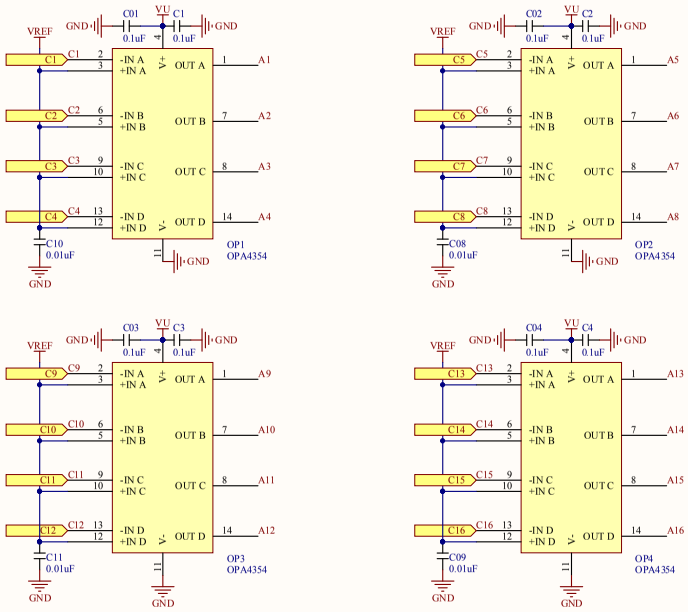
Voltage comparators



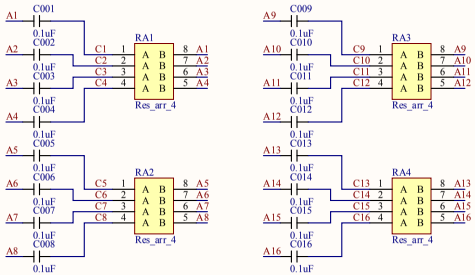
4:1 multiplexers



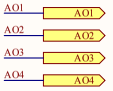
Transimpedance amplifiers



Amplifier gain



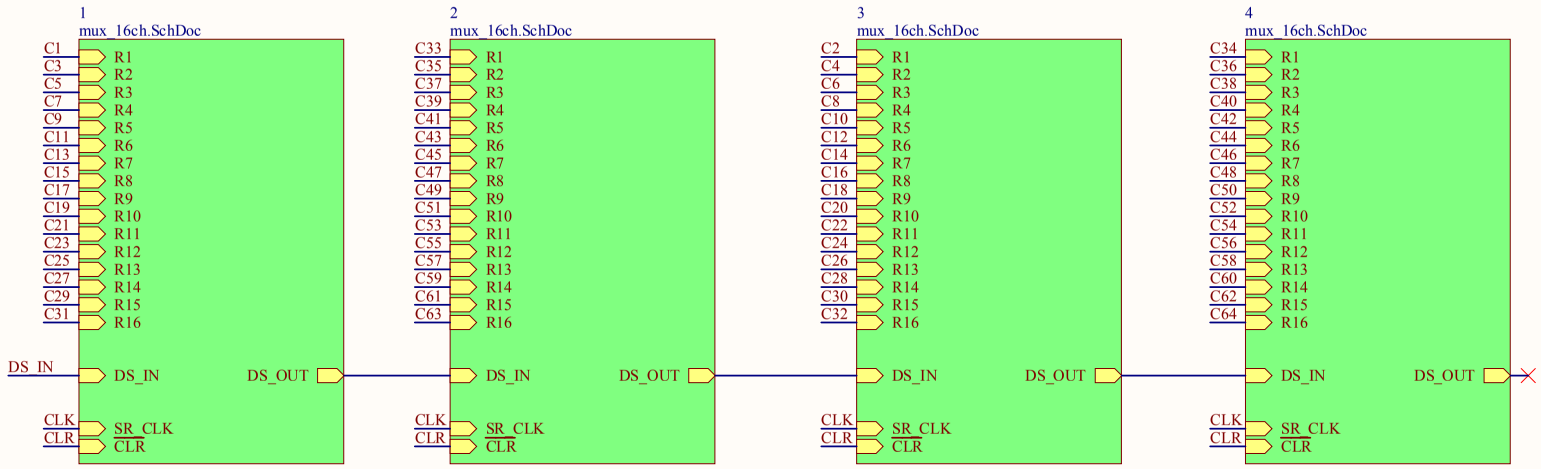
Comparator output



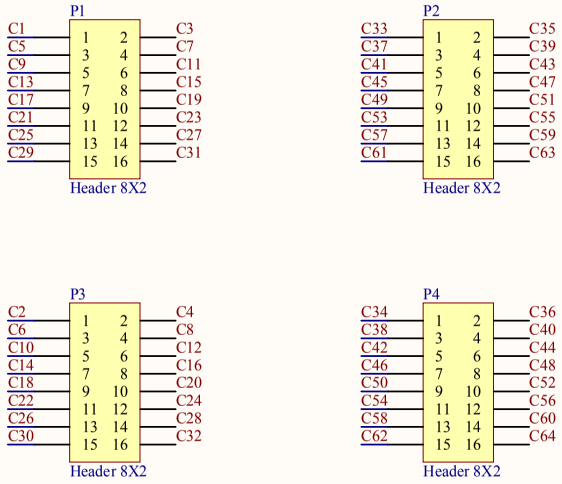
Title		
Size A3	Number	Revision
Date: 09-Aug-16		Sheet of
File: C:\Users\..\adc 16ch.SchDoc		Drawn By:

Column multiplexing board

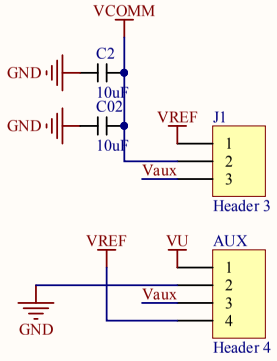
Column multiplexers



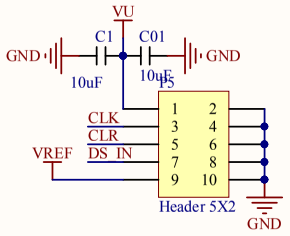
Connection to sensor columns



Voltage select



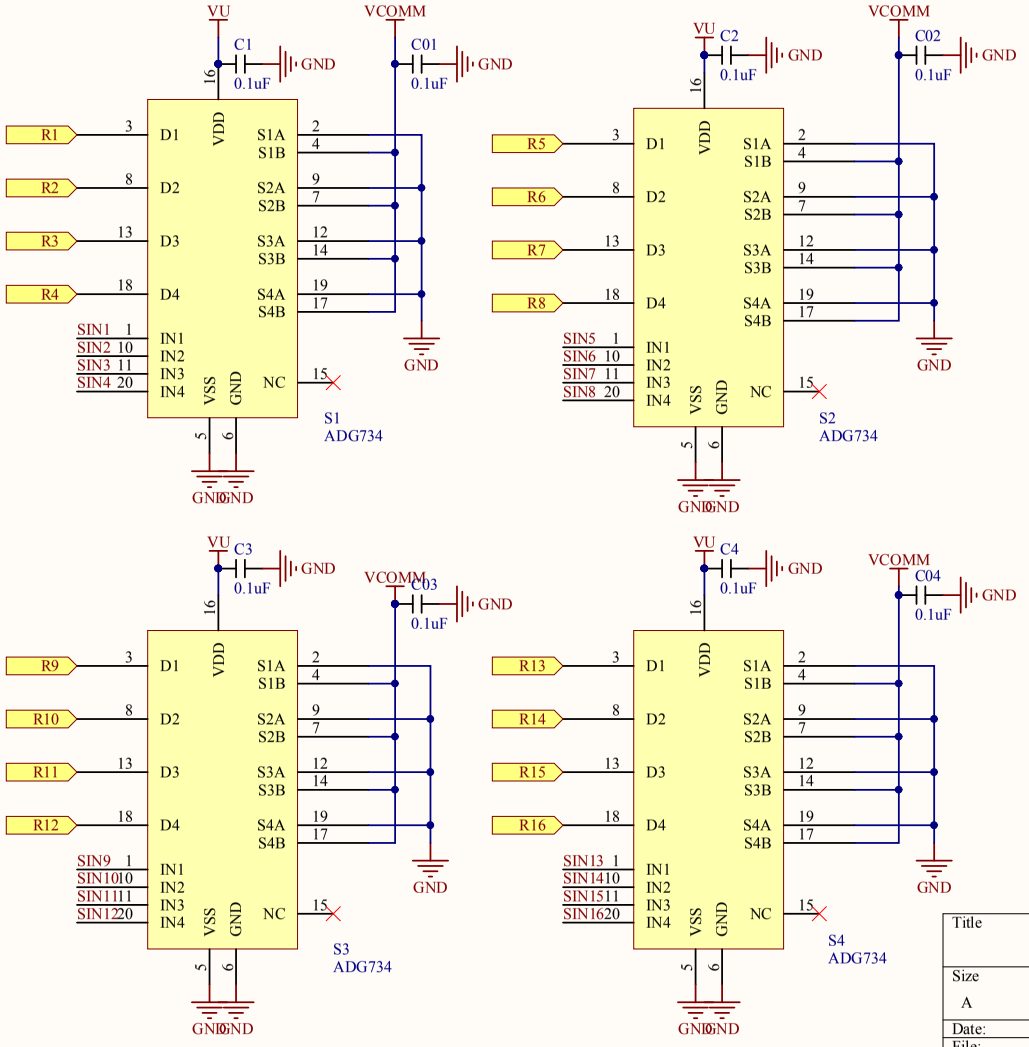
Connection to Analog board



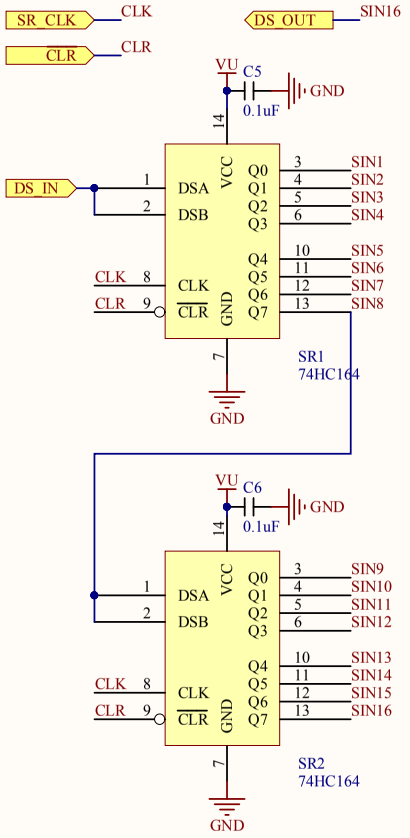
Title		
Size: A	Number	Revision
Date: 09-Aug-16		Sheet of
File: C:\Users\..\mux_board.SchDoc		Drawn By:

Column multiplexer

Analog switches



Shift register control



Title		
Size: A	Number	Revision
Date: 09-Aug-16		Sheet of
File: C:\Users\..\mux_16ch.SchDoc		Drawn By:

Table 2.2: Bill of materials for column multiplex board.

Item	Designator	Quantity
Capacitor 0.1uF	1_C1, 1_C2, 1_C3, 1_C4, 1_C5, 1_C6, 2_C1, 2_C2, 2_C3, 2_C4, 2_C5, 2_C6, 3_C1, 3_C2, 3_C3, 3_C4, 3_C5, 3_C6, 4_C1, 4_C2, 4_C3, 4_C4, 4_C5, 4_C6,	24
Capacitor 10uF	C1, C2	2
Quad switch ADG734	1_S1, 1_S2, 1_S3, 1_S4, 2_S1, 2_S2, 2_S3, 2_S4, 3_S1, 3_S2, 3_S3, 3_S4, 4_S1, 4_S2, 4_S3, 4_S4	16
Shift register 74HC164	1_SR1, 1_SR2, 2_SR1, 2_SR2, 3_SR1, 3_SR2, 4_SR1, 4_SR2	8
Header 4	AUX	1
Header 3	J1	1
Header 8X2	P1, P2, P3, P4	4
Header 5X2	P5	1

The source files for the PCB layout are included in ADC.zip and MUX.zip for the ADC board and multiplexer board respectively.

2.2.3 Points to take note

Due to the limited pins from the Atlys board, only 16 ADC channels were implemented, thus requiring the use of 4:1 multiplexers (four scans per column). If a development board with more pins were available, the sampling rate can be increased at least 4 times.

Special care must be taken when routing the V_{ramp} trace. It is advisable that the next version include a buffer for each MCP6564 chip to decouple the effects of comparator kickback from affecting the other comparator channels.

The circuit was designed to use the LM334 current source and capacitor CR to generate the ramp signal. This was non-optimal however, as the LM334 could not sustain a constant current as the potential at the output terminal approaches supply voltage. In the end, an RC circuit was used to generate the ramp signal instead. It can be fed through the "Aux ramp" port, or modified from the LM334 socket.

The PCB was designed with several contingency plans in place. Besides "Aux ramp", it is also possible to use an alternate (external) power supply through VAux. Both 5V (VU) and 3.3V are available from the Atlys board too. It is advisable to use the 3.3V supply as it supports higher current. VU supports a maximum of 1 A only. Selection between VU and VAux is done using jumper JP2. Finally, the unused pins from the VHDCI connector were also brought out on headers P6 and P7.

There are also pads to accommodate additional capacitors. These are pre-fixed with the digit 0 **after the letter**. (e.g. 0C1 is essential but 0C01 is optional). Capacitors with labels 00x (where $x \in \mathbb{R}$) are for smoothing of the analogue signal (V_{in} in Fig. 1.3). A value of 15pF is recommended. Larger values result in clean signals but reduces signal slew rate, which will require a longer settling period before the can signal be read.

3 FPGA logic

3.1 Architecture

An FPGA was used to drive the circuit, exploiting its ability to perform parallel computation with timing precision. Fig. 3.1 illustrates the functional block diagram of the system.

The Microblaze soft microprocessor forms the interface between the sampling logic and PC. A serial UART communication channel forms a bi-directional bridge between the PC and Microblaze, through which commands and status information are exchanged. Meanwhile, events generated by the sampling logic are communicated in the form of UDP packets through a separate Ethernet connection, in order to achieve higher throughput.

3.2 Sampling Logic

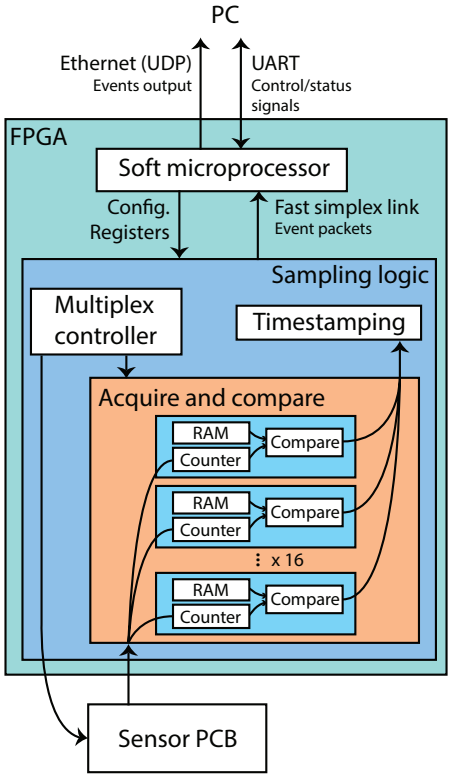


Figure 3.1: Functional block diagram

3.2.1 Inputs

LSA_top.vhd is a facade to the individual logical components. It provides 8 32-bit configuration registers which are set through the microblaze GPIO ports. The timing parameters are determined by registers `row_change_period`, `row_read_duration`, `col_change_period` and `col_read_duration`. Their relations are described in Fig. 3.3.

As this is a prototype system, there are no hardware error checks implemented. This means that the system may enter undetermined states if invalid values are used (for instance when `col_change_period` > `row_change_period`). Therefore, it is important for the checks to be implemented in software to ensure that the values are valid at all times.

The other parameters are responsible for how the sensor events are generated, and are described below.

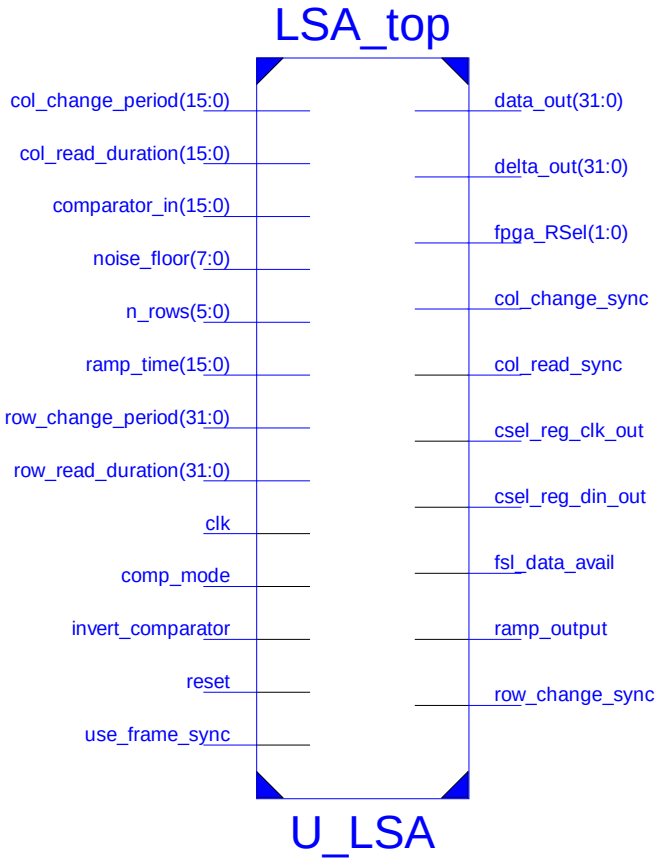


Figure 3.2: LSA_top function block

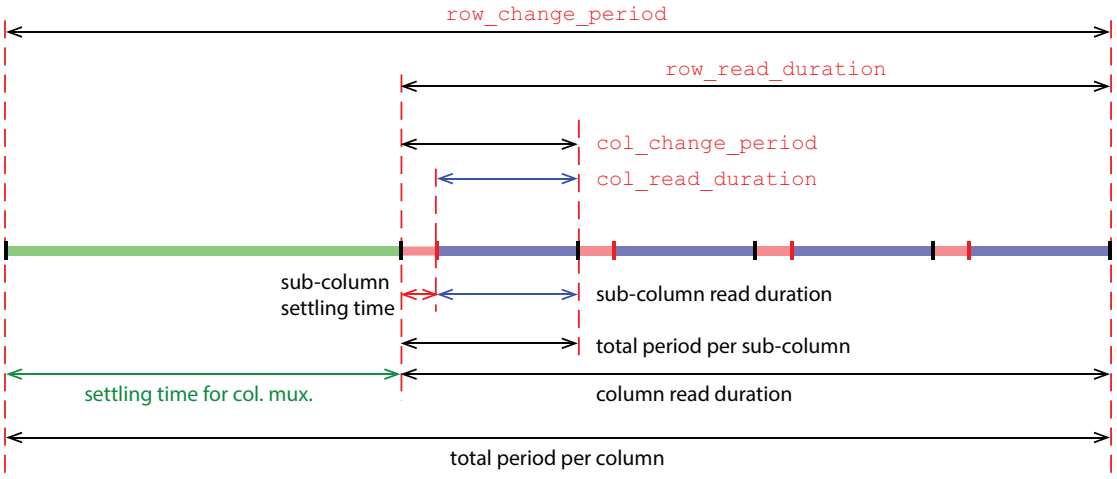


Figure 3.3: Configurable timing parameters and their naming convention. Names in `red` are variable names. The parameters have a unit of 10 ns.

`noise_floor`

Determines the minimum difference in $t_{elapsed}$ before an event is generated. Value is in units of 3 ns.

`n_rows`

Sets the number of columns to multiplex, in case a sensor with < 64 columns is used. Default is `0x3F`.

`ramp_time`

The duration of which ramp voltage is rising. It should be <`col_read_duration` to allow the sampling logic to be reset before a subsequent read. However, it is necessary to ensure that `ramp_time` is long enough for the ramp voltage to reach its peak. Value is in units of 3 ns.

`comp_mode`

Unused for now. It is useful for mechanoreceptor modelling in the future.

`invert_comparator`

When asserted, a falling edge indicates the end of a sample (default = 0).

`use_frame_sync`

When asserted, all events generated during the same frame will get the same time stamp (default = 1).

3.2.2 Outputs

`ramp_output`

This signal is used to charge the capacitor during the ramp period.

`fpga_RSel`

A 2-bit port connected to multiplexer address pins, mapping the 64 rows to 16 ADC channels.

`csel_reg_clk_out`

Clock signal for shift registers in column multiplexing board.

`csel_reg_din_out`

Input to shift registers in column multiplexing board.

`delta_out`

This 32-bit output data channel carries event packets for any pressure changes detected. Only the address, polarity and timestamp information are provided in each packet.

`data_out`

This 32-bit output data channel carries event packets for pressure measurement. The address, polarity and intensity information are included in each packet.

`fsl_data_avail`

As per the fast simplex link (FSL) protocol, this bit signals the microblaze that data is available on the `delta_out` and `data_out` ports.

`col_change_sync`, `col_read_sync` **and** `row_read_sync`

These are debug signals that provide reference when probing the system with an oscilloscope.

3.3 Sub-components

Within `LSA_top.vhd` there are several sub-components. Here we provide a general overview of their function. For detailed operation the reader is advised to read the source code.

3.3.1 MUX_ctrl.vhd

This component generates all the necessary trigger pulses that synchronizes the sampling procedure. It provides the control signal for the multiplexers and shift registers on the ADC and MUX PCB boards respectively. It also provides the ramp signal for charging/discharging the RC ramp generation circuit on the PCB.

3.3.2 CLK_mgr

A sub-component of the `MUX_ctrl` that is responsible for clock division.

3.3.3 TIME_stamper.vhd

This component is responsible for tagging each event with a time-stamp and placing the data on the output channels. Time stamps are in 100 μs precision.

3.3.4 COMP_array.vhd

This component is essentially a facade that coordinates the 16 timers used for ADC. It contains a pre-synthesized clock multipler `fClock.xco` which provides a 300 Mhz clock for higher timing precision. It also waits for all timers to finish sampling before resetting them.

3.3.5 COMP_col.vhd

This component implements the logic for starting and stopping the sampling timer for each ADC channel. It also performs the comparison between previous and current readings to determine if a change has occurred, and if so, the polarity of the change. This is a critical component of the system. In the future, it can be modified to simulate mechanoreceptor responses.

3.3.6 RAMP_counter.vhd

This is a sub-component of `COMP_col` and implements the ADC timer. Unlike all other components, it is driven by a 300 Mhz clock. Therefore, care must be taken to account for the stricter timing constraints within this component.

4 PC interface

4.1 LSA control tab

The PC interface was written in C# using Visual Studio 2010. Fig. 4.1 shows the main interface when the software is launched. Steps for operation are as follows:

1. Select the corresponding Ethernet and serial ports to open.
2. Click on “Connect” and “Open” to establish communication channels.
3. Using the presets drop-down menu, select “5.2 kHz”. This will populate the corresponding fields with the right settings and also communicate the new parameters to the LSA.
4. Click on the “Reset intensity” button to zero the intensity display window.

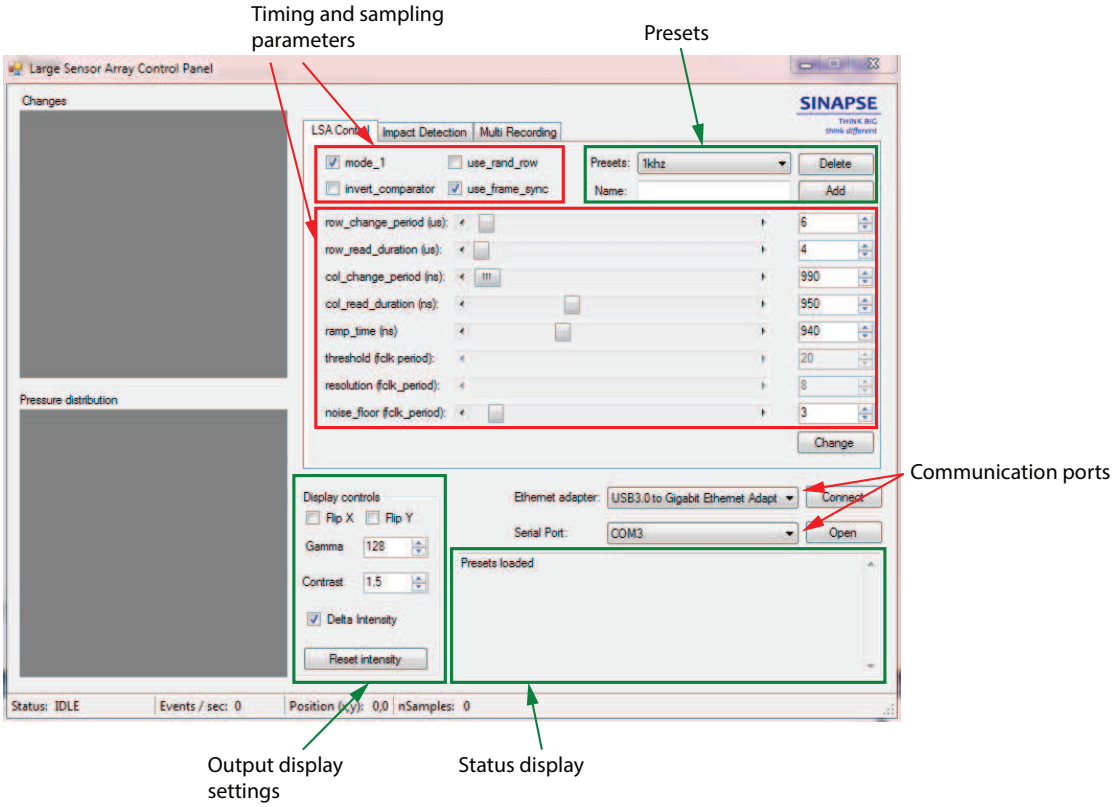


Figure 4.1: PC control panel for LSA.

4.1.1 Timing and sampling parameters

The timing parameters can be changed by either dragging the horizontal scrollbar or writing the value directly in the box to the right. The values between these two controls are synchronized. Upon a change of values, the software will send the corresponding command to the LSA to change the corresponding registers. The status of this change can be observed in the status display.

The software implements a rudimentary check on the validity of the sampling parameters. However, should there be any errors in the updating of values, the user can click on the "Change" button to re-send all values to the LSA.

4.1.2 Presets

Due to the many parameters to define, a preset feature was implemented to save and restore all the settings at once. When the software is first launched, it looks for the presets file "`presets.xml`" which should reside in the same directory as the executable.

The settings in the file will be pre-loaded if they exist. The presets can be applied by selecting from the drop-down menu.

To save the current settings as a new preset, enter a new name in the "Name" field, and click on "Add". The new preset should now be saved and visible in the drop-down menu.

To delete a preset, first select it from the drop-down menu, then click on "delete".

4.1.3 Display options

Depending on the orientation of the sensor, it may be necessary to flip the X and Y axis for a more intuitive display. This is achieved by checking the "Flip X" and "Flip Y" check boxes respectively. The "Gamma" and "Contrast" values control how the intensity values are mapped to the 8 bit pressure map. Finally, the "Delta intensity" check box informs the software that the intensity values from the LSA are differential in nature.

For debugging purposes, the user can click on anywhere within the "Changes" display window to activate a crosshair, pinpointing a specific taxel (Fig. 4.2). The row and column of this taxel will be displayed at the bottom of the screen, corresponding to the trace on the PCB. This is useful for locating a particular taxel that exhibits anomalous behavior. Double clicking the "Changes" window will de-activate this feature.

4.2 Recording LSA data

Raw data from the LSA can be recorded using the interface software.

1. Navigate to the "Multi Recording" tab (Fig. 4.3).
2. Select the directory to save the recording file.
3. Click on the "Record" button to begin recording. Alternatively, press "R" on the keyboard at any time to begin recording.
4. To end the recording, click on "Record" button or press "R" on the keyboard again.

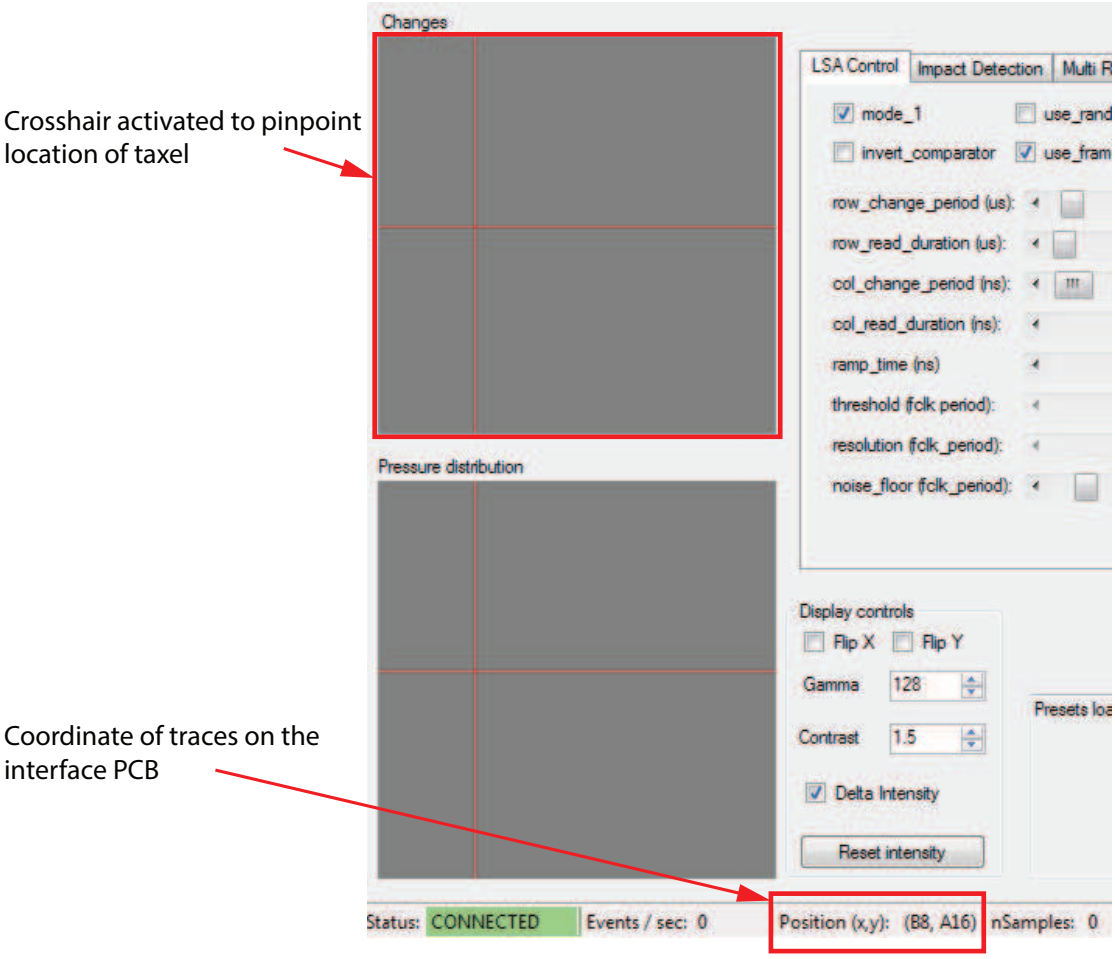


Figure 4.2: Pinpointing coordinates of a particular taxel.

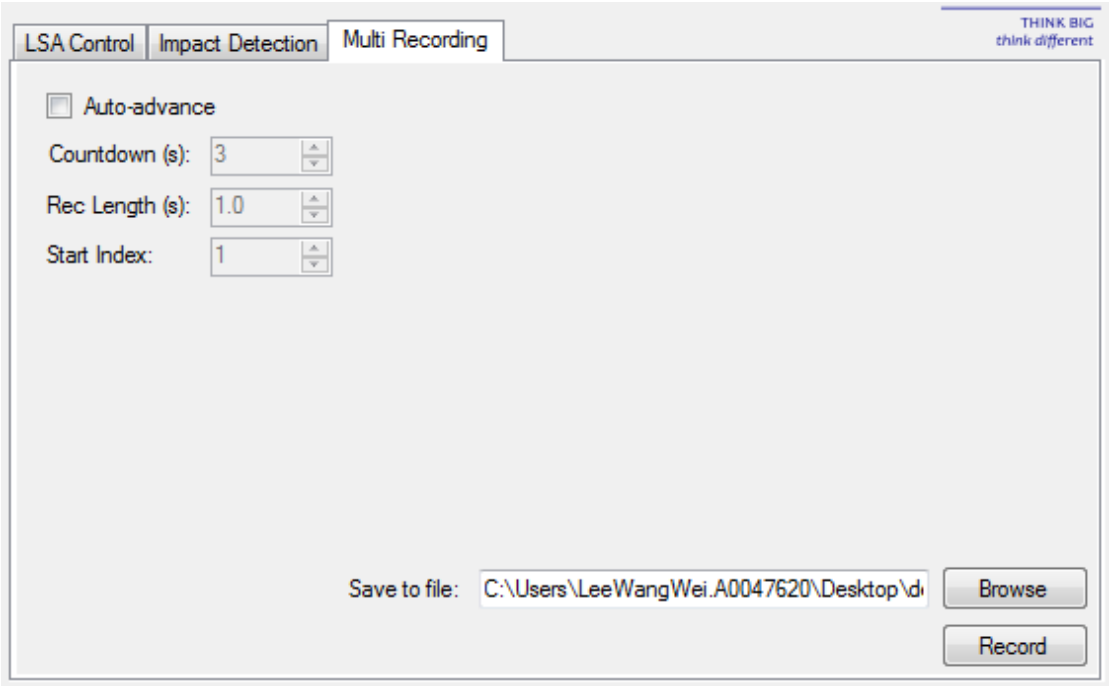


Figure 4.3: Recording interface.

During recording, the display windows on the left do not function. All bytes from the LSA are saved into the "`*.lsa`" file in binary. When a recording is stopped, a corresponding "`*.lss`" file is created. This is an empty file. The purpose of the file is to signal to other software that the "`*.lsa`" file is ready to be read.

4.2.1 Automated recording

When recording multiple trials for dataset creation, the automated recording function can be used to minimize interaction between the user and the software.

1. Click on the "Auto-advance" check box under the "Multi Recording" tab.
2. Set the countdown interval and recording duration in "Countdown" and "Rec Length" fields respectively. Countdown is the time between successive recordings.
3. "Start Index" is a first number that is appended to the filename. This index is incremented with subsequent recordings.
4. Provide a valid directory and filename.
5. Click on the "Record" button. Countdown begins. Low tone audio beeps are played with each second of the countdown.
6. A final, high-toned beep is played to indicate the start of the recording.
7. The process repeats itself automatically until the "Record" button or "R" keyboard button is pressed again to stop the recording session.

Note that the software will overwrite existing files with the same filename without warning.

4.2.2 Time synchronization

At times it is necessary to synchronize the recording of the LSA with other sensors. The LSA hardware is designed to provide a 3.3V rising edge on pin JA4 of the PMOD port of the Atlys board to indicate a start of recording, and a falling edge to indicate the end of a recording. The LSA also generates events that indicate the start and stop of

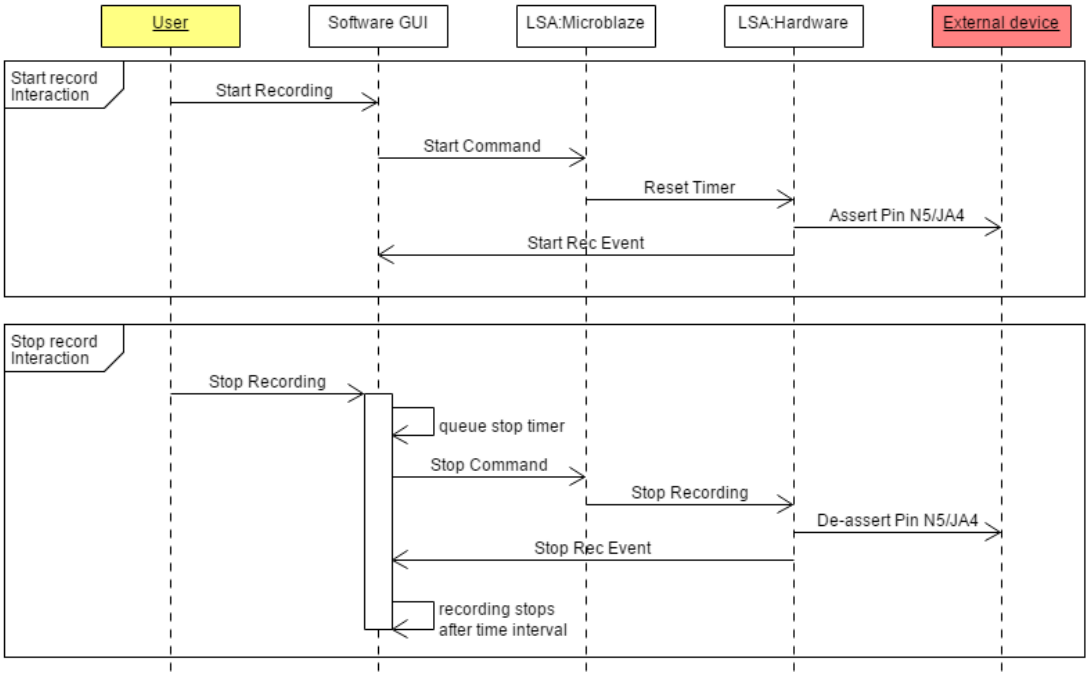


Figure 4.4: Sequence of interaction when a start and stop command is issued by the user.

recording. Fig. 4.4 illustrates the sequence of interactions that take place during the start and stop of a recording.

5 Communication

Communication between different components of the system are illustrated in Fig. 5.1. Binary format is used in all channels except the status updates, which are in ASCII. The following sections describe the formats used for individual communication channels.

5.1 UART command packets

Bits	7	6	5	4	3	2	1	0
$Byte_4$	1	R_2	R_1	R_0	Cmd_37	Cmd_27	Cmd_17	Cmd_07
$Byte_3$	0	Cmd_36	Cmd_35	Cmd_34	Cmd_33	Cmd_32	Cmd_31	Cmd_30
$Byte_2$	0	Cmd_26	Cmd_25	Cmd_24	Cmd_23	Cmd_22	Cmd_21	Cmd_20
$Byte_1$	0	Cmd_16	Cmd_15	Cmd_14	Cmd_13	Cmd_12	Cmd_11	Cmd_10
$Byte_0$	0	Cmd_06	Cmd_05	Cmd_04	Cmd_03	Cmd_02	Cmd_01	Cmd_00

Table 5.1: Command from PC to Microblaze. Each command is 5 bytes. There is a 3-bit address field ($R_2R_1R_0$) to indicate the register to write. The value fields are indicated using the notation $Cmd_{[byte\#]}$[bit#].

5.2 Fast simplex link

Bits 31:24	T_{15}	T_{14}	T_{13}	T_{12}	T_{11}	T_{10}	T_9	T_8
Bits 23:16	T_7	T_6	T_5	T_4	T_3	T_2	T_1	T_0
Bits 15:8	Pol	Z_2	X_5	X_4	X_3	X_2	X_1	X_0
Bits 7:0	Z_1	Z_0	Y_5	Y_4	Y_3	Y_2	Y_1	Y_0

Table 5.2: Format for 32 bit output delta_out from sampling hardware to Microblaze. $T_\#$ = 16 bit timestamp for the change event, $X_\#$ = 6 bit X coordinate, $Y_\#$ = 6 bit Y coordinate, $Z_\#$ indicates type of event and Pol indicates polarity.

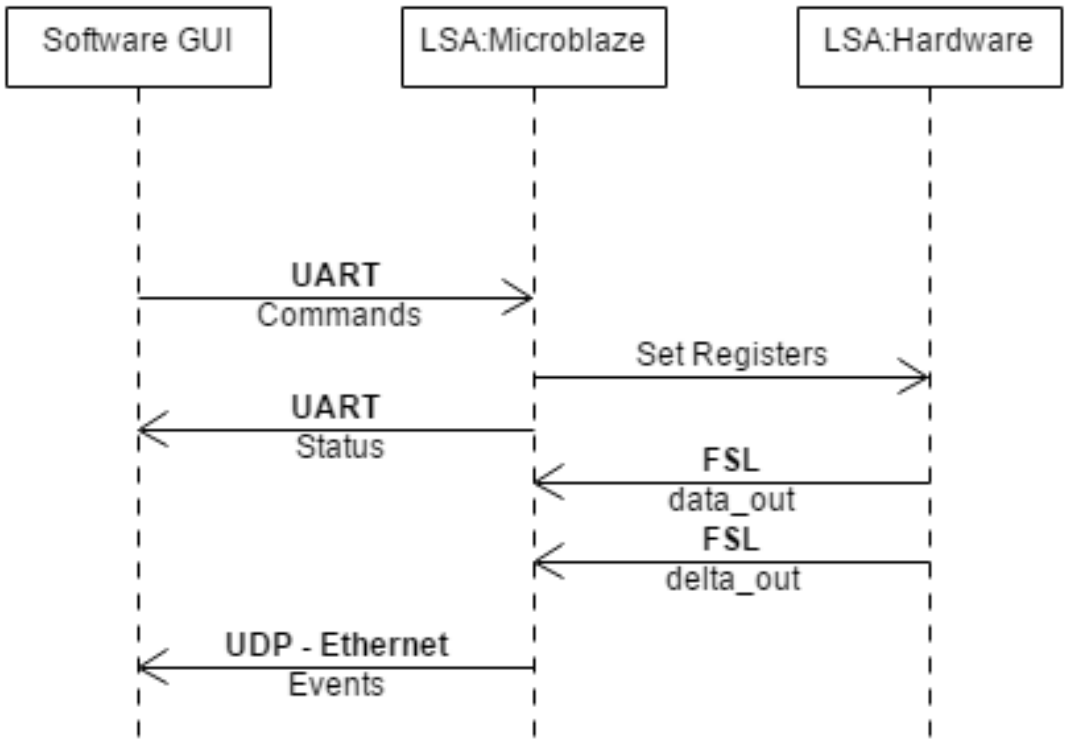


Figure 5.1: Communication channels implemented between different components of the system.

`Bits 31:24`	$\mathtt{I}_{15}$	$\mathtt{I}_{14}$	$\mathtt{I}_{13}$	$\mathtt{I}_{12}$	$\mathtt{I}_{11}$	$\mathtt{I}_{10}$	$\mathtt{I}_{9}$	$\mathtt{I}_{8}$
`Bits 23:16`	$\mathtt{I}_{7}$	$\mathtt{I}_{6}$	$\mathtt{I}_{5}$	$\mathtt{I}_{4}$	$\mathtt{I}_{3}$	$\mathtt{I}_{2}$	$\mathtt{I}_{1}$	$\mathtt{I}_{0}$
`Bits 15:8`	`Pol`	$\mathtt{Z}_{2}$	$\mathtt{X}_{5}$	$\mathtt{X}_{4}$	$\mathtt{X}_{3}$	$\mathtt{X}_{2}$	$\mathtt{X}_{1}$	$\mathtt{X}_{0}$
`Bits 7:0`	$\mathtt{Z}_{1}$	$\mathtt{Z}_{0}$	$\mathtt{Y}_{5}$	$\mathtt{Y}_{4}$	$\mathtt{Y}_{3}$	$\mathtt{Y}_{2}$	$\mathtt{Y}_{1}$	$\mathtt{Y}_{0}$

Table 5.3: Format for 32 bit output `data_out` from sampling hardware to Microblaze. $\mathtt{I}_{\#}$ = 16 bit intensity of the change, $\mathtt{X}_{\#}$ = 6 bit X coordinate, $\mathtt{Y}_{\#}$ = 6 bit Y coordinate, $\mathtt{Z}_{\#}$ indicates type of event and `Pol` indicates polarity.

At the moment, $Z_2Z_1Z_0$ = 1XX indicates that the event is an intensity event and $Z_2Z_1Z_0$ = XX1 indicates that it is a start/stop recording event. As more event types are needed, this format will have to change.

5.3 Event based output

Bits 39:32	1	Pol	Z_2	Z_1	Z_0	U	T_15	T_14
Bits 31:24	0	T_{13}	T_{12}	T_{11}	T_{10}	T_9	T_8	T_7
Bits 23:16	0	T_6	T_5	T_4	T_3	T_2	T_1	T_0
Bits 15:8	0	U	X_5	X_4	X_3	X_2	X_1	X_0
Bits 7:0	0	U	Y_5	Y_4	Y_3	Y_2	Y_1	Y_0

Table 5.4: Format for each event packet Microblaze to PC through Ethernet. $T_{\#}$ = Timestamp or 16 bit intensity of the change (depending on Z_2), $X_{\#}$ = 6 bit X coordinate, $Y_{\#}$ = 6 bit Y coordinate, $Z_{\#}$ indicates type of event and Pol indicates polarity. U = unused for now.

The format for an individual event from Microblaze to PC is described in Table 5.4. Note that there can be multiple such event packets within the payload of a single UDP packet.

5.4 Registers

Register Name	No. of bits	Mapping
use_rand_row	1	REG0[0]
comparator_mode	1	REG0[1]
invert_comparator	1	REG0[2]
use_frame_sync	1	REG0[3]
row_change_period	32	REG1[31:0]
row_read_duration	32	REG2[31:0]
col_change_period	16	REG3[15:0]
col_read_duration	16	REG3[31:16]
ramp_time	16	REG4[15:0]
noise_floor	8	REG5[15:8]
nrows	6	REG5[21:16]
record	1	REG7[16]

Table 5.5: Register map

Appendix 2

Guide to using CONE

Overview

This manual describes the CONE algorithm implemented in the TNNLS paper. It assumes that the user is experienced in using MATLAB, and has basic knowledge in convex optimization and spiking neural networks.

CONE stands for Convex Optimized syNaptic Efficacies. It is a supervised learning algorithm that maps multiple input spike trains to a target output spike train with temporal precision (i.e. the output spike times must match the target spike train). It does so by adjusting the weights assigned to each input synapse (See Fig. 1). Unlike many other similar algorithms, CONE solves for the value of **w** using convex optimization.

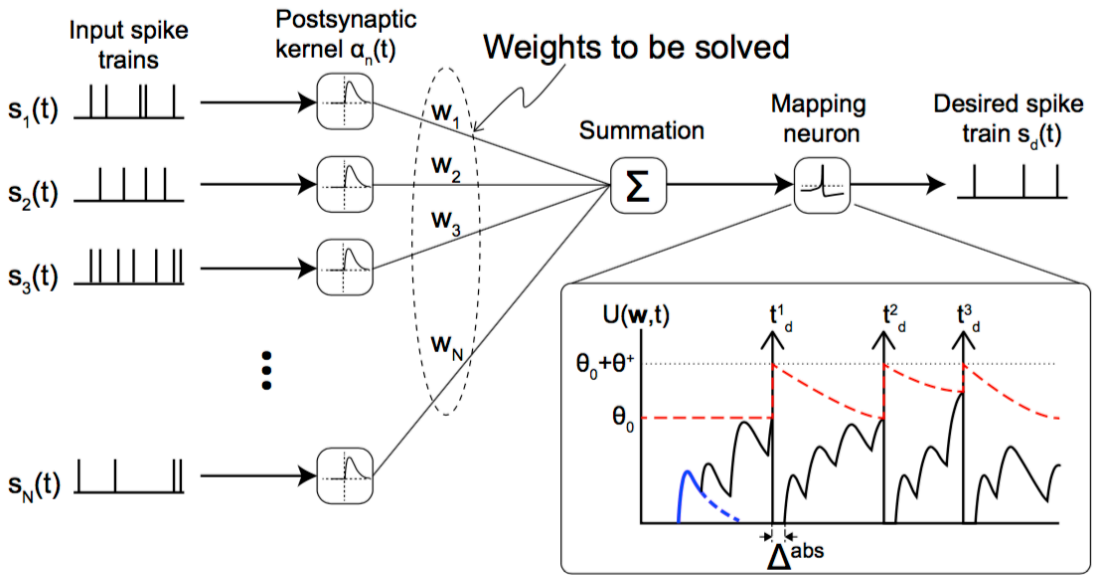


Fig. A: CONE solves for weights such that the output spikes match a desired spike train.

The entire solution process involves the following components:

Step 1 – Formulate problem as convex objective function

Given input and target spike trains $s_n(t)$ and $s_d(t)$ respectively (Eq. 2 and 3 in paper), and knowing the spike kernel (Eq. 5) to be used, we can express the membrane trajectory of the spiking neuron as a function of weight matrix **w** as U(**w**,t) (Eq. 7).

Next, assuming we need millisecond precision, we then compute the expression of U(**w**,t) at all time points of the simulation. I.e. if the last target output spike is at t=100ms, we need to have U(**w**,t) for t = [0:1:100]ms. Usually, I recommend that we compute the expression for t = [0:1:100 + τ]ms where τ is the decay time constant of the kernel (Eq. 5). This is to ensure that no spikes will be emitted after the last target spike.

Next, given that the target spike train is known, we know when the neuron should spike, and when it should not (Eq. 9a, b, c). We also introduce robustness through the use of post-target membrane potentiation (Eq. 10). These constraints are applied at different times of the membrane voltage trajectory to set the boundaries, which is visually described in Fig. B below. The mathematical equivalent is Eq. 11, which is the CONE-X formulation.

CONE-R differs slightly from CONE-X in that the hard constraints are replaced by penalty terms (Eq. 12 and 13 instead of Eq. 9a and 9b).

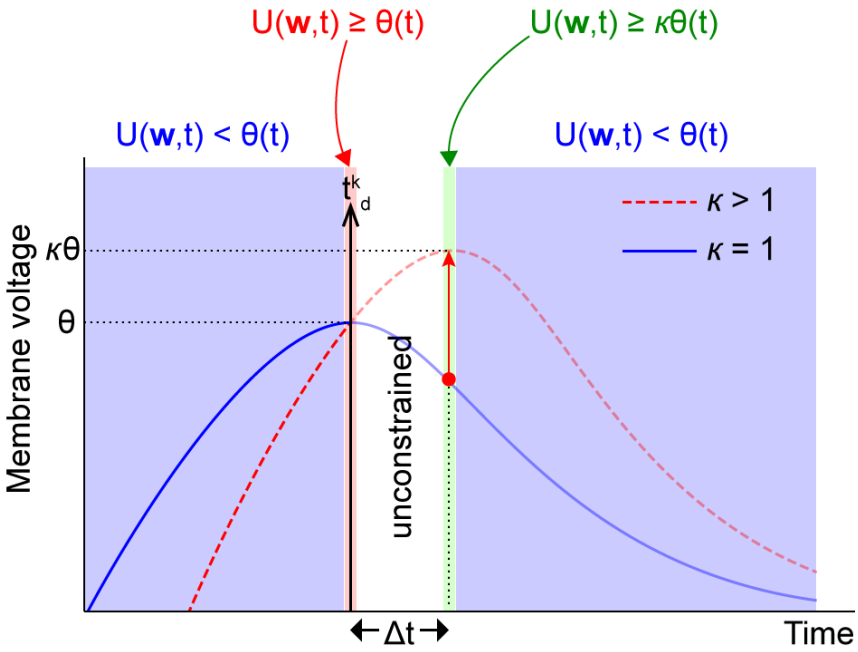


Fig. B: A visual illustration of where the constraints are applied along the membrane voltage trajectory.

Step 2: Translate objective function into a standard format

There are many convex optimization solvers available, from the free SeDuMi and SDPT3 to commercial ones such as those by Gurobi. To generate a problem formulation that is portable among the solvers, we make use of toolboxes such as CVX and YALMIP. Here, CVX is used as an example.

CVX in a plugin for MATLAB which can be installed from http://cvxr.com/cvx/.
An example formulation from step 1 is translated into CVX commands as follows:

$$\text{Minimize:} \quad \hat{\lambda}_1 \frac{\|\mathbf{w}\|_1}{\|\mathbf{w_{max}}\|_1} + \hat{\lambda}_2 \frac{\|\mathbf{w}\|_2}{\|\mathbf{w_{max}}\|_2} + \hat{\lambda}_3 \frac{B(\mathbf{w})}{B_{\max}}$$

Subject to:

$$\begin{aligned} & U(\mathbf{w}, t_d^k) = \theta(t_d^k - t_d^{k-1}) \quad k = 1 \cdots K \\ & U(\mathbf{w}, t') < \theta(t' - t_d^{k-1}) \quad t' \notin \{t_d^k \le t \le t_d^k + \Delta^{\text{abs}}\} \\ & lb \preceq \mathbf{w} \preceq ub. \end{aligned} \tag{11}$$

Let:

$L1 = \frac{\hat{\lambda}_1}{\|w_{max}\|}$

$L2 = \frac{\hat{\lambda}_2}{\|w_{max}\|_2}$

a_mat = coefficients of $\mathbf{w}$ in $U(\boldsymbol{w}, t_d^k)$ where
`size(a_mat) = (numel(K), numel(w))`

b_mat = values of θ at all target spike times t_d^k where
`size(b_mat) = (numel(K),1)`

c_mat = coefficients of $\mathbf{w}$ in $U(\boldsymbol{w}, t')$ where
`size(c_mat) = (numel(total duration – k – k*refractory), numel(w))`

d_mat = values of θ at all non-spiking times t' where
`size(d_mat) = (numel(total duration – k - k*refractory), 1)`

The code will be:

```
cvx_begin % begin CVX command
  variable weights(n_synapse) % specify variable and size
  minimize(L2*norm(weights) + L1*norm(weights,1)) % obj func
subject to % constraints
  a_mat * weights <= b_mat
  c_mat * weights == d_mat
  lb*ones(n_synapse,1) <= weights <= ub*ones(n_synapse,1)
cvx_end % end CVX command
```

Look into the source code (ds_solve_weights_cvx.m) for more examples.

Step 3: Feed into solver

This is the easiest step. Just run the code described in step 2. CVX will call on a predefined solver to solve for the weights. At the end of the operation, CVX will generate the variables 'weights' and 'cvx_status' in the MATLAB workspace. If the solver was successful, cvx_status = 'solved' and the value of the weights will be valid. These weights can be plugged directly into the neuron and it should spike at exactly the target time points.

For details on how to adjust the settings in CVX, look into the manual at http://cvxr.com/cvx/doc/CVX.pdf

Tips:

CVX appears to be faster than YALMIP as many of its functions are precompiled mex files. However, YALMIP has better integration with Gurobi, which is a commercial solver that also checks for problem feasibility. So CONE-X runs better using YALMIP, while most other problems can be formulated using CVX.