

UNIVERSITY OF GENOVA

Polytechnic School



Doctoral Program in Science and Technology for  
Electronic and Telecommunication Engineering (XXXI cycle)

**Electronic systems for  
the restoration of the sense of touch  
in upper limb prosthetics**

Marta Franceschi

\* \* \*

**Supervisor:** Prof. M. Valle

A thesis submitted for the degree of  
*Doctor of Philosophy*

February 2019





# Abstract

In the last few years, research on active prosthetics for upper limbs focused on improving the human functionalities and the control. New methods have been proposed for measuring the user muscle activity and translating it into the prosthesis control commands. Developing the feed-forward interface so that the prosthesis better follows the intention of the user is an important step towards improving the quality of life of people with limb amputation. However, prosthesis users can neither feel if something or someone is touching them over the prosthesis and nor perceive the temperature or roughness of objects. Prosthesis users are helped by looking at an object, but they cannot detect anything otherwise. Their sight gives them most information. Therefore, to foster the prosthesis embodiment and utility, it is necessary to have a prosthetic system that not only responds to the control signals provided by the user, but also transmits back to the user the information about the current state of the prosthesis.

This thesis presents an electronic skin system to close the loop in prostheses towards the restoration of the sense of touch in prosthesis users. The proposed electronic skin system includes an advanced distributed sensing (electronic skin), a system for (i) signal conditioning, (ii) data acquisition, and (iii) data processing, and a stimulation system. The idea is to integrate all these components into a myoelectric prosthesis.

Embedding the electronic system and the sensing materials is a critical issue on the way of development of new prostheses. In particular, processing the data, originated from the electronic skin, into low- or high-level information is the key issue to be addressed by the embedded electronic system. Recently, it has been proved that the Machine Learning is a promising approach in processing tactile sensors information. Many studies have been shown the Machine Learning effectiveness in the classification of input touch modalities.



More specifically, this thesis is focused on the stimulation system, allowing the communication of a mechanical interaction from the electronic skin to prosthesis users, and the dedicated implementation of algorithms for processing tactile data originating from the electronic skin. On system level, the thesis provides design of the experimental setup, experimental protocol, and of algorithms to process tactile data. On architectural level, the thesis proposes a design flow for the implementation of digital circuits for both FPGA and integrated circuits, and techniques for the power management of embedded systems for Machine Learning algorithms.

# Contents

|   |           |
|---|-----------|
| List of Figures   | ix        |
| List of Tables  | x         |
| <b>I Introduction</b>                                       | <b>1</b>  |
| 1 Introduction  | 3         |
| <b>II System and Applications</b>                           | <b>9</b>  |
| <b>2 Electronic Skin System</b>                             | <b>11</b> |
| 2.1 Introduction . . . . .                                  | 11        |
| 2.2 Sense of Touch . . . . .                                | 12        |
| 2.3 Electronic Skin . . . . .                               | 14        |
| 2.4 Electronic Skin as a System . . . . .                   | 18        |
| 2.4.1 Tactile Sensing . . . . .                             | 19        |
| 2.4.2 Interface and Acquisition System . . . . .            | 22        |
| 2.4.3 Tactile Data Processing and Decoding System . . . . . | 23        |
| 2.5 Conclusion . . . . .                                    | 24        |
| <b>3 Tactile Feedback System</b>                            | <b>26</b> |
| 3.1 Introduction . . . . .                                  | 26        |
| 3.2 Motivation . . . . .                                    | 29        |
| 3.3 Transmission Methodologies Overview . . . . .           | 31        |
| 3.3.1 Peripheral Nerve Stimulation . . . . .                | 31        |
| 3.3.2 Targeted Reinnervation . . . . .                      | 33        |
| 3.3.3 Vibrotactile . . . . .                                | 33        |
| 3.3.4 Electrotactile . . . . .                              | 34        |

|            |   |           |
|------------|---|-----------|
| 3.3.5      | Mechanotactile . . . . .                                  | 35        |
| 3.3.6      | Temperature . . . . .                                     | 37        |
| 3.3.7      | Audio . . . . .   | 37        |
| 3.3.8      | Augmented Reality . . . . .                               | 37        |
| 3.4        | Electrocutaneous Stimulation System . . . . .             | 38        |
| 3.5        | Proposed Methods . . . . .                                | 42        |
| 3.5.1      | Experimental Procedure . . . . .                          | 43        |
| 3.5.2      | Data Analysis . . . . .                                   | 47        |
| 3.5.3      | Results and Discussion . . . . .                          | 47        |
| 3.6        | Open issues . . . . .                                     | 54        |
| 3.7        | Conclusion . . . . .                                      | 56        |
| <b>4</b>   | <b>Applications</b>                                       | <b>59</b> |
| 4.1        | Introduction . . . . .                                    | 59        |
| 4.2        | System to Close the Loop in Prosthetic . . . . .          | 61        |
| 4.2.1      | System Architecture . . . . .                             | 61        |
| 4.2.2      | Tactile data processing . . . . .                         | 64        |
| 4.2.3      | Electrotactile spatial coding . . . . .                   | 66        |
| 4.3        | Experimental Results . . . . .                            | 66        |
| 4.3.1      | Experimental Evaluation . . . . .                         | 66        |
| 4.3.2      | Data analysis . . . . .                                   | 69        |
| 4.3.3      | Results . . . . .   | 70        |
| 4.4        | Discussion . . . . .                                      | 72        |
| 4.5        | Conclusion . . . . .                                      | 76        |
| <b>III</b> | <b>Circuits and techniques for the embedded implemen-</b> |           |
|            | <b>tation of tactile data processing and decoding</b>     | <b>78</b> |
| <b>5</b>   | <b>Tactile data processing</b>                            | <b>80</b> |
| 5.1        | Introduction . . . . .                                    | 80        |
| 5.2        | Classification Algorithms in Machine Learning . . . . .   | 82        |
| 5.3        | ML based on Tensorial Kernel Approach . . . . .           | 83        |
| 5.3.1      | Support Vector Machine . . . . .                          | 84        |
| 5.3.2      | Kernel Computation . . . . .                              | 86        |
| 5.3.3      | Singular Value Decomposition . . . . .                    | 87        |
| 5.3.4      | Tensor Unfolding . . . . .                                | 87        |

|           |  |            |
|-----------|--|------------|
| 5.4       | Computational Complexity . . . . .                         | 89         |
| 5.5       | Hardware Implementation of SVD . . . . .                   | 90         |
| 5.6       | Conclusion . . . . .                                       | 93         |
| <b>6</b>  | <b>Approximate Techniques for Hardware Implementation</b>  | <b>96</b>  |
| 6.1       | Introduction . . . . .                                     | 96         |
| 6.2       | Approximate Techniques: An Overview . . . . .              | 98         |
| 6.3       | Inexact Speculative Adder (ISA) . . . . .                  | 101        |
| 6.4       | Inexact Speculative Multiplier (ISM) . . . . .             | 103        |
| 6.5       | ISA in Machine Learning . . . . .                          | 104        |
| 6.5.1     | CORDIC Algorithm . . . . .                                 | 104        |
| 6.5.2     | Hardware Implementation . . . . .                          | 105        |
| 6.5.3     | Experimental Results . . . . .                             | 106        |
| 6.6       | Conclusion . . . . .                                       | 108        |
| <b>7</b>  | <b>Approximate Arithmetics for Hardware Implementation</b> | <b>109</b> |
| 7.1       | Introduction . . . . .                                     | 109        |
| 7.2       | Motivation . . . . .                                       | 111        |
| 7.3       | Tunable Floating-Point . . . . .                           | 112        |
| 7.4       | TFP Units . . . . .  | 114        |
| 7.4.1     | TFP Multiplier . . . . .                                   | 116        |
| 7.4.2     | TFP Adder . . . . .  | 116        |
| 7.4.3     | Hardware Implementation . . . . .                          | 117        |
| 7.5       | TFP Simulation . . . . .                                   | 117        |
| 7.6       | TFP for Deep Learning . . . . .                            | 118        |
| 7.6.1     | Training in TFP . . . . .                                  | 121        |
| 7.6.2     | Error Characterization . . . . .                           | 122        |
| 7.6.3     | Advantages of TFP for Deep Learning . . . . .              | 124        |
| 7.7       | TFP for Embedded Machine Learning . . . . .                | 125        |
| 7.7.1     | SVD Algorithm . . . . .                                    | 125        |
| 7.7.2     | Simulation and Results . . . . .                           | 127        |
| 7.8       | Conclusion . . . . .                                       | 131        |
| <b>IV</b> | <b>Outlook and Conclusion</b>                              | <b>133</b> |
| <b>8</b>  | <b>Outlook and Conclusion</b>                              | <b>135</b> |

|                                       |     |
|---------------------------------------|-----|
| Appendix A Ethical Committee Proposal | 138 |
| Appendix B IntFES v2 Stimulator       | 157 |
| Bibliography                          | 159 |

# List of Figures

|      |   |    |
|------|---|----|
| 1.1  | Towards the restoration of sensory feedback. From left to right: prosthetic hand covered with electronic skin, electronics system in the prosthesis pocket, electrotactile stimulation delivered to the skin of the residual limb. . . . .                            | 5  |
| 2.1  | Section of glabrous skin showing physical location and classification of various mechanoreceptors [7]. . . . .  | 14 |
| 2.2  | A brief chronology of the evolution of e-skin [38]. . . . .   | 17 |
| 2.3  | General physical structure of the e-skin system []. . . . .   | 19 |
| 2.4  | A general functional block diagram of the interface electronics ( [12]).  | 23 |
| 3.1  | Classification of Sensory Feedback Methods. . . . .   | 27 |
| 3.2  | The restoration of sense of touch in prosthesis. . . . .  | 30 |
| 3.3  | Peripheral nerve stimulation (source [77]). . . . .   | 31 |
| 3.4  | Targeted reinnervation (source [87]). . . . .   | 32 |
| 3.5  | Examples of vibrators used in vibrotactile feedback (source [94]). . . .  | 33 |
| 3.6  | Examples of electrotactile electrodes (source [94]). . . . .  | 34 |
| 3.7  | Mechanical pressure feedback device and Silicon bulb mechanical feedback (source [94]). . . . .   | 36 |
| 3.8  | Pressure feedback cuff (source [94]). . . . .   | 36 |
| 3.9  | Experimental setup. The indenter is used for mechanical stimulation, while all other elements are used for electrotactile stimulation experiments.  | 40 |
| 3.10 | Typical stimulation waveform. Notation: A pulse amplitude; w pulse width; d inter-pulse delay; T inter-pulse interval (pulse rate = frequency = $1/T$ ). . . . .  | 41 |
| 3.11 | Reference position for matrix electrode placing is indicated by a black dot: it corresponds to the intersection between two superficial flexors, i.e. the palmaris longus and the flexor carpi ulnaris muscles. Reference pad for that position is number 22. . . . . | 42 |

|      |   |    |
|------|---|----|
| 3.12 | (a) Electrostimulation tests: a sketch of the matrix electrode (4 columns, 6 rows) is placed on the table next to the forearm. (b) Spatial correspondence between the matrix electrode and the geometrical arrangement of the pads in the sketch is preserved. . . . .  | 44 |
| 3.13 | Experiments with mechanical stimulation. A screen is placed between the participants forearm and the sketch of the matrix electrode. (b) Top view. (b) Side view. . . . .   | 45 |
| 3.14 | The results for individual subjects (P1-P8). Reported percentages are associated to identifying the right pad (light blue), missing the pad but addressing the right column (orange), missing the pad and the column (grey), no answer (yellow). . . . .  | 48 |
| 3.15 | The summary results for all subjects. The bars show the success rates (mean standard deviation) in identifying the right pad (a), pointing to the right pad or first neighbors (F.N.) within the same column (b), pointing to the right pad or any pad belonging to the same column (c) and pointing to the right pad or any of its first neighbors, regardless of the column (d). . . . .  | 50 |
| 3.16 | Success rates for the identification of each pad. The scheme reported in (a) illustrates the orientation of the matrix electrode with respect to the forearm. . . . .   | 51 |
| 3.17 | Study of the adaption phenomenon testing electrocutaneous stimulation (synchronously or sequentially) over one single pads or more. . .   | 55 |
| 3.18 | Rubber Hand Illusion versus Artificial Rubber Hand Illusion setup. .  | 57 |
| 4.1  | Ottobock Hand . . . . .   | 59 |
| 4.2  | (a) System Architecture. The system comprises e-skin with 64 sensing points, charge amplifier assessing the response of each taxel to mechanical stimulus, data acquisition card to sample the signals, and multichannel stimulator with flexible matrix electrodes integrating 32 electrode pads to deliver the electrotactile stimulation to the subject. (b), (c) E-skin and Electrode Array: Sensor Fusion. The taxel signals are processed using integrative method and the signals from the neighboring taxels (2 x 2) are fused in order to map more taxels to less electrode pads (red and black dashed lines). . . . . | 62 |

|     |  |    |
|-----|--|----|
| 4.3 | Electrical response (taxel signals) due to a pressure applied to the e-skin using a finger. The mechanical stimulus comprised a step input (pressure-hold-release) delivered to a single taxel. TOP: color map representing a snapshot of all the taxel signals while pressure is applied for (a) raw and (b) processed data. BOTTOM: signal traces for the stimulated taxel (inset, yellow arrow) and its neighbors for (c) raw and (d) processed data. The signal plots have the line color of the corresponding taxels (see inset). The signals are zero-mean but were shifted using offsets for the sake of plot readability. The contact and release events elicit positive and negative deflections in the raw signals, respectively. Processed signals are clean from spurious activations and capture also the static properties of the mechanical stimulus. . . . . | 65 |
| 4.4 | Touch Modalities. Movement test patterns applied to the e-skin. Each line in this table corresponds to a single touch modality (pattern). S (Shape), T (Trajectory) and D (Direction): touch modality features. .  | 68 |
| 4.5 | Example of movement test patterns applied to the e-skin by an experimenter using a roller. TOP: longitudinal (left), transversal (center), diagonal (right) single lines. MIDDLE: square (left), rectangular (center), triangle (right). BOTTOM: letters: T and L (left), N and mirrored V (center), X and mirrored T (right). . . . .   | 69 |
| 4.6 | Overall success rate of recognizing either (a) all three movement features (shape, trajectory, direction) together, or (b) shape, (c) trajectory and (d) direction provided that the shape has been correctly predicted. This analysis is performed for single lines (RED bars), geometric shapes (BLUE bars) and letters (BLACK bars). Bars and stars indicated statistical significance (*, $p < 0.05$ ; **, $p < 0.01$ ; ***, $p < 0.001$ ) . . . . .   | 70 |
| 4.7 | Success rate of recognizing the shape (a), trajectory (b), and direction (c), for each category of lines (longitudinal, transversal, and diagonal indicated by RED bars) and of geometries (triangle, rectangle and square indicated by BLUE bars). Bars and stars indicated statistical significance (*, $p < 0.05$ ; **, $p < 0.01$ ; ***, $p < 0.001$ ). . . . .  | 72 |



|      |   |     |
|------|---|-----|
| 4.8  | Three best recognized patterns: single lines (a), geometries (b), letters (c). For each panel, pictures at the top illustrate how each pattern was applied to the e-skin, pictures below indicate the electrodes activated during electrotactile stimulation to communicate the pattern to the subject, while the success rate and the textual code (Table 1) for the pattern are reported at the bottom. . . . . | 73  |
| 4.9  | Rubber hand integrating the e-skin. . . . .   | 76  |
| 5.1  | Several examples of touch modalities ( Source [183]). . . . .   | 81  |
| 5.2  | Machine Learning approach. . . . .  | 82  |
| 5.3  | Online classification exploiting the Tensorial Kernel approach. . . . .   | 84  |
| 5.4  | Examples of SVM classification method. . . . .  | 85  |
| 5.5  | Unfolding of the $(I_1 \times I_2 \times I_3)$ -tensor A to the $(I_1 \times I_2 I_3)$ -matrix $A_{(1)}$ , the $(I_2 \times I_3 I_1)$ -matrix $A_{(2)}$ , and the $(I_3 \times I_1 I_2)$ -matrix $A_{(3)}$ (source [22]). . . . .   | 88  |
| 5.6  | Tested touch modalities: brushing a paintbrush, sliding the finger, and rolling a washer (source [186]). . . . .  | 90  |
| 5.7  | General flow diagram for the hardware implementation of the one sided Jacobi algorithm for SVD computation. . . . .   | 91  |
| 5.8  | Architectures definition based on the different implementation methods (source [187]). . . . .  | 92  |
| 5.9  | Power consumption versus number of training tensors - three touch modalities (source [196]). . . . .  | 93  |
| 5.10 | Power consumption versus number of touch modalities 100 training tensors (source [196]). . . . .  | 93  |
| 6.1  | Block schematic of the Inexact Speculative Adder (ISA) [200]. Every speculative path comprises a carry speculation block (SPEC), an adder block (ADD) and a double-direction error compensation block (COMP).101  |     |
| 6.2  | Example of arithmetic computation in an ISA with 4-bit ADD, 2-bit SPEC, 1-bit COMP correction and 2-bit COMP reduction. . . . .   | 102 |
| 6.3  | Architecture of the CORDIC in rotation mode. . . . .  | 105 |
| 6.4  | Normalized costs and relative errors for both sine and cosine computations of the approximate CORDIC implementations. . . . .   | 107 |
| 7.1  | Trends of average power dissipation for comparison TFP-mult vs. b32-mult [241]. . . . .   | 112 |

|      |  |     |
|------|--|-----|
| 7.2  | Representation of the significand $m$ : $f$ fractional bits; L last bit; G guard bit; R round bit; T sticky bit. . . . .           | 115 |
| 7.3  | FP Rounding: $\circ$ rounding position. . . . .  | 115 |
| 7.4  | Rounding in variable position for $m = 24$ and $m = 21$ . RW holds the position of the rounding bit, marked as $\bullet$ . . . . . | 115 |
| 7.5  | Architecture of TFP-mul. . . . .   | 116 |
| 7.6  | Architecture of TFP-add (significand only). . . . .  | 117 |
| 7.7  | Neural network structure. . . . .  | 119 |
| 7.8  | Gradient Descent Algorithm in Neural Network. . . . .  | 119 |
| 7.9  | Example of activation functions. . . . .   | 120 |
| 7.10 | Neural network with two hidden layers (depth=2). . . . .   | 121 |
| 7.11 | Training: interpolated functions by NN of Figure 7.10. . . . .   | 121 |
| 7.12 | Approximation error for NN interpolated function. . . . .  | 123 |
| 7.13 | Simulation flow: SVD and TFP-SVD. . . . .  | 127 |
| 7.14 | Average error for SYMM (dashed lines) and JR (continuous lines) algorithms under TFP rounding modes. . . . .                       | 129 |
| 7.15 | Average error for SVD algorithm under TFP rounding modes. . . . .  | 129 |
| 7.16 | Average power dissipation ( $mW$ ) for TFP multiplier at 100 MHz. . . . .  | 130 |
| 7.17 | Energy to perform the JR required in the OSJ algorithm. . . . .  | 131 |

# List of Tables

|     |   |     |
|-----|---|-----|
| 2.1 | Characteristics of different sensors for the e-skin fabrication ([45]). . .                     | 20  |
| 2.2 | Design requirements for tactile sensing system (adapted from [7, 13, 20]).                      | 21  |
| 5.1 | Computational complexity of the studied ML algorithms (source ). . .                            | 90  |
| 6.1 | Approximate techniques at different layers (source [205]) . . . . .                             | 98  |
| 6.2 | ISA configurations of the approximate CORDIC implementations . .                                | 107 |
| 7.1 | Average power dissipation for TFP-mult and b32-mult (RTN) [241]. .                              | 112 |
| 7.2 | Dynamic range for some FP formats. . . . .  | 113 |
| 7.3 | Average error and average power dissipation for TFP training. . . .                             | 122 |
| 7.4 | Quantization error for different TFP precisions. . . . .  | 124 |
| 7.5 | Quantization error and average power dissipation for flexible inference<br>( $e = 8$ ). . . . . | 125 |
| 7.6 | Number of JR depending on the convergence criteria. . . . .                                     | 131 |
| B.1 | Technical Characteristics of MaxSens Stimulator . . . . .                                       | 158 |

# Part I

## Introduction



# Chapter 1

## Introduction

Interactions with our surroundings make up a major part of our lives. We receive information about the world around us every second, every day, through the five major sensory modalities: one of these is touch. Unlike other senses, the sense of touch is not restricted to any particular part of our body as we receive touch information through the skin extending all over it. However, there are parts of the body more sensitive than others, e.g., fingers are very sensitive to touch.

When something comes into contact with our body, even without looking at it, we can determine if it is hard or soft, cold or hot, wet or dry, and find out its shape or texture. Besides, we can feel pain when a sharp object is touching us or we can realize if something, for example a bottle, is slipping out of our hands. We routinely use the sense of touch with remarkable ease, effortlessly, numerous times during the activities of our daily life.

Given technological advances in mechanics, electronics, etc. (e.g., smaller motors, more powerful processors, and better sensors built within the hand), in the last few years research on prostheses has been exploring a way to substitute a missing sense of touch by using an alternative intact sense. To date, modern prostheses are successfully controlled by using muscle electrical activity. A number of methods have been proposed for measuring the user muscle activity and translating it into the prosthesis control commands [1]. For example, prosthesis users can open or close the prosthetic hands, but they can neither feel if something or someone is touching them over the prosthesis nor perceive the temperature or roughness of objects. Prosthesis users are helped by looking at an object, but they cannot detect anything otherwise. Their sight gives them most information. Without the sense of touch, they would have no physical self-awareness.

There are between 50 and 270 new upper-limb amputees every year in Europe, with an estimated population of around 1900 traumatic upper-limb amputees and 94000 total upper-limb amputees. In the USA, there are an estimated 1.6 million people live without a limb, 34% of which are upper limbs. From these numbers it appears how important are improvements in user compliance in the prosthetic market. Restoring sensory feedback is a long-standing challenge in prosthetic research.

The idea to provide a sensory feedback to an amputee controlling upper-limb prosthesis is not a new concept [2]. This was quite a popular research topic during 60s and 80s, but new technologies now available allow the implementation of better solutions. Vincent Hand [3] is the only device recently supplied that can be used as a replacement for the lost limb integrating a simple sensory feedback.

In addition to increase prosthesis functionality (tactile information is given back to the user), providing sensory feedback to users could foster embodiment (amputees may have the impression that the feeling prosthesis is part of their own body), having a direct potential for therapeutic applications and, consequently, substantially improving the quality of life of prosthesis users.

This proposed research would address this gap.

Cosmic Lab (<http://www.cosmiclab.diten.unige.it/>) group has been working on a system to close the loop in prostheses, restoring the sense of touch in prosthesis user. As shown in Figure 1.1, the idea is to cover the prosthesis with an electronic skin providing a dense matrix of sensitive piezoelectric tactile receptors (taxels), to embed an electronic system integrating signal conditioning, data acquisition, and tactile data processing in the prosthesis pocket, and to translate and transmit tactile sensor data to the user through electrotactile stimulation.

The first electronic skin was a bench prototype based on Piezoelectric polymer films of Polyvinylidene Fluoride (PVDF), which meets the requirements of mechanical flexibility, high sensitivity, detectability of dynamic contact events, wide dynamics (light/strong touch), low cost, light weight and robustness. For the consecutive realization, screen printing technology has been used for ad-hoc electronic skin design, optimizing taxel size and sensor pitch according to application requirements. An elastomer layer has been integrated on top for stress transmission and sensor protection.

For the embedded electronic system that has to acquire the tactile data, process and extract structured information, a dedicated real-time hardware implementation of tactile data processing algorithms has been studied. The first studies have highlighted

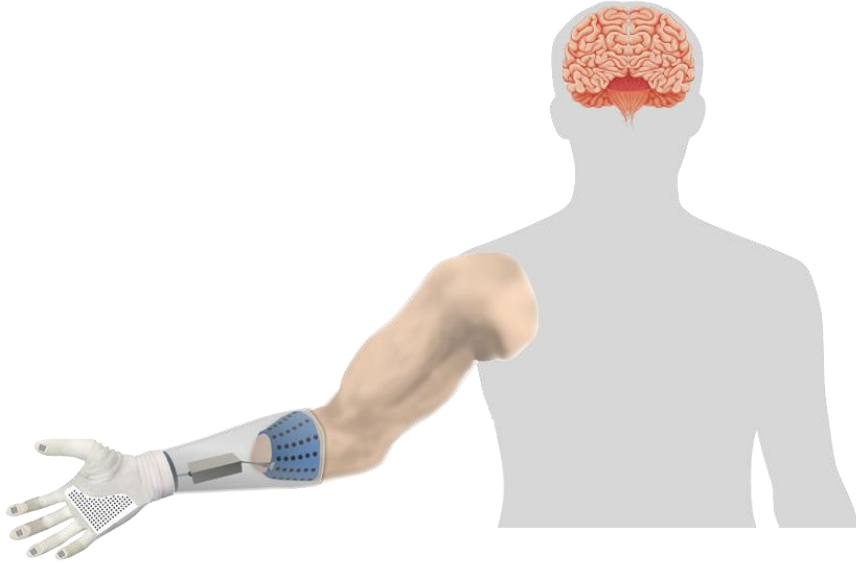


Figure 1.1: Towards the restoration of sensory feedback. From left to right: prosthetic hand covered with electronic skin, electronics system in the prosthesis pocket, electro tactile stimulation delivered to the skin of the residual limb.

that special attention must be given to the features of this system, such as power consumption, complexity, and delay. The requirements related to the development of embedded data processing unit for e-skin are still far from being achieved. Therefore, new methods and techniques to reduce hardware complexity and power consumption of the embedded electronic system have been investigated.

Electrotactile stimulation delivering low-level electrical current pulses to the skin to depolarize skin afferents, thereby eliciting tactile sensations has been studied and tested. Methods have been developed to interpret tactile data in real time and retrieve touch information such as contact location, area and duration and to efficiently deliver artificial tactile information (recorded by artificial skin) to the prosthesis user through multichannel electrocutaneous stimulation.

Within this project, this thesis is focused on the tactile feedback system, allowing the communication of a mechanical interaction from the electronic skin to prosthesis users, and the dedicated embedded implementation of tactile data processing algorithms.

The conducted work regards different aspects of the Cosmic Lab project from system to architectural level. At system-level thesis includes:



- Design of the experimental setup, experimental protocol and pilot tests;
- Design of algorithms to process tactile data acquired from the electronic skin;

and at architectural level:

- Design flow for the implementation of digital circuits for both FPGA and integrated circuits;
- Techniques for the power management of embedded systems for Machine Learning algorithms.

Chapter 2 presents the system approach for an artificial skin implementation. Starting with the explanation of why the human skin plays an important role in the human life and why researchers are trying to reproduce an artificial skin, an overview of the human sense of touch and of human skin, and then a perspective of the artificial skin are introduced.

Artificial or electronic skin being able to mimic functionalities of human or animal skin would open the door to many possible applications including soft robotics, prosthetics, artificial intelligence and health monitoring. Interest in large-area networks of electronic devices inspired by human skin is motivated by the promise of creating autonomous artificial intelligence, medical diagnostics, and replacement prosthetic devices capable of providing the same, if not better, level of sensory perception than the organic equivalent.

Then, the state of the art of electronic skin is described as a system, detailing all the aspects such as: tactile sensors, the electronic interface and the embedded electronic system.

Chapter 3 concerns the study of tactile feedback system. What the sensory feedback is and what techniques are used towards the restoration of the sensory feedback in prosthetics are discussed considering the state-of-the-art of many tactile transmission methodologies. Many studies are focused on evaluating different methods to provide prosthesis with sensory feedback.

More specifically, the chapter is focused on electrocutaneous stimulation: the tactile feedback technique investigated in the conducted experiments/studies. The aim is to increase the reliability of information transmission when using non-invasive interfaces ,i.e., electrocutaneous stimulation, with many stimulation points. A novel dual-parameter modulation (intensity and frequency) is developed and tested to assist

the subject in correctly identifying an active pad within the matrix. Furthermore, for the first time, the performance of a matrix electrotactile interface is compared to that of the natural skin being mechanically stimulated over analogous contact areas.

In Chapter 4, the system implementation to transmit mechanical information from a multi point tactile sensor (e-skin) to the human subject using multichannel electrotactile stimulation is described. The system is evaluated by assessing the ability of the human subjects to perceive the properties characterizing dynamic and versatile mechanical interaction with the skin. Precisely, from electrostimulation the subjects try to recognize spatial and temporal features of the mechanical stimuli moving over the e-skin surface. The system comprises an advanced tactile sensor with many sensing elements, acquisition electronics and multichannel stimulator connected to flexible matrix electrodes placed on the natural skin (forearm).

Although the concept of applying e-skin technology for somatosensory feedback in prosthetics has been proposed earlier [4], the present study is the first evaluation of an online system implementing the prospective communication link, i.e., the information transmission from e-skin to the human subject.

Chapter 5 introduces the topic about how to give sensor data back to the prosthesis user and how to help the human brain to successfully interpret the elicited artificial tactile information. Among open questions in prosthetics is which kind of information - whether *raw* data or *processed* - about a touched object should be sent back to the user. One approach would be to send the sensor signals directly to the user, who needs to meaningfully interpret this information. Alternatively, learning from robotics, sensor data can be locally processed at the body periphery (prosthesis socket with embedded electronics) and high level tactile information (e.g., texture properties, grasp stability) can be extracted and delivered to the user.

In particular, a Machine Learning algorithm is employed for tactile data processing unit with high level information extraction and, then, an analysis regarding the hardware implementation requirements for processing unit is reported. The analysis demonstrates that the hardware implementation of the processing unit is still far from being achieved as a processing unit of a system to close the loop in prosthetics should be real-time and low-power. Therefore, methods and techniques to reduce hardware complexity and power consumption of the embedded electronic system have to be explored.

Chapter 5 supports Chapter 6 and Chapter 7.

Approximate computing has become a major field of research in the sense that it could significantly improve energy efficiency and performances of modern digital circuit. More specifically, Chapter 6 aims at implementing approximate circuit techniques in the FPGA implementation of real-time tactile data processing for e-skin system application. It focuses on the implementation of the Coordinate Rotation Digital Computer algorithm, as it is used for several computing tasks such as Singular-Value Decomposition, the most computationally expensive algorithm for Machine Learning approaches that has ever been considered in Chapter 5. This first attempt of approximate CORDIC implementation on FPGA uses Inexact Speculative Adder (ISA) architectures, a circuit-level technique optimized for high-speed arithmetic computations.

As the current challenge of development of an embedded and real-time system for Machine Learning data processing relies on an efficient implementation and low power requirements, in Chapter 7 an approximate computing technique capable of improving energy efficiency by relying on the ability to tolerate some loss of accuracy is introduced. Chapter 7 provides an overall description of the Tunable Floating-Point and a detailed analysis of two Machine Learning algorithms implementing the Tunable Floating-Point approach.

The use of Tunable Floating-Point precision is an interesting approach in computations, which allows to arbitrarily set the precision for each operation by selecting a specific number of bits for significand and exponent in the floating-point representation. By profiling and tuning the precision for a given algorithm, an efficient trade-off can be achieved to allow an acceptable target error while lowering the energy cost of computations.

Machine Learning algorithms are exploited to evaluate the Tunable Floating-Point, reducing power consumption while still producing acceptable accuracy of results in both algorithms. The Singular-Value Decomposition (SVD) algorithm is investigated as it represents the most computationally expensive algorithm for Machine Learning approaches that has been considered for this real-time embedded prosthetics.

# Part II

## System and Applications



# Chapter 2

## Electronic Skin System

### 2.1 Introduction

This chapter contextualizes the general framework in which the herein presented project was developed illustrating the system for an artificial skin implementation. Firstly, an overview of the human sense of touch and of human skin, then a perspective of the artificial skin are presented. Secondly, the state of the art of electronic skin is described as a system: starting from tactile sensors, going through the electronic interface and the embedded electronic system, all the way to a communication interface.

Among various human sensations like sight, hearing, taste and smell, touch is a critical co-existing sensation required to interact with the surrounding environments [5]. The exquisite sensitivity provided by the sense of touch enables us to discriminate various surface textures, precisely grasp and manipulate objects, etc. Unlike the other senses based on discrete sensory organs, the sense of touch arises from receptors distributed throughout the entire body [6]. In fact, these receptors are embedded in the skin which is the outer covering of the body.

In humans, the skin is the largest organ of the integumentary system. Because it interfaces with the environment, skin plays an important immunity role in protecting the body as anatomical barrier from pathogens and damage between the internal and external environment. Additionally, its other functions are insulation, temperature regulation, sensation.

The sense of touch also plays a significant role in many applications in robotics [7,8], and advanced prosthetics [9]. Many studies concern tactile sensing technologies for robot hands, for minimal invasive surgery [10], biomedical applications, slip detection in hand prostheses [11], whole-body tactile sensing, large area tactile skins [12] and

so on. For instance, as robots move from laboratories to domestic environments (real-world), they will be required to perform manipulation tasks in unstructured environments. Such robots must be able to achieve sophisticated interactions with the environment and to perform complex tasks such as grasping objects with arbitrary unknown shapes [13], and avoiding slip while applying minimal force to the grasped objects. In these settings, the ability of tactile sensing becomes a particularly valuable and desirable.

During the past decade tremendous efforts and progress have been made by the industry and academia to mimic the sense of touch in humans. Besides, researchers are also addressing the development of electronic skin system consisting of the mechanical arrangement of the skin itself (i.e. soft or rigid mechanical support, structural and functional material layers, etc.) plus an electronic embedded system in charge of locally processing tactile data, which has to be transmitted to a robot or a prosthesis user, for example in robot or prosthetic applications.

## 2.2 Sense of Touch

According to Loomis and Lederman [14], the sense of touch in humans comprises three main submodalities, i.e., cutaneous, kinesthetic, and haptic characterized on the basis of the site of sensory inputs.

The cutaneous sense receives sensory inputs from the receptors embedded in the skin. In fact, the cutaneous system involves physical contact with the stimuli and provides awareness of the stimulation of the outer surface of body by means of receptors in the skin and associated somatosensory area of central nervous system (CNS).

The kinesthetic system, receiving sensory inputs from the receptors within muscles, tendons, and joints [14, 15], provides information about the static and dynamic body postures on the basis of 1) afferent information originating from the muscles, joints, and skin; and 2) efference copy, which is the correlate of muscle efference available to the higher brain.

The haptic sense perceives heat, cooling, and various stimuli that produce pain by using significant information about objects and events both from cutaneous and kinesthetic systems [14, 16].

The human sense of touch deals with the spatiotemporal perception of external stimuli through a large number of receptors (e.g., mechanoreceptors for pressure/vibration, thermoreceptors for temperature, and nociceptors for pain/damage [17]) that are distributed all over the body with variable density. The response to mechanical stimulus

is mediated by mechanoreceptors that are embedded in the skin at different depths. Their number, per square centimeter area, is estimated to be 241 in the fingertips and 58 in the palm of adult humans [18].

In details, thanks to studies about the tactile afferents in the late 1960's [19], there are four main types of mechanoreceptors embedded in the human skin throughout the body [20], and each is responsible for the reception of specific stimuli; these four include Pacinian corpuscles, Meissners corpuscles, Merkel discs, and Ruffini cylinders, all of which can be divided into two categories: slowly adapting (SA) mechanoreceptors and fast adapting (FA) mechanoreceptors [21]. The slowly adapting mechanoreceptors including Merkel discs (SA-I mechanoreceptors) and Ruffini cylinders (SA-II mechanoreceptors) respond to low frequency stimuli, and they describe the static properties of a stimulus. In contrast, high frequency stimuli make the fast adapting mechanoreceptors, i.e., Meissners corpuscles (FA-I mechanoreceptors) and Pacinian corpuscles (FA-II mechanoreceptors) fire in response. The border distinctness and the size of the receptive field differentiate SA-I mechanoreceptors from SA-II mechanoreceptors, and FA-I mechanoreceptors from FA-II mechanoreceptors. In other words, the receptive fields of SA-I mechanoreceptors and FA-I mechanoreceptors have more distinct borders and smaller sizes than those of SA-II mechanoreceptors and FA-II mechanoreceptors. Figure 2.1 illustrates the characteristics of human mechanoreceptors and summarizes the characteristics of each mechanoreceptor [22].

Pacinian corpuscles are nerve endings responsible for sensitivity to deep-pressure touch and high-frequency vibration. They are located beneath the bottom layer of skin, called the dermis which is subcutaneous fat. They are considered fast adaptive receptors. The Pacinian corpuscles are oval shaped and are approximately 1 *mm* in length and 0.6 *mm* in diameter.

Meissner's corpuscles are responsible for sensitivity to light touch. They are located in the top layer of skin, called the epidermis. Similar to Pacinian corpuscles, they are fast adaptive receptors. Meissner's corpuscles are between 80 and 150 *m* in length and between 20 and 40 *m* in diameter.

Merkel discs provide information regarding pressure, vibration, and texture. They are the most sensitive of the four mechanoreceptors to vibrations at low frequencies, at around 15 *Hz*. They are located in epidermis. In contrast to the two previous receptors, they are slowly adaptive receptors. This means they have sustained response to stimulation since they are not capsulated, as are the two previous ones.



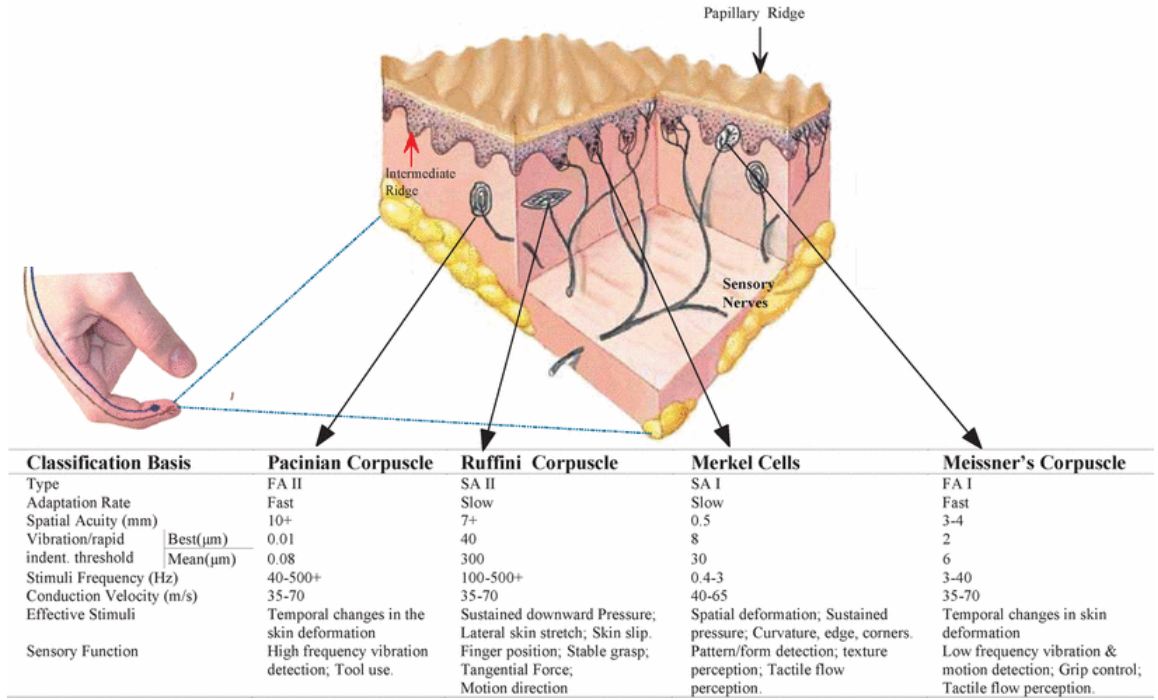


Figure 2.1: Section of glabrous skin showing physical location and classification of various mechanoreceptors [7].

Ruffini cylinders are sensitive to lateral skin stretch and contribute to the kinesthetic sense and control of finger position and movement. They register mechanical deformation within joints, more specifically angle change, with a specificity of up to 2, as well as continuous pressure states. They are also believed to be useful for monitoring slippage of objects along the surface of the skin, allowing modulation of grip on an object. Ruffini cylinders are located in the dermis layer.

## 2.3 Electronic Skin

Electronic skin refers to flexible, stretchable and self-healing electronics that are able to mimic functionalities of human or animal skin. Advances in electronic skin research, that focuses on designing materials that are stretchy, robust, and flexible, would open the door to many possible applications including soft robotics, prosthetics, artificial intelligence and health monitoring.

Interest in large-area networks of electronic devices inspired by human skin is

motivated by the promise of creating autonomous artificial intelligence (e.g., robots), medical diagnostics, and replacement prosthetic devices capable of providing the same, if not better, level of sensory perception than the organic equivalent. Endowing robots with sensing capabilities could extend their range of applications to include highly interactive tasks, such as caring for the elderly [23], and sensor skins applied on or in the body could provide an unprecedented level of diagnostic and monitoring capabilities [24].

An artificial skin with such sensory capabilities is often referred to in the literature as sensitive skin, smart skin, or electronic skin (e-skin).

In 1974, Clippinger demonstrated a prosthetic hand capable of discrete sensor feedback [25]. Nearly a decade later, Hewlett-Packard (HP) marketed a personal computer (HP-150) that was equipped with a touchscreen, allowing users to activate functions by simply touching the display. It was the first mass-marketed electronic device capitalizing on the intuitive nature of human touch. In 1985, General Electric (GE) built the first sensitive skin for a robotic arm using discrete infrared sensors placed on a flexible sheet at a resolution of 5 cm. The fabricated sensitive skin was proximally aware of its surroundings, allowing the robot's arm to avert potential obstacles and effectively maneuver within its physical environment. Despite the robotic arm's lack of fingers and low resolution, it was capable of demonstrating that electronics integrated into a membrane could allow for natural human-machine interaction.

In the 1990s, scientists began using flexible electronic materials to create large-area, low-cost and printable sensor sheets. Jiang et al. proposed one of the first flexible sensor sheets for tactile shear force sensing by creating silicon (Si) micro-electromechanical (MEM) islands by etching thin Si wafers and integrating them on flexible polyimide foils. Around the same time, flexible arrays fabricated from organic semiconductors began to emerge that rivaled the performance of amorphous Si [26]. Just before the turn of the millennium, the first Sensitive Skin Workshop was held in Washington DC under the aegis of the National Science Foundation and the Defense Advanced Research Projects Agency, bringing together approximately sixty researchers from different sectors of academia, industry, and government. It was discovered that there was significant industrial interest in e-skins for various applications, ranging from robotics to health care.

Significant progress in the development and advancement of e-skin has been achieved in recent years, and particular emphasis has been placed on mimicking the mechanically compliant yet highly sensitive properties of human skin. Suo and coworkers have developed stretchable electrodes [27], and Rogers and coworkers have

transformed a typically brittle material, Si, into flexible, high-performance electronics by using ultrathin (100 nm) films connected by stretchable interconnects [28]. Someya and coworkers have fabricated flexible pentacene-based organic field-effect transistors (OFETs) for large-area integrated pressure-sensitive sheets with active matrix readout [29], while Bauer and coworkers have investigated novel pressure sensing methods using foam dielectrics [30] and ferroelectrets [31] integrated with FETs. Bao's group has investigated the use of microstructured elastomeric dielectrics for highly sensitive capacitive pressure sensors [32] and has developed a composite conductive elastomer exhibiting repeatable self-healing and mechanical force sensing capabilities [33]. Other groups have developed stretchable optoelectronics, including light-emitting diodes (LEDs) [34] and organic photovoltaics (OPVs) [35] for integration with e-skin. A timeline outlining the major milestones towards the development of e-skin is depicted in Figure 2.2.

Important considerations for the development of e-skin are the choice of materials used in its fabrication and the ability to confer the mechanical properties of human skin (e.g. stretchability and flexibility) into its artificial counterpart.

A remarkable characteristic of human skin is its ability to stretch and flex with bodily movements without incurring damage. While arrays of flexible electronics have been developed by using very thin plastic substrates [36], stretchable devices have been more difficult to achieve, and new processes and materials are often required [37]. In general, stretchable devices can be fabricated by developing devices comprised of intrinsically stretchable materials or by appropriate geometrical arrangement of conventional materials.

The human skin is also able to perceive simultaneous and multiple stimuli. Hence, in order to simulate the multifunctional nature of skin, e-skin should sense many parameters simultaneously. The tendency of simple, discrete devices to respond to multiple stimuli can be advantageous for this purpose.

Moreover, the human skin and e-skin should be aesthetically similar for allowing for the possibility of integrating e-skins with the human body in prosthetic applications, for example. In this direction, much progress has been made toward developing biocompatible and biodegradable devices.

E-skin could enhance capabilities, such as improved sensitivity, higher receptor density, and faster response times, and could endow robots and prosthetics with these capabilities that surpass those of our own skin.

## Evolution of E-Skin

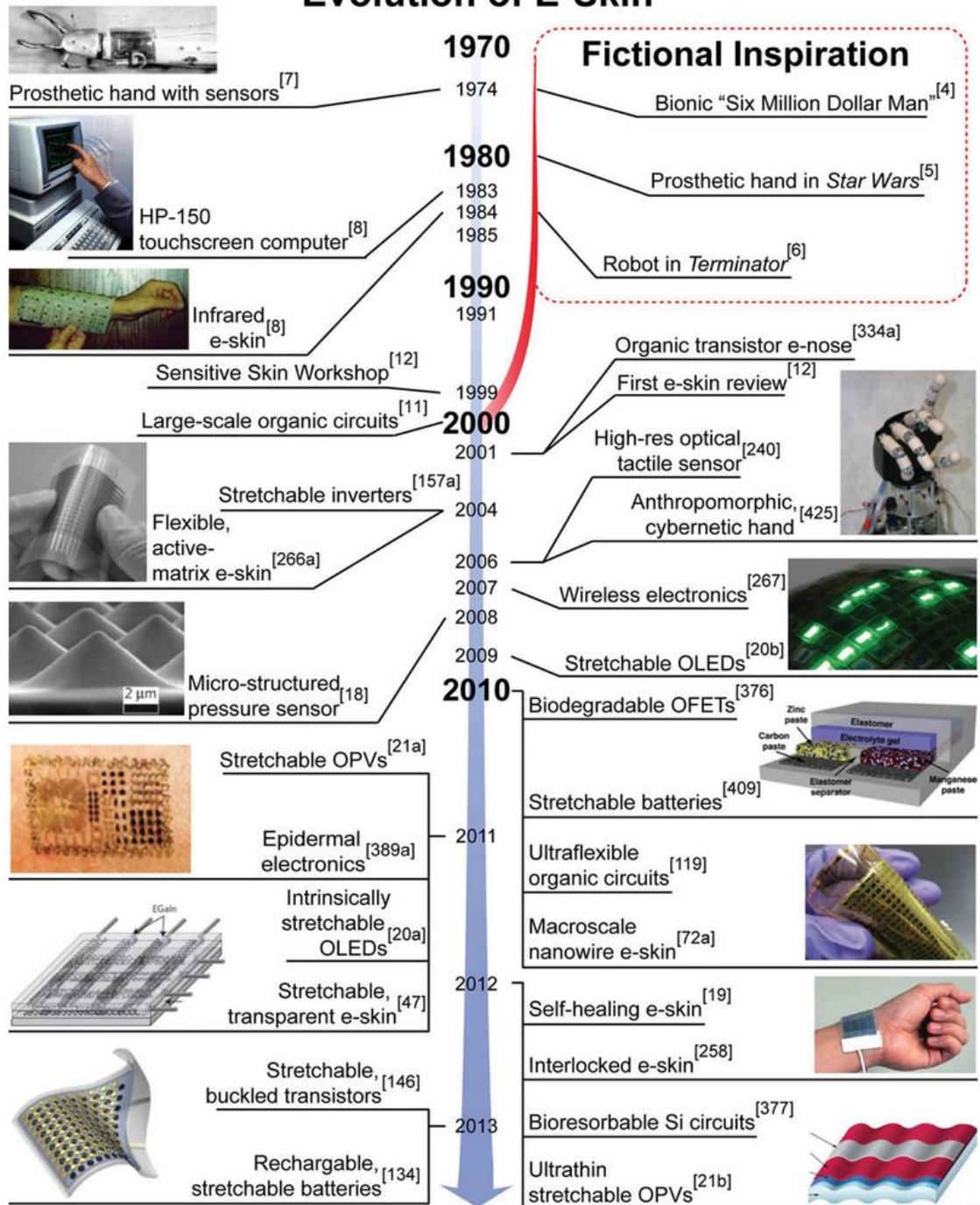


Figure 2.2: A brief chronology of the evolution of e-skin [38].

## 2.4 Electronic Skin as a System

The development of the e-skin is a very challenging goal which should be tackled from a system perspective. The e-skin is usually intended as a hybrid stack-wise arrangement which incorporates tactile sensing (structural and functional materials, signal conditioning and acquisition, signal processing) and interpretation. Sensory inputs similar (but not limited) to those possessed by humans are essential to provide the necessary feedback to explore the environment and interact with objects.

In addition to tremendous effort being concentrated on the development of the e-skin hardware, the design of an effective digitization which requires software to provide a communication interface with the user is needed. The signals produced from sensors in e-skins have to be digitalized (from analog to digital signal) for interfacing with modern digital microprocessors. After digitization, software algorithms could be used to process, interpret, and act on the large amounts of data in a coherent fashion. Further integration of other signal processing methods (such as wireless technologies) could be used to create super skins, such as implantable biofeedback devices for healthcare applications.

An electronic skin system should consist of:

1. Sensing arrays being able to reproduce the sense of touch in humans.
2. Interface electronics to convert from analog to digital tactile signals.
3. Tactile data processing and decoding system.

An example of this general structure is shown in Figure 2.3 (adapted from [39]). The bottom layer (substrate) is made of a structural material which can be rigid (e.g. the robot mechanical structure) or soft. Next layer (electronic layer) hosts the electronic circuits. Conventional electronics is typically integrated on very hard and flat (brittle) surfaces. Here the need is to conform to curved surfaces, requiring flexibility but also stretchability, to a certain extent, to follow all movements and deformations of the parts into which the electronic layer is integrated. The adoption of a flexible substrate does not necessarily guarantee the flexibility of the entire electronic circuit, as a too dense or not well organized layout may drastically limit the flexibility of the overall structure. In fact, even if the substrate is stretchable, the routing lines are intrinsically not, unless a dedicated design is adopted. Some interesting concepts are related to the creation of compliant and stretchable interconnections [40,41] and a very widespread approach to materials and mechanics for stretchable

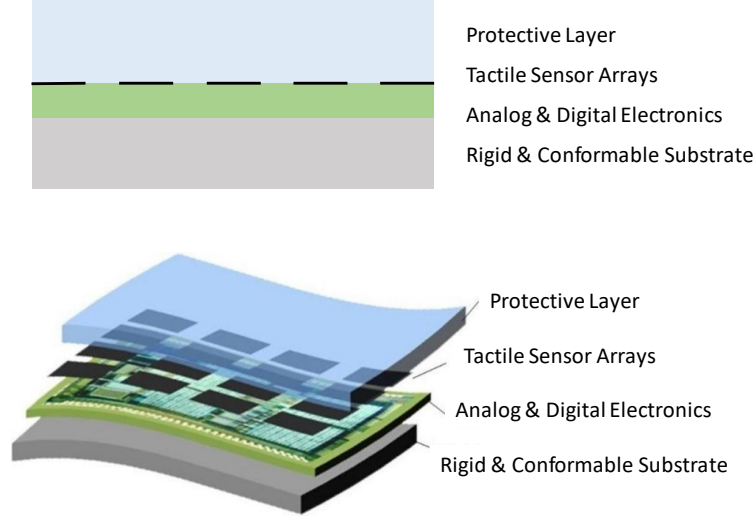


Figure 2.3: General physical structure of the e-skin system [42].

electronics is contained in [42]. Requirements on conformability and stretchability put severe constraints on the reliability of the electronics, as mechanical stress on the electronic circuits can cause faults on interconnections and circuits itself. Next, sensors are embedded in a protective layer. The geometry of the sensor array (i.e. overall area, sensor size, sensor pitch, sensor distribution) depends on the transducer technology and on the given application requirements. The protective layer, which is usually polymer based (e.g. PDMS), protects the whole from damages induced by contact with objects, environmental chemical agents and water, etc. Moreover, it implements a mechanical filtering of the input stimulus, concentrates/distributes the mechanical stimulus onto the sensor array below depending on the thickness and the compliance of the layer. As a consequence, e-skin spatial resolution depends on the sensor geometrical arrangement as well as on the features of this protective layer.

### 2.4.1 Tactile Sensing

In the development of artificial skin, the materials chosen for its fabrication should reflect the flexibility and stretchability of natural human skin. To achieve these properties, the choice of materials is critical, and technological advancements in e-skin have been largely possible through the development of new materials and processing methods for the fabrication of stretchable and flexible devices. In addition to mechanical compliance, good electrical performance and compatibility with large-area processing techniques are important to create highly functional, low-cost devices.

Different kind of sensors (see Table 2.1) could be embraced for the fabrication of an e-skin showing multimodal sensing capabilities [43], bendability, flexibility and stretchability, hopefully shrink and wrinkle ability as human skin has [44]. On the basis on what happens in human skin, transduction technologies and corresponding sensors should enable such capabilities as normal and shear force sensing, tensile strain monitoring and vibration detection (at least up to 800 Hz) [38], and e-skin featuring a large frequency bandwidth which spans from zero to 1kHz is desirable. According to the application, the spatial resolution (defined as the smallest distance between two distinguishable contact points [13]) should range from a minimum of 1mm to a maximum of 20-30 mm. Detectable force should span in a range of three orders of magnitude (e.g. 1- 1000 g [38]). Even if human skin features high hysteresis, it is preferable that e-skin presents a low hysteresis, to avoid significant processing and complex electronics. The requirements outlined above together with some others (extracted and adapted from literature, in particular from [7, 13, 20] are summarized in Table 2.2. The requirements are general and can be satisfied totally or partially, according to the target application. Many of the previous requirements are satisfied by many examples reported in the literature, even if, to our knowledge, no e-skin implementation satisfies all of them.

| Tactile sensors              | Working principle   | Advantage   | Disadvantage   |
|------------------------------|---|---|--|
| <b>Piezoresistors</b>        | Its resistance varies with the deformation caused by applied force.   | Simple electronics; High sensitivity;<br>Ease of integrating in MEMS;<br>Resistant to interference.   | Hysteresis; Temperature sensitivity; Fragile and rigid; High cost.                         |
| <b>Capacitive sensors</b>    | Its capacitance varies with the deformation caused by applied force.  | Sensitivity of small force change;<br>Reliability; Large dynamic range, suitable for both dynamic and static force measurement;<br>Low temperature sensitivity;<br>Low power consumption. | Limited spatial resolution;<br>Noise sensitivity;<br>complex electronics.                  |
| <b>Piezoelectric sensors</b> | An electric voltage will be produced when a force applied to it.      | No need for power supply;<br>High reliability; Fast dynamic response.   | Low spatial resolution;<br>High temperature sensitivity; Inability to sense static value.  |
| <b>Optical sensors</b>       | The intensity or the spectrum of light varies with the applied force. | Immune to electromagnetic fields;<br>High spatial resolution.   | Fragile and rigid; Large size;<br>Inability to transparency and highly reflective surface. |

Table 2.1: Characteristics of different sensors for the e-skin fabrication ([45]).

| Design criteria   | Character  |
|---|--|
| Detectable force range (Dynamic range)                    | 0.01 N– 0 N(1000 : 1)  |
| Tactile Sensing Element (Taxel)<br>pitch (for array only) | $\leq 1\text{mm}$ for small sensing areas $\geq 5\text{mm}$<br>for large less sensing areas        |
| Spatial resolution  | $\leq 1\text{mm}$ for fingertips 5mm 20–30mm<br>(e.g. limbs, torso, etc.)                          |
| Sensor frequency bandwidth<br>(sensor response time)      | About 1kHz (1ms)   |
| Temporal variation  | Both dynamic and static.   |
| Mechanical sensing detection capability                   | Normal and shear forces; vibrations.   |
| Sensor Mechanical characteristic                          | Flexible, stretchable, conformable and<br>soft, robust and durable.                                |
| Sensor Electrical characteristic                          | Low power, minimal wiring and cross<br>talk, electrically and magnetically minimal<br>sensitivity. |
| Sensor response   | Monotonic, not necessarily linear, low<br>hysteresis, stable and repeatable.                       |

Table 2.2: Design requirements for tactile sensing system (adapted from [7, 13, 20]).

**Piezoresistive sensors** transduce a change in the resistance of a device into a measurement of strain and have been investigated extensively. Piezoresistive polymer composites have been extensively investigated as strain- and force-sensitive materials because of their low cost and easy integration into devices. The change in resistance can be derived from several factors, including changes in the geometry of the sensing element for example.

**Capacitive sensors** for tactile sensing have demonstrated high strain sensitivity, compatibility with static force measurement, and low power consumption [46, 47]. A major advantage of capacitive sensors is that their governing equation is simple, which



simplifies device design and analysis. Furthermore, capacitive sensors are susceptible to interference from external sources [48].

**Piezoelectricity sensors** are able to generate a voltage in response to an applied force. The force causes a change in the length and separation between dipoles in the material [49], leading to the build-up of compensating charges on the electrodes. Owing to the high sensitivity of piezoelectric sensors to dynamic pressure and their fast response speed, they are often used to measure the vibrations associated with slip, and their transient sensing capabilities resemble those of the RA receptors in human skin. However, piezoelectric materials exhibit drift in sensor response over time and have unreliable static sensing properties.

**Optical sensors** convert a tactile input into an electrical output using light as an intermediate. These sensors consist of a light source, transmission medium, and detector. The modulation of light intensity through force-sensitive wave guides [50] or flexible optical fibers has been used to fabricate optical sensors. While these fiber-optic sensors had low wiring complexity, exceptional linearity, and negligible drift, they were not stretchable.

## 2.4.2 Interface and Acquisition System

Interface electronics includes blocks for signal conditioning and data acquisition [51–53]. Figure 2.4 shows a general functional block diagram of interface electronics illustrated in [12].

Signal Conditioning implements a set of circuit level functions such as low noise amplification, input/output impedance adaptation, setting the reference DC values (e.g. signal ground), lowpass or bandpass filtering (e.g. antialiasing lowpass filter) [53]. The output of the signal conditioning circuit is input to analog to digital (A/D) converter. Usually a dedicated signal conditioning channel is needed for each sensor element in the array with a single A/D converter. Signal conditioning channels (one for each sensor element of the array) are time multiplexed via an analog multiplexer at the input of the A/D converter.

Data acquisition involves addressing the signal conditioning channel and digitization (i.e. analog to digital conversion) of the analog input. The tactile sensing control can query and read tactile data from any sensor in the system. The ability to query

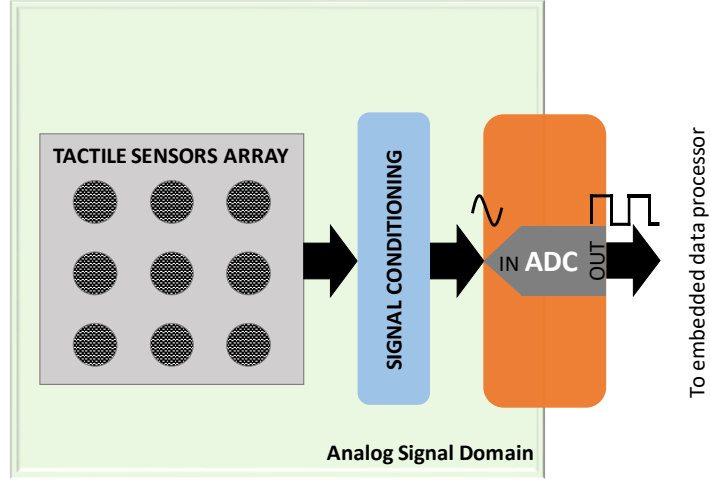


Figure 2.4: A general functional block diagram of the interface electronics ( [12]).

individual tactile sensors is helpful e.g. for diagnostics and calibration. Issues like wiring complexity also influence the interface electronics, in particular for large skin area arrays. The addressing can be serial and fixed [54] to decrease complexity at the expenses of an increase of array scanning rate. An alternative arrangement for signal conditioning and data acquisition is to translate the sensed signal into a frequency value, which is subsequently digitized and acquired via a digital counter. The advantage of this approach is the robustness of the sensor output signal with respect to noise and disturbances. Wiring is minimized as the sensor output is a single signal/wire. On the other hand, the acquisition time can be very long as it depends on the oscillating frequency and on the sensing resolution.

### 2.4.3 Tactile Data Processing and Decoding System

A step forward towards the integration of the e-skin into robots or prosthesis devices focuses on dedicated real-time hardware implementation of tactile data processing algorithms. The embedded electronic system should comply with severe constraints imposed by the application, e.g. real time response, reliability, low power consumption and low cost. The system should read the electrical conversion of the applied mechanical stimuli, extract and transmit structured or unstructured information.

A complete sensing system (hardware and software) for an anthropomorphic robotic hand which closely matches the low levels structure (up to the Tactile Data Transmission Level) is presented in [55]. The system can be interfaced to different transmission

protocols (e.g. CAN, USB, RS232). Signal processing tasks and classification (e.g. slip detection, contact, etc.) are implemented by the embedded electronic system. In [56], the basic sensing module (HEX-o-SKIN) approach is implemented by hexagonal PCB which hosts a microcontroller for data preprocessing (e.g. clustering, feature extraction, etc.) and transmission interface, and a bunch of custom and off-the-shelf sensors (e.g. proximity, pressure, vibration, temperature, etc.). HEX-o-SKIN can be arranged in a modular way to configure skin patches hierarchically interfaced to the robot backbone.

## 2.5 Conclusion

In the past decade, the e-skin development has accelerated dramatically owing to the availability of new materials and processes. Interest in e-skin has been driven by its potential to: 1) enable highly the development of interactive and versatile robots that are capable of performing complex tasks in less structured environments; 2) facilitate conformable displays and optics; and 3) revolutionize healthcare by providing biomimetic prostheses, constant health monitoring technologies, and unprecedented diagnostic and treatment proficiency.

Despite rapid progress, there is a continuing need for further development before the goal of integrating multiple functionalities into large-area, low-cost sensor arrays is reached. Furthermore, the ability to mimic the mechanical properties of human skin (e.g., flexibility and stretchability) is critical.

One of the most important functions of skin is to facilitate the sense of touch, which includes normal force sensing for grip optimization, tensile strain sensing for proprioception, shear force sensing for object manipulation, and vibration sensing for slip detection and texture analysis. While the commonly used transduction methods (such as piezoresistive, capacitive, piezoelectric, optical, and wireless) are readily available, advancements in device structures and materials have produced dramatic improvements in tactile sensor performance.

Not only tremendous effort has been concentrated on developing e-skin hardware, but also has been done towards the system digitization, which requires software to provide a communication interface with the user. The signals produced from sensors in e-skins are analog, and signal digitization is required for interfacing with modern digital microprocessors. Ideally, digitization would be performed at the sensor device level in a manner that does not require considerable power consumption in order to integrate a large density of sensors. After digitization, software algorithms could

be used to process, interpret, and act on the large amounts of data in a coherent fashion. Further integration of other signal processing methods (such as wireless technologies) could be used to create super skins, such as implantable biofeedback devices for healthcare applications.

Several highly integrated e-skins demonstrating multiple functionalities for applications such as biomedical devices, robotics, and optoelectronics have been recently reported. The rapid pace of progress in e-skin technology suggests that the fabrication of a more complex e-skin with properties far surpassing those of their organic equivalent will soon be possible.

# Chapter 3

## Tactile Feedback System

### 3.1 Introduction

The results presented in this chapter are not published yet. The candidate contributed to the this work by designing algorithms for the control software of the tested system, developing of the experimental setups, planning the experimental protocols and pilot tests, and performing measurements. She also conducted a comprehensive statistical analysis of the collected data, gave all motivations, compared the work w.r.t. the state-of-the-art and provided discussion and conclusions.

Peerdeman et al. [57] developed a survey examining the requirements for feedback (and control). In hierarchical importance, the feedback priorities are:

1. Continuous and proportional feedback on grasping force should be provided;
2. Position feedback should be provided to user;
3. Interpretation of stimulation used for feedback should be easy and intuitive;
4. Feedback should be unobtrusive to user and others;
5. Feedback should be adjustable.

Several different methods can be employed to provide amputees with sensory feedback and they can be defined in two different ways. Sensory feedback systems distinguish invasive and non-invasive methods. The invasive methods elicits sensory feedback by interfacing directly to physiologically relevant neural structures in the peripheral nervous system (PNS) or the central nervous system (CNS); the non-invasive methods (e.g. using temperature, mechanical pressure, and augmented reality) provides

feedback to intact sensory systems (e.g. tactile stimuli on the residual limb, chest, etc.).

Alternatively, the sensory feedback systems could be divided into: somatotopically matched, modality-matched, and sensory substitution methods. Somatotopic matching refers to methods in which the feedback signal is perceived as being anatomically matched in location to where the stimulus is being applied to the prosthesis (i.e. invasive methods or sensory feedback applied to phantom mapping). Modality matching refers to a feedback signal that is congruent to the external stimulus detected by the prosthetic sensor; however, the feedback signal may not be presented to a location physiologically representative of the hand or limb. Sensory substitution categorizes a group of feedback systems that apply a feedback signal that is not matched in modality to the stimulus occurring at the prosthesis. Furthermore, the feedback signal is presented to a location that, physiologically, will not be perceived to the user as in the same corresponding location on their missing limb. The success of the approach depends on the users ability to interpret the type and location of the stimulus and associate it with the prosthesis. The most common methodology has been to translate tactile information from the prosthesis to the amputee using vibration, electrocutaneous or auditory stimuli.

A mind map of the different feedback methods is shown in Figure 3.1.

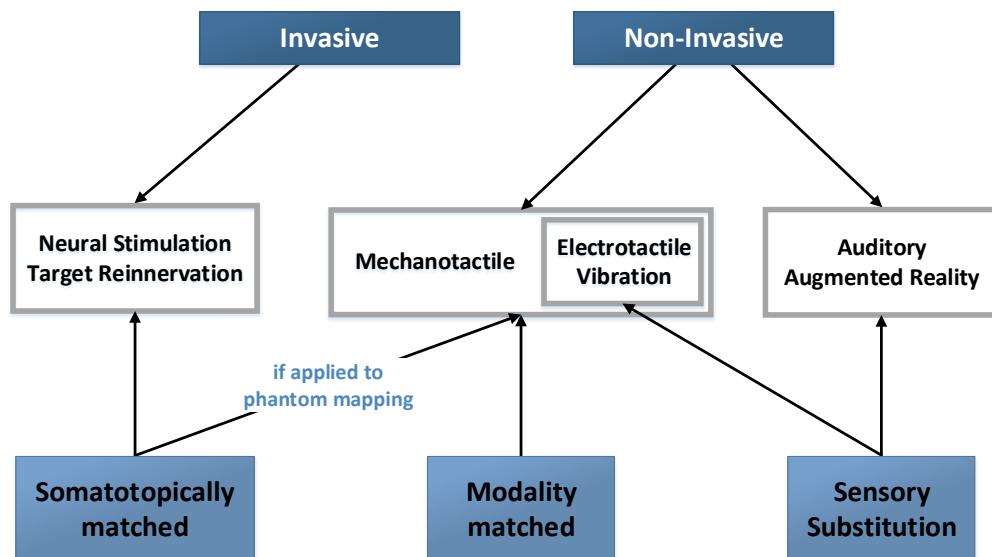


Figure 3.1: Classification of Sensory Feedback Methods.

Although first results with direct neural stimulation [58–61] were promising [59], more extensive studies are needed to understand how to safely stimulate afferent pathways of the human nervous system to provide effective sensory feedback. Besides, these approaches are invasive and require a surgical procedure, which may strongly affect their acceptance by the prosthesis users.

Non-invasive sensory feedback systems could prove to be an interesting alternative to invasive solutions. Implied assumption is that it might not be necessary for an artificial system to exactly restore the biological information transmission, provided that an intuitive communication between the prosthetic device and the human brain is established through a non-invasive interface

In addition to the sensory modality (what is fed back), another crucial aspect is the timing of sensory feedback (when is fed back). The latency between a variation of the output stimulus corresponding to a sensory input variation should be as short as possible for an effective use of such information. In the human sensory system, tactile stimuli take 14-28 ms to reach the cuneate nucleus [62]. Therefore it is reasonable to believe that to avoid increasing this value significantly, artificial sensory feedback should be delivered to the individual in a fraction of that time (e.g. 3-5 ms). Short latencies are also important for the brain to develop a sense of embodiment (body ownership) of the prosthesis. Indeed, the attribution of a visible hand to the self depends on a match between the afferent somatic signals and visual (and eventually audio) feedback from the hand [63]. Self-attribution occurs with temporal delays up to 300 ms, as reported by Shimada et al. in [64].

Johansson and Birznieks wrote that both friction between the object and the fingertips and the shape of grasped surfaces, recognized by tactile mechanisms, are reflected in the applied fingertip forces within  $\sim 100$  ms of initial contact (in [65]). Besides, they claimed that accidental slips and unexpected perturbations of a grasped object elicit responses in tactile afferent triggering specific behavioral consequences even faster ( $\sim 65$  ms). To conclude, peripheral nerve conduction times and muscular force generation delays account for  $\sim 45$  ms of the delay and at least 15 ms is required for central processing<sup>8 10</sup>.

The prevalence of stimulation, i.e. how continuous sensory feedback is to be provided is still a debating point within the field. Traditionally, researchers have implemented systems that presented the sensory feedback in a continuous fashion. However, continuous feedback yields to adaptation, meaning that the stimulation is no longer or just barely perceived by the individual after a short while. Considerations

regarding the adaptation phenomenon will be debated in paragraph “Open Issues” (Section 3.6).

In the Section 3.3 the principal features, advantages and disadvantages of some sensory feedback methodologies are presented.

## 3.2 Motivation

The incidence of upper limb amputees in European countries ranges from 50 to 270 per year [66], [67], with around 1900 traumatic upper limb amputees per year and a total of 94 000 upper limb amputees in the European community. [68] estimated there are 1.6 million people living in the USA with the loss of a limb, 34% of which were upper limb.

Upper limb loss is one of the most difficult challenges for prosthetic replacement, given the complexity of fine sensory input and dexterous function of a hand. Closely mimicking the performance of a human hand and arm is technically challenging. A normal hand is capable of coordinating movements with 27 degrees of freedom to perform strength-based grasping functioning as well as highly coordinated precision movements [69]. In the last years, there have been extensive advances in motorized, multi articulated prosthetic arms that are capable of a wide range of grasps and movement [70]. These devices are of three types: 1) cosmetic; 2) body powered; and 3) myoelectric. Body-powered prostheses are controlled by amputees body movements (e.g., shoulder shrugging) transferred through cable and/or harness systems to a terminal device such as hand or hook. These devices, although characterized by reliability, durability, cost, weight, and tension feedback to the body, suffer from the need for (gross) body movements, energy expenditure, and less cosmetic appearance than a myoelectrical prostheses. Myoelectric hands leading industrial developers are Ottobock (Germany), LTI (USA), and Motion Control (USA). Yet surveys [71] on the use of these artificial hands reveal that 30%-50% of amputees do not use their prosthetic hand regularly, basically due to its low functionality, poor cosmetic and unnatural appearance, lack of sensory feedback, and low controllability. Even though prostheses are becoming very sophisticated robotically, they are not satisfactory to users and are often abandoned by amputees as these devices do not allow amputees to perform tasks that are necessary for activities of daily living (ADLs).

Most of upper limb amputees discontinue usage of their prosthesis [72] as the inability to grasp/manipulate objects and the inability to sense and explore the surrounding is too disturbing. Typically, prosthetic users adopt a system of strategies



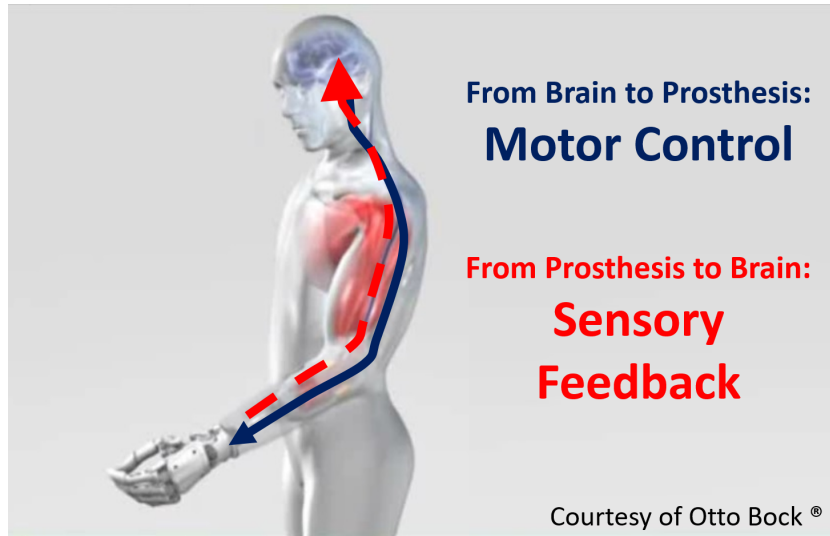


Figure 3.2: The restoration of sense of touch in prosthesis.

to compensate for this lack of sensory information; they rely heavily on visual feedback as well as on indirect feedback mechanisms [73]. Moreover, amputees experience phantom pain (i.e. pain that feels like it is coming from a body part that is no longer there) from the missing limb. Users do not perceive the prosthesis as a part of their own body but rather as an external device, experiencing a decrease of confidence in its use.

Restoring sensory feedback is a long-standing challenge in prosthetic research [74], [2] as contemporary myoelectric prostheses respond to electrical muscle activity and thereby restore lost motor functions, but the amputee users do not feel their artificial limbs. Apart from a single recent example [3], the commercial prostheses do not provide any somatosensory feedback to the user. It is known from the human motor control that a bilateral communication between the brain and the limb is essential for the smooth execution of the movements [75]. Therefore, it is even more important considering that sensory feedback is truly instrumental for the motor control in able-bodied subjects, especially during dexterous activities such as manipulation and grasping [76]. By using a prosthesis equipped with the sensory feedback-restoring device, the subject will be able to control the force exerted with the prosthesis and recognize the stiffness and shape of objects in a perceivable and effortless manner. The subject could have a decrease in phantom limb pain and, most importantly, perceive the prosthesis as a part of his own body. This could foster to improve the quality of life of amputees.

### 3.3 Transmission Methodologies Overview

There is a variety of feedback methods that have currently been deployed within literature including the use of temperature, vibration, mechanical pressure and skin stretching, electrotactile stimulation, audio feedback, and augmented reality. Some of these methods are described with an assessment and discussing any challenges and opportunities that are involved in each technique.

#### 3.3.1 Peripheral Nerve Stimulation

Authors of [4], [78] consider that the most obvious solution to simulate sensory feedback could be to electrically stimulate physiological channels as the nervous system functions on electrical voltage potentials. Peripheral nerve stimulation relies on the principle that, following upper limb amputation, the original afferent neural pathways are preserved proximally and can be exploited for interfacing with prostheses [79]. This principle suggests that natural physiological feedback can be restored through strategic electrical stimulation of nerve afferent using invasive neural electrodes. To date, peripheral nerve stimulation has been investigated in amputees using two styles of electrodes: nerve cuff electrodes, where the electrode wraps around the exterior of the fascicle [25]; and longitudinal intrafascicular electrodes, where electrodes are

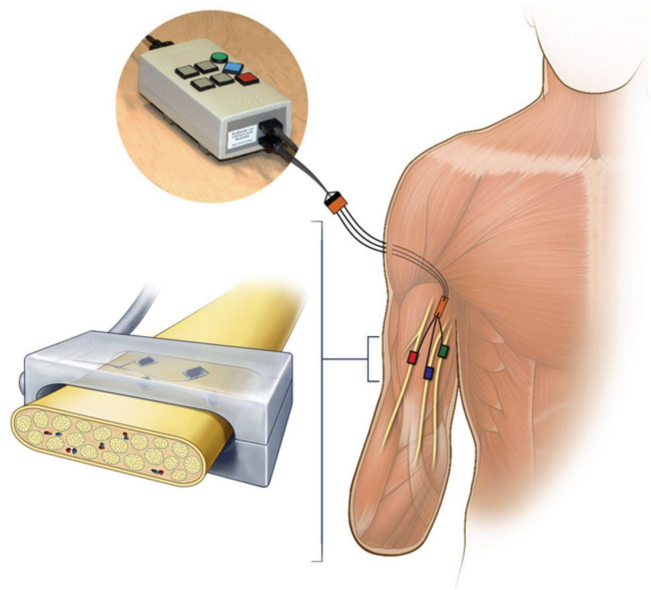


Figure 3.3: Peripheral nerve stimulation (source [77]).

placed in the nerves longitudinally [80–84]. As a technique for sensory feedback, peripheral nerve stimulation holds inherent technological limitations. Ultimately, the success of eliciting a particular sensation in a certain location is dependent on the systems ability to selectively stimulate specific sensory afferents in a particular fascicle. Current electrodes lack this selectivity and as a result, spatial resolutions of referred sensations are often large, encompassing entire fingers or areas as large as the palm [85], [86]. Beyond spatial discrimination, this lack of selectivity often results in a loss of naturalness in elicited sensations. Although participants do report tactile sensations, they are frequently accompanied by foreign sensations resembling vibration, tapping or fluttering on the skin [4], [85]. Furthermore, the long-term stability of intrafascicular electrode stimulation in human participants has yet to be comprehensively studied [79], [60]. Therefore, whether the body acclimates to the stimuli or if the system requires adjustments of the stimulus parameters over time remains unknown. For this method to move forward, the longevity, and ultimately feasibility, of using peripheral nerve stimulation in clinical or long-term prosthetic applications must be proven.

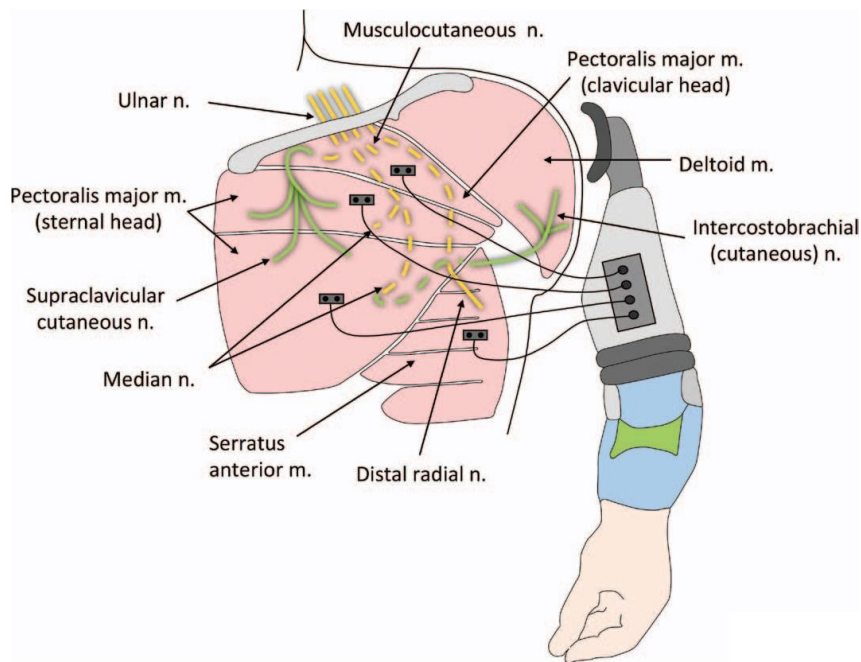


Figure 3.4: Targeted reinnervation (source [87]).

### 3.3.2 Targeted Reinnervation

Targeted Reinnervation (TR) is a surgical procedure that moves the motor and sensory nerves that previously innervated the amputated limb to muscle and skin target sites [88]. This surgery was initially performed to increase the number of motor control sites for myoelectric prostheses and to allow for intuitive control [88–90]. However, it was found that the redirected sensory afferents also reinnervate overlying skin. This reinnervation creates an expression of the hand map such that when touched, the patients feel as if they are being touched on the missing limb [91–93]. TR allows the reinnervated sites to be selectively placed [90]. also been shown to be repeatable and discrete.

Although TR feedback techniques are promising, research is still ongoing to develop means of effectively utilizing the reinnervated skin sites as a further limitation lies in the need for surgery to utilize this techniques.

### 3.3.3 Vibrotactile

Vibrational feedback typically uses small commercially available vibrators, which are applied to the skin surface and activate the Pacinian corpuscle mechanoreceptors in the skin. These are usually small and light weight. The user learns to associate the vibration at that site with one of the senses from their prosthetic hand.

The two main features of the stimulus are vibration amplitude and frequency, but other features like pulse duration, shape, and duty cycle, can be modulated to convey different kinds of information [95], [96].

Generally incorporating vibration feedback reduces cognitive load required to pick up objects compared to using visual feedback alone, however, this was not consistent



Figure 3.5: Examples of vibrators used in vibrotactile feedback (source [94]).

across all the subjects. Vibrational feedback requires users to undergo training in order to develop the full benefit [97].

Despite the fact that vibrational feedback offers a cheap and lightweight system of feedback that users prefer the sensation over electrotactile feedback [98], one limitation is the delay in stimulation as it can decrease embodiment [99], [64]. Recent studies are working on this aspect. Another limitations is that the perception of vibrational frequency can be affected by how tightly a vibration motor is attached [100], which raises difficulties in predictive and reliable sensory feedback.

### 3.3.4 Electrotactile

Electrotactile (or electrocutaneous) feedback communicates sensory information to the prosthetic user via electrodes placed on the users skin. Electrotactile stimulation evokes sensations within the skin by stimulating afferent nerve endings through a local electrical current. The sensation is not necessarily confined to the zone under the electrodes but elicited sensations could spread if these are placed near nerve bundles [101].

Electrotactile stimulators can be designed to be either current- or voltage-regulated. With current-regulated stimulation, the current is not affected by changes in the tissue load and impedance at the electrode interface; on the other hand, voltage-regulated stimulation minimizes the possibility of skin burns owing to high current densities. Current amplitude, pulse waveform (biphasic/monophasic, rectangular/sinusoidal), pulse frequency and duration, duration of pulse bursts, and location of stimulation (with multiple electrodes), which leads to a higher bandwidth being available [102], are the principal features of the stimulation. Electrocutaneous stimulation can evoke a



Figure 3.6: Examples of electrotactile electrodes (source [94]).

range of sensations that have been qualitatively described by participants as a tingling, itch, vibration, buzz, touch, pressure, pinch and sharp or burning pain [101] depending on the stimulating voltage, current and waveform, as well as on the electrode size, material and contact force, and the skin location, hydration, and thickness. Electrodes with a small area are required when only a limited area is available, even though larger electrodes provide a more comfortable sensation [101]. Hence, a trade-off between electrode size and skin area should be identified when performing multi-site stimulation.

A few studies have demonstrated the benefit of using electrotactile feedback, such as [103]. Strbac et al. [104] demonstrated a different electrode design that enabled users to distinguish up to 16 stimulation locations, with up to five different frequencies at once, to provide multiple levels of feedback. Test results from a small number of able-bodied and amputee subjects demonstrated that six electrodes with four different frequency signals could be identified with more than 90% accuracy by the subjects after minimal training.

Electrotactile feedback shows potential for a quick and easily controllable method of feedback that users can identify multiple sites of feedback at once. Typically electrotactile devices consume less power and respond faster than vibrotactile systems as there are no moving mechanical parts. However, in order to perceive electrotactile stimulation each person's minimum sensation threshold and pain threshold is different and the perception of electrotactile information changes with the placement of the electrodes [105], with movements as small as 1 mm having an influence [101]. In addition, skin conditions can also influence the comfort and dynamic range of electrotactile stimulation [101]. Not only does this mean that re-calibration of thresholds are required every time electrodes are placed on the user; but that the pulse width, frequency and amplitude may need readjusting to achieve the same perception each time. Elicited sensations are too dependent on many stimulation parameters as well as physiological factors [101]. Therefore, the ability to repeatedly isolate and elicit a specific sensation becomes an involved task. In a prosthetic context, sensory feedback devices should have long-term stability and consistency of the prosthetic-to-user communication channel. Without stability in the elicited sensations, the user may face substantial challenges in learning to interpret feedback.

### 3.3.5 Mechanotactile

Mechanotactile feedback is accomplished when a force normal to the skin is applied by a pusher to convey sensory information. The main features of this class of stimulators

are accuracy (how accurately the output force/pressure resembles the sensory input), precision, range, resolution and bandwidth (or its inverse concept, i.e. response time).

In amputee studies, incorporation of mechanotactile feedback has been shown to improve performance during object manipulation tasks [106]. Additionally, as a feedback signal, mechanotactile tactors can provide graduated levels of force (or pressure) and typically enable the user to discriminate between various levels [107], [108]. Casini et al. [109] demonstrated the application of distributed haptic force to help a user determine an object as hard, medium or soft. A combination of pressure and skin stretch on the bicep was used as the feedback mechanism for the subject.

Compared with other feedback systems, mechanotactile systems typically consume more power and are often larger and heavier than vibrotactile or electrotactile devices; development into minimizing these drawbacks is ongoing [110], [111].

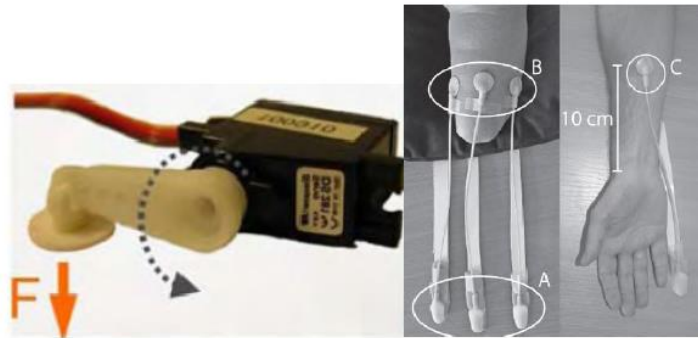


Figure 3.7: Mechanical pressure feedback device and Silicon bulb mechanical feedback (source [94]).



Figure 3.8: Pressure feedback cuff (source [94]).

### **3.3.6 Temperature**

Temperature feedback has only been deployed to communicate identify force of their grip and the position of their fingers [57]. However, temperature provides users with extra information about their environment, and potential dangers or warnings that involve heat. Producing heat on the upper arm to correspond with temperature detected at the prosthetic hand was the only method of temperature feedback found within literature.

A potential focus of research would be to incorporate temperature feedback with another feedback method so that they occur simultaneously, since it is not a priority to occur by itself.

### **3.3.7 Audio**

Wilson and Dirven [112] demonstrated the potential of deploying audio to communicate sensory feedback from a prosthesis. They examined the test subject's ability to interpret modulation of two audio channels to control a computer simulation. Their data showed that the subject could interpret two channels, but there was a 602 ms delay and the audio feedback resulted in a high cognitive load. Other studies [113] utilised triads to communicate the movement of a robotic hand. The sound of cello corresponded to the force on the thumb and a piano sound represented the force on index finger. The subjects were also able to use the audio feedback to help improve their movements and control when grasping objects.

Each of these audio feedback experiments was given their high cognitive load required, further investigation is required to determine their effectiveness whilst background noise is occurring.

### **3.3.8 Augmented Reality**

Markovic et al. [114] used Google glasses to communicate the aperture angle, contact time, grasping force and EMG strength for sensory feedback of a prosthetic hand to its user. Subjects used the visual feedback to improve their task performance when moving objects that required various strengths without breaking them. The subjects noted, however, that they typically only glanced at the information and did not use EMG strength signals.

Moreover, [115] suggests that although performance repeatability can be increased with augmented feedback, it increases the cognitive load required from the user.



### 3.4 Electrocutaneous Stimulation System

Traditionally, the non-invasive sensory feedback systems rely on a few discrete sensing and stimulation units [4], [94]. In a typical approach, a sensor is used to measure a global prosthesis variable (e.g., overall grasping force), and this information is then transmitted to the prosthesis user through a single stimulation unit, which can be an electrode placed on the residual limb [94], [102]. The intensity and/or frequency of stimulation can be modulated according to the measured variable. For example, the higher the grasping force, the higher is the stimulation intensity delivered to the subject, which leads to a stronger tactile sensation [101]. The subject needs to learn to associate the elicited sensation to the measured variable. This can be a challenging task, and typically, only few levels of grasping force/hand aperture can be reliably communicated [59, 86, 116, 117].

The contemporary methods for feedback restoration are characterized by a limited information transfer. To mitigate this drawback, feedback interfaces comprising several stimulation units were presented. In principle, the multichannel stimulation could allow to better exploit the inherent potential of the human skin as the feedback stimulation can be distributed over a larger skin area (spatial coding). An array of vibration motors was used to communicate hand aperture [98] and grasping force [118] feedback. Advanced interfaces for electrotactile stimulation integrating multiple electrode pads have been recently tested [104]. In [119], an electrode array integrating 16 pads placed circumferentially around the forearm was used to deliver force feedback from a Michelangelo prosthesis. A matrix electrode with 4 x 8 pads has been used to transmit tactile data recorded by an electronic skin to the subject forearm [120]. The tactile data recorded by four neighboring taxels were fused and delivered through a spatially-congruent electrode pad. The spatial coding and artificial skin were used in another experiment [121] where the stimulation was delivered through a 4 x 2 arrangement of conventional self-adhesive concentric electrodes. Preliminary experiments on the ability to localize touch delivered to the artificial skin by identifying the elicited electrotactile sensation have been performed on two subjects. These studies have shown the potential of the multichannel stimulation, however, they also pointed out that spatial localization is a challenging task, especially considering a low density of tactile receptors on the human forearm (stimulation target in prosthetics).

The focus here is not on what information to transmit to the prosthesis user, but the aim is to increase the reliability of information transmission when using non-invasive

interfaces with many stimulation points. More specifically, a novel electrotactile interface that has not been tested before is used.

The studies in the literature investigating multichannel stimulation rely on linear arrays (e.g., rows or columns of stimulation points) [98, 104, 118], whereas a dense 6x4 electrode matrix is proposed. An electrode matrix is particularly suitable for transmitting tactile information from an artificial skin covering the prosthetic device, as demonstrated in our previous study [120]. Arranging the electrodes in a matrix is useful for applications because it leads to a compact interface, however, this can be a challenge for spatial localization of the tactile stimulus. The spatial localization is important for transmitting an accurate tactile information to the user of a prosthesis (e.g., contact location) or when conveying other prosthesis variables using spatial coding (e.g., aperture [98]). This provides the motivation for the present study, which investigates if the quality of localization can be improved by exploiting the flexibility in parameter modulation provided by an electrotactile interface.

The electrotactile stimulation allows independent modulation of intensity and frequency, while these parameters are intrinsically coupled in commonly used vibration motors [122]. A novel dual-parameter modulation (intensity and frequency) has been developed and tested to assist the subject in correctly identifying an active pad within the matrix. Furthermore, for the first time, the performance of a matrix electrotactile interface is compared to that of the natural skin being mechanically stimulated over analogous contact areas. The detailed setup for the multi-channel electrotactile feedback system is illustrated in Figure 3.9.

The MaxSens stimulator prototype developed by Tecnalia Research and Innovation generates electrostimulation profiles to be transmitted to the participant through a matrix of electrodes applied to his/her forearm. The system was presented previously and tested in the context of transmitting multimodal proprioceptive [104] and high-resolution force feedback [119], facilitating the closed-loop control of a myoelectric prosthesis. The MaxSens is actually the next generation of the IntFES stimulation system [123], adjusted for electrotactile feedback applications (lower stimulation amplitudes with higher resolution).

The stimulator comprises a single stimulation unit that can generate continuous, biphasic, symmetric and rectangular current pulses [104]. The biphasic waveform has been preferred to monophasic pulses due to less skin reddening and more comfortable sensations [102]. The waveform is presented in Figure 3.10, with all the relevant stimulation variables marked on the plot. The stimulation unit is connected to a matrix electrode through an analog multiplexer, which can distribute the pulses in

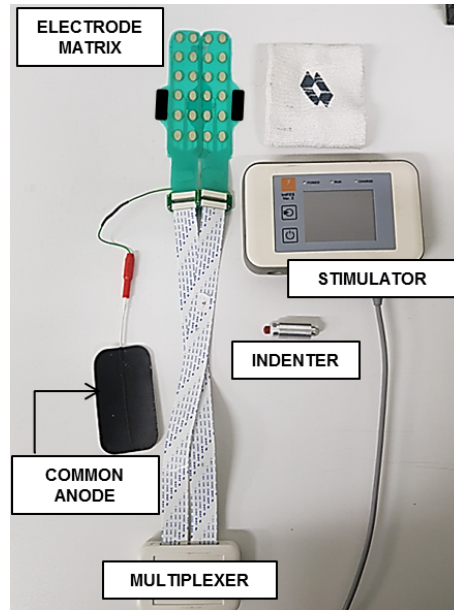


Figure 3.9: Experimental setup. The indenter is used for mechanical stimulation, while all other elements are used for electrotactile stimulation experiments.

time and space over the matrix electrode, thus providing multichannel stimulation interface. The MaxSens is fully programmable and the stimulation parameters can be adjusted online by sending text commands from the host PC via Bluetooth connection. The communication protocol is supported by several programming languages (C, C#, JAVA, LabView, MATLAB), enabling custom-made control software to be developed for a preferred platform. Importantly, stimulation parameters regulate the intensity and frequency of electrostimulation as well as the location (by changing the active electrode). The adjustable stimulation parameters are: (1) the current amplitude that can be changed in the range 0-5 mA with increments of 0.1 mA, (2) the pulse width (from 50 to 1000 s with increments of 10 s), and (3) the stimulus frequency (1 - 400 Hz, 1Hz step). Current regulated stimulation has been chosen to minimize the influence of capacitive effects at the electrode-skin interface and ensure consistent activation of the cutaneous nerves [124]. This guarantees that the nerve stimulation and thereby elicited tactile sensation will not be altered throughout the experiment due to changes in skin moisture and hydrogel adhesion.

In this study, the stimulator was connected to two custom designed flexible matrix electrodes developed by Tecnalia Research and Innovation. Each matrix electrode consists of 16 oval units (=pads) with the longitudinal radius of 5 mm and transverse radius of 3 mm. They are arranged in 6 x 2 geometry, plus 4 lateral pads (two at

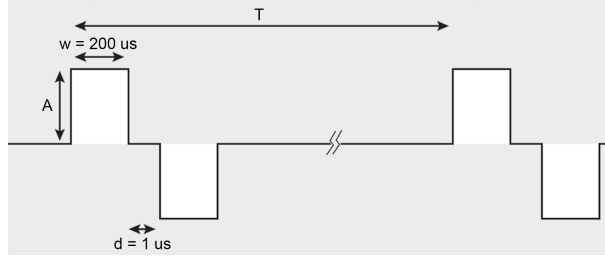


Figure 3.10: Typical stimulation waveform. Notation:  $A$  pulse amplitude;  $w$  pulse width;  $d$  inter-pulse delay;  $T$  inter-pulse interval (pulse rate = frequency =  $1/T$ ).

each side). The center-to-center distance between two adjacent pads is 20 mm in the longitudinal and 14 mm in the transverse directions. Each pad is made of Ag/AgCl conductive layer and conductive hydrogel circular elements of 5 mm radius (AG730, Axelgaard, DK) are added on top of each pad to assure good electrical contact between the pad and the skin. An insulation coating is distributed on top of the electrode, excluding the pad areas. The pads on the matrix were designated as cathodes whereas a single self-adhesive electrode (ValuTrode Foam [125]) placed on the dorsal side of the forearm acted as the common anode (Figure 3.9). The ValuTrode bottom electrode is made of glycerin, water and poly(acrylate) co-polymer. It is a well-known product on the market, recognized for its durability and multiple applications to the skin. The rectangular ValuTrode electrode with size 5 x 9 cm<sup>2</sup> was used.

For the experiment, the two flexible matrix electrodes have been overlapped in their central part in order to obtain a rectangular array including 6 x 4 pads (Figure 3.9), distributed over a total area of 11 x 5 cm<sup>2</sup>. Hereafter, this rectangular array will be called the matrix electrode. The 4 lateral pads (two at each side) were not used in the present study. In the overlapped zone, the four pads of the matrix electrode on top have been deactivated. The matrix electrode was placed on the volar side of the subject forearm, while the common electrode was positioned on the dorsal side.

The rationale for the design of the matrix electrode is a trade-off between the number of pads, pad spacing and electrode size to transversally cover the volar aspect of the forearm (Figure 3.9), which is good point of application for electrotactile feedback due to high tactile sensitivity and acuity.

The electrotactile interface has been designed so that the spacing between the pads is higher than the spatial discrimination threshold on the forearm [126] while the number of pads is still enough for a flexible mapping of prospective feedback variables. Moreover, the specific design of the multichannel interface enables an intuitive spatial

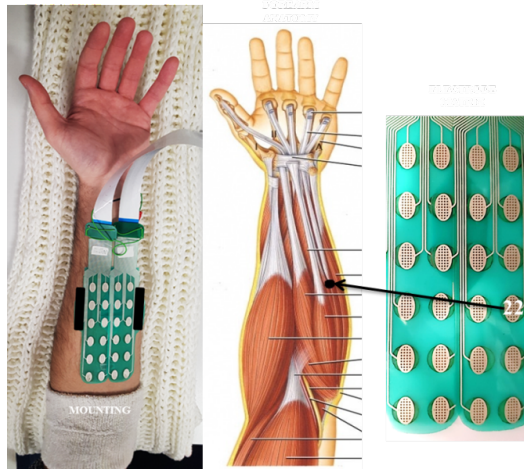


Figure 3.11: Reference position for matrix electrode placing is indicated by a black dot: it corresponds to the intersection between two superficial flexors, i.e. the palmaris longus and the flexor carpi ulnaris muscles. Reference pad for that position is number 22.

mapping between contact on the prosthesis and stimulation: the four columns can be mapped to the four fingers (4 x 5) and the thumb can be represented on the remaining row (4 x 1). In this configuration, the phalanges are mapped into the different (bottom, middle, top) subsections of the columns / row.

The overlapping electrodes were placed on the volar side of the subject forearm (Figure 3.11). Particular care has been taken for electrode placement and the same procedure has been used for all the tests, to make results comparable for different participants and for same participant over different trials. Columns were aligned with the four fingers and a reference point for electrode positioning on the longitudinal direction was associated to the intersection of two specific muscles, as indicated in Figure 3.11. In particular, the intersection between two superficial flexors, i.e. the palmaris longus and the flexor carpi ulnaris muscles has been used as the reference position for a specific pad (number 22). The indicated position was identified by asking the participant to close the hand and contract the muscles of the forearm. The electrodes were wrapped by a bandage to prevent movement and improve contact.

### 3.5 Proposed Methods

The present study assessed the subject ability to localize the electrical stimulus delivered through a compact matrix electrode with many pads. The results demonstrated that conventionally applied electrotactile stimulation (single frequency) can reach

similar performance in tactile acuity as natural mechanical stimulation. With a novel dual-parameter modulation scheme, the electrotactile interface provided higher discriminability than the mechanical stimulation. This is an important outcome for the provision of sensory feedback in prosthetics, as it implies that an electrotactile matrix interface can be used to transmit reliable high-fidelity feedback from the prosthesis, by exploiting the flexibility in spatial and parameter modulation characteristic of electrotactile stimulation.

A novel non-invasive interface for multichannel electrotactile feedback is presented, comprising a matrix of 24 pads (6 x 4), and the ability of able-bodied human subjects to localize the electrotactile stimulus delivered through the matrix is investigated. More specifically, a conventional stimulation (same frequency for all pads) and a specific dual-parameter modulation scheme (interleaved frequency and intensity) designed to facilitate the spatial localization over the electrode were tested. The mechanical stimulation of the same locations on the skin was used as the benchmark. An experimental campaign on eight able-bodied subjects demonstrated that the proposed interleaved coding of the electrostimulation signal substantially improved the spatial localization compared to the same-frequency stimulation. The results also showed that the same-frequency stimulation was equivalent to the mechanical stimulation, whereas the performance with the dual-parameter modulation was significantly better. These are encouraging outcomes for the application of multichannel interface with many stimulation pads for the restoration of feedback in prosthetics. The high-resolution augmented interfaces might be used to explore novel scenarios for intuitive communication with the prosthesis user enabled by maximizing information transmission.

### 3.5.1 Experimental Procedure

The main experimental campaign involved eight able-bodied volunteers (2 females, 6 males,  $35 \pm 8$  years). The experimental study has been approved by the Regione Liguria Ethical Committee (approval ID 172REG2016, approval date September 13, 2016). An informed consent form was signed by each participant prior to the experiments.

The subject sat comfortably on a chair in front of a table. The forearm of the non-dominant arm (always left, as all participants were right-handed) was placed on the table surface, with the volar side oriented upwards. The skin was preliminary prepared by moisturizing with a water-soaked cotton cloth to enhance the attachment of the electrodes and improve electrical conductivity.

The main aim of the experimental procedure was to assess the subject ability in identifying the location of the electrotactile stimulus delivered through the matrix

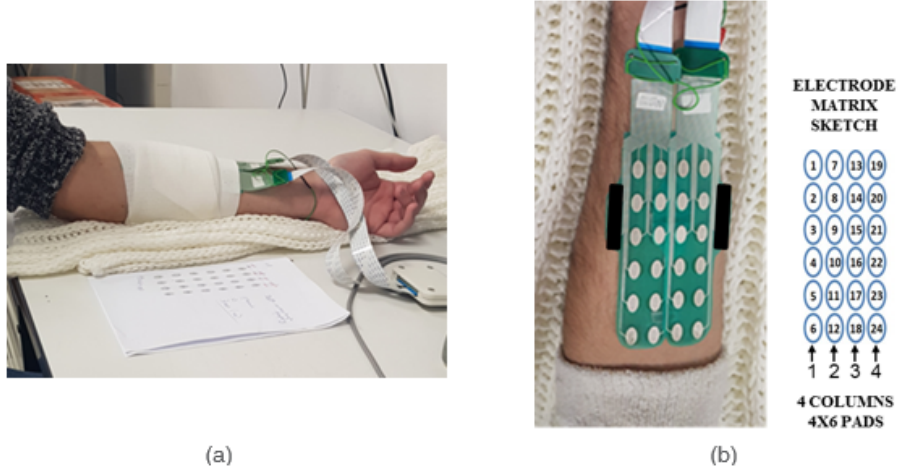


Figure 3.12: (a) Electrostimulation tests: a sketch of the matrix electrode (4 columns, 6 rows) is placed on the table next to the forearm. (b) Spatial correspondence between the matrix electrode and the geometrical arrangement of the pads in the sketch is preserved.

electrode. This was compared to the quality of localization when the mechanical stimulation was applied to the skin (benchmark condition). To foster attention and concentration, a silent environment was chosen to avoid any distraction for the participant. To maintain alertness and minimize adaptation, the subjects were always given sufficient rest during experiments [127], [128].

The stimulation software for managing experiments was developed in Matlab 2015b.

The experiment was divided in two sessions performed in consecutive days. The electrical stimulation (Figure 3.12) was tested using two coding schemes: 1) a conventional approach with uniform frequency (50 Hz) for all the pads (hereafter denoted as same-frequency condition) and 2) a dual-parameter modulation of intensity and frequency, interleaved across the electrode columns (hereafter denoted as interleaved stimulation). The main idea of the latter approach was that additional cues (parameter modulation) would assist the subject in recognizing the location of the active pad. In this scheme, the pads within the columns 1-4 were activated at the frequencies of 10, 400, 10 and 400 Hz, respectively. In addition, the stimulation at 10 Hz was delivered at a lower intensity compared to 400 Hz. The specific frequencies and intensities were determined through pilot tests. The interleaved stimulation was tested in the first session, and conventional electrical and mechanical stimulation in the second session. The session duration was low enough (i.e.  $\sim 1$ -1.5 hours) to avoid fatigue and distraction of the participant.

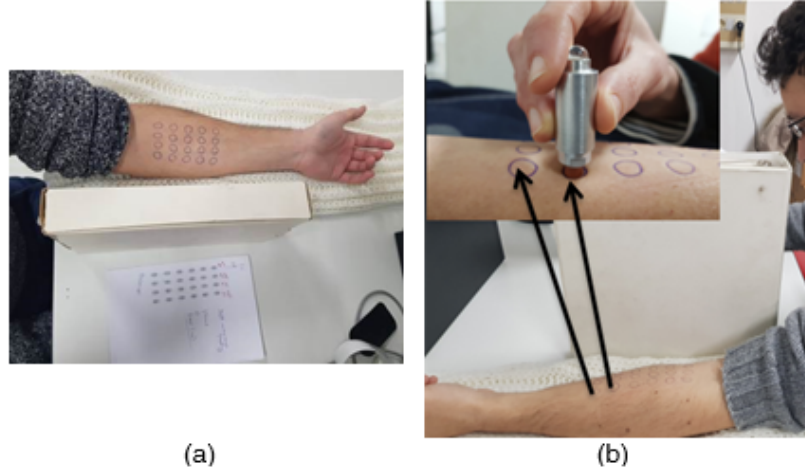


Figure 3.13: Experiments with mechanical stimulation. A screen is placed between the participants forearm and the sketch of the matrix electrode. (b) Top view. (b) Side view.

In the second session, the electrical stimulation was tested first and the mechanical was tested next (Fig. 5). When electrodes were removed, the position of the pads was still clearly visible via skin reddening. The experimenter therefore marked the position of each pad by using skin-friendly marker. A rubber indenter (radius 4mm, contact area of approximately 8-10mm diameter  $\sim$  same size of the pad) was used to mechanically stimulate the skin of the participant (inset in Figure 3.13b).

Each stimulation modality (interleaved, same-frequency and mechanical) was tested using the same experimental protocol comprising three phases: intensity adjustment, training and testing.

**Intensity adjustment.** For electrotactile conditions, the participant was first asked to define the intensity of the stimulation of each pad for clear perception, avoiding any discomfort. For that, the stimulation intensity was increased gradually (0.1mA each step) and the participant was asked to stop the experimenter when clear sensation was achieved. The pads were activated in a systematic way, column by column. Inside each column, adjacent pads were not activated sequentially: a specific order was chosen to minimize adaptation (same order for all columns, referring to column 1 in Figure 3.12b: 1-3-6-4-2-5). Whenever numb feelings were reported, strongly affecting the possibility of localizing the stimulus as the sensation spread over to the whole forearm, the electrode array position was slightly adjusted.

In the case of interleaved stimulation, the experimenter additionally needed to adjust the intensities for the 10 Hz and 400 Hz stimulation. To set the low stimulation intensity for 10 Hz, the subject was asked to look for low but clear sensation. These



values (level 1) were commonly associated to 1-2 steps above the sensation threshold. To set the higher stimulation intensity for 400 Hz, the subject was asked to look for high but not painful sensation, typically stopping 2-3 steps above level 1. After setting the 2 intensity values for the two frequencies, the experimenter let the participant experience both sensations moving all over the columns, and small adjustments were allowed.

The intensity of the mechanical stimulation was preliminarily tuned for the stimulation to be clearly perceived by the participant. In any case, preliminary studies showed that there was no relevant difference in pad recognition for different intensities of the mechanical stimulation.

**Training.** After defining stimulation intensities, a training session was performed for the participant to familiarize with electrostimulation and build a mental mapping between the experienced sensation and the position of the stimulated pad and/or the pad number. For that, a sketch of the matrix electrode including the real-size 24 numbered active pads was placed on the table adjoining the forearm, preserving spatial correspondence with the matrix electrode (Figure 3.12).

The subject was first trained by experiencing sequential stimulation over each column from top (wrist) to bottom (elbow), while the experimenter orally reported the pad number. Here the participant knew in advance which column and pad would be stimulated and he/she was expected to associate felt sensation to the pad location (number).

In the second training stage, the column was known, but the pads within the column were stimulated in random order. The participant guessed the pad number and then the experimenter provided verbal feedback about the correct answer (reinforced learning).

**Testing.** During testing, the task for the subject was to identify the active pad, and no feedback was provided about the correctness of the guess. The single pads were activated in a pseudo-random order so that each pad was presented two times (48 stimulations). The participant was asked to identify the activated pad, by indicating its number or identifying its position over the sketch. In few cases, the subject could not decide on the location, and this was registered as a missed sensation.

In the training and testing phase during electrical stimulation, the participant was allowed to freely direct the look from the forearm to the sketch and back (Figure 3.12a). During mechanical stimulation, a screen was placed between the participants forearm and the sketch of the matrix electrode to prevent the participant having visual cues to identify the stimulation location (Figure 3.13). In all modalities, the duration

of the stimulus delivered to the subject was 2 s. In both conventional and interleaved stimulation modalities, pulse width and inter-pulse delay were set to  $w = 200$  s and  $d = 1$  s, respectively.

### 3.5.2 Data Analysis

The main outcome measure was the success rate (SR) in locating the stimulus. This included the identification of the exact pad at which the stimulation was delivered. However, our intended application is in prosthetics, where small errors can be often tolerated. Therefore, the SR was computed also for pointing to the first neighbor around the correct pad (one-position error) and to the pad within the same column as the correct pad (correct column). The latter (correct column) is of interest when mapping prosthesis variables to the electrode pads, since mistaking the column could represent a much larger error.

The SRs were computed per subject for each specific stimulation modality (mechanical, same frequency, and interleaved stimulation). The SRs of all subjects were then averaged to obtain the overall mean SR and its standard deviation. The results were reported as mean  $\pm$  standard deviation in the text and figures.

The data were tested for normality using Kolmogorov-Smirnov test. In all cases, the tests indicated normal distributions, and therefore one-way repeated measure ANOVA was used to assess statistically significant differences at the level of the group followed by Tukeys honestly significant difference test for post hoc pairwise comparison. One-way ANOVA tests were used to compare the success rates in recognizing the specific pad or column across stimulation modalities. The threshold for the statistical significance was adopted at  $p < 0.05$ , and the statistical analysis was conducted in Matlab R2015b (MathWorks, US).

### 3.5.3 Results and Discussion

Figure 3.14 shows the performance for individual subjects across stimulation modalities. The bars represent the SR in (i) correctly identifying the right pad (light blue), (ii) wrongly identifying the pad but pointing to the right column (orange), (iii) wrongly identifying the pad and the column (grey). The variability across subjects is noticeable for mechanical stimulation.

The summary results, i.e. overall SR, is shown in Figure 3.15. In general, pad recognition was not an easy task for the subjects (Figure 3.15a). The overall SR for the mechanical stimulation was  $17 \pm 9\%$ . The electrotactile stimulation using the same

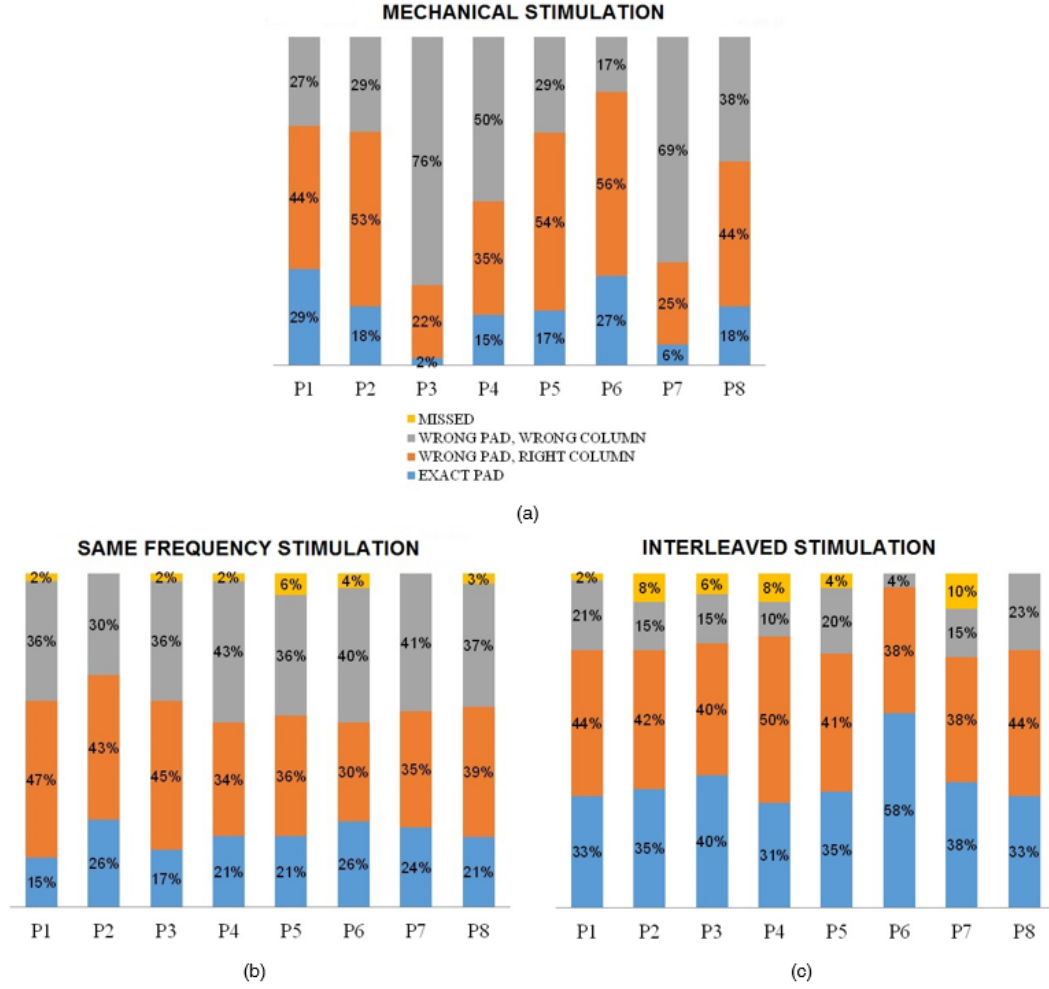


Figure 3.14: The results for individual subjects (P1-P8). Reported percentages are associated to identifying the right pad (light blue), missing the pad but addressing the right column (orange), missing the pad and the column (grey), no answer (yellow).

frequency for all pads (50 Hz) was characterized with the similarly low SR ( $21 \pm 4\%$ ). Therefore, the same-frequency electro tactile stimulation provides comparable quality of spatial localization to that of the mechanical stimulation. However, with both modalities the performance was still substantially better than pure chance, where the subject would simply randomly select one of the pads ( $1/24 \sim 4\%$ ). Importantly, the SR for the electro tactile stimulation that used the interleaved frequencies and intensities was significantly better ( $38 \pm 9\%$ ) compared to both mechanical ( $p < 0.001$ ) and the same-frequency electrical stimulation ( $p < 0.001$ ). The performance almost doubled with the interleaved stimulation scheme. Therefore, the dual-parameter modulation substantially improved the subjects ability to correctly localize the elicited tactile sensation.

The summary performance in localizing the stimulus up to an error margin around the active pad are reported in Figure 3.15b-d. Figure 3.15b gives percent of trials in which the subject pointed to a correct pad or its immediate neighbor within the same column (one-position, within-column error tolerance). Figure 3.15c is a percent of trials in which the subject pointed to a correct pad or any other pad that belonged to the same column (within-column error tolerance). Again, the SRs in the case of one-position error (Figure 3.15b) for interleaved stimulation ( $70\pm11\%$ ) was statistically significantly higher than for the same-frequency electrical ( $42\pm6\%$ ,  $p < 0.01$ ) and mechanical stimulation ( $45\pm20\%$ ,  $p < 0.01$ ). If a small localization error can be tolerated, the interleaved stimulation can therefore lead to a very good performance (e.g. SR up to 96% for subject P6). More generally, with the interleaved stimulation, the subjects could reliably detect the right column (Figure 3.15c). The success rate for this modality was significantly better ( $80\pm7\%$ ) than for the same-frequency ( $60\pm5\%$ ,  $p < 0.01$ ) and the mechanical stimulation ( $59\pm21\%$ ,  $p < 0.01$ ). Finally, Figure 3.15d reports for all modalities the percent of trials in which the subjects pointed to a correct pad or its immediate neighbor, regardless of the column. This figure further emphasizes the equivalence of mechanical stimulation (SRs:  $63\pm21\%$ ) and same-frequency electrostimulation (SRs:  $64\pm9\%$ ). The interleaved coding leads again to a higher average SR ( $\sim 79\pm8\%$ ), though this time there was no statistically significant difference with the other two modalities.

The overall success rates for the recognition of individual pads of the matrix electrode in each of the stimulation modalities are shown in Figure 3.16b (mechanical), c (same-frequency), and d (interleaved stimulation). The figure once again demonstrates that the interleaved modality is the technique which allows for the best recognition of single pads. With mechanical and same-frequency stimulation, there is a trend that the pads on the borders of the electrode area are more successfully recognized compared to the inner pads. In the case of interleaved stimulation, the SR increases for most of the pads and some inner pads reach comparably high SRs.

It has been investigated if the modulation of additional parameters (frequency and intensity) can improve the spatial localization of the electrotactile stimuli. The (natural) mechanical stimulation of the skin was adopted as the benchmark.

The first important conclusion of the study is that the electrotactile stimulation delivered conventionally, using the same frequency for all the pads, resulted in a similar performance as the mechanical stimulation. The electrotactile stimulation is non-specific and activates a combination of mechanotactile receptors. In addition, the electrical current spreads in the tissue, especially in this configuration where the

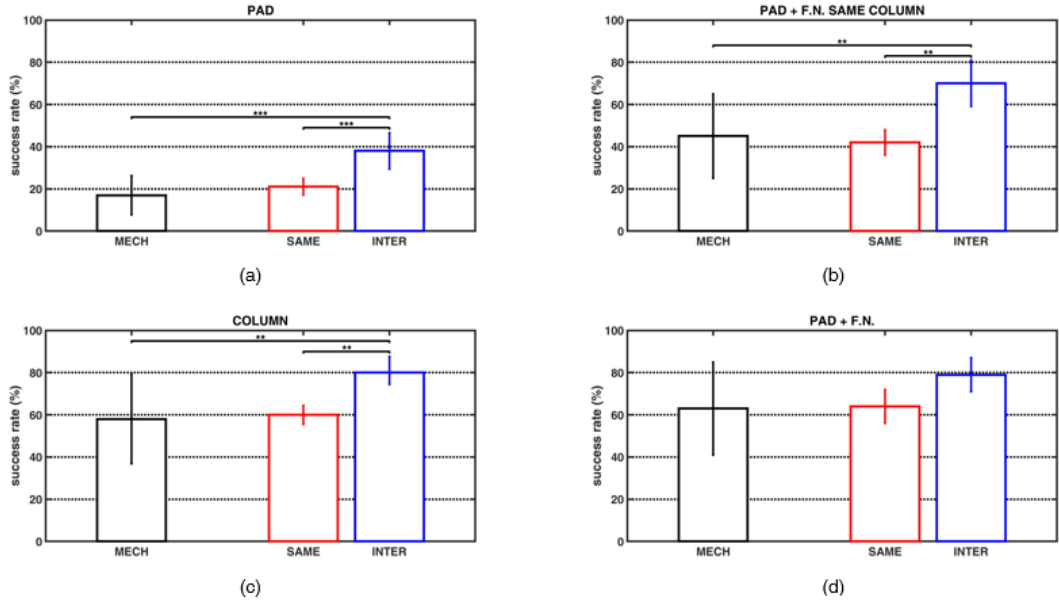


Figure 3.15: The summary results for all subjects. The bars show the success rates (mean standard deviation) in identifying the right pad (a), pointing to the right pad or first neighbors (F.N.) within the same column (b), pointing to the right pad or any pad belonging to the same column (c) and pointing to the right pad or any of its first neighbors, regardless of the column (d).

common electrode is positioned outside of the integrated matrix. The fact that the electrical and mechanical stimulation performed similarly is an encouraging outcome for the application of multichannel interfaces with significant number of pads to the feedback restoration in prosthetics.

The second important conclusion is that the electrical stimulation has an intrinsic potential, namely, the flexibility in parameter modulation, which can be used to increase the reliability of information transmission to the subject. The present experiment has demonstrated how dual-parameter modulation can be used to substantially improve the performance in spatial localization of the elicited tactile sensation. The subjects were far more successful in identifying the stimulation location using the interleaved stimulation modality compared to other modalities in those analyses related to addressing the correct pad (Figure 3.15a), or when accepting (small or large) errors within the right column (Figure 3.15b and Figure 3.15c). The interleaved stimulation is therefore advantageous for spatial localization, and in addition, the performance increased substantially above the inherent potential of the human skin, as tested by stimulating the skin mechanically.

This leads to an interesting possibility in prosthetics, where an electrotactile interface can be used to equip a prosthesis user with an artificial tactile sense that

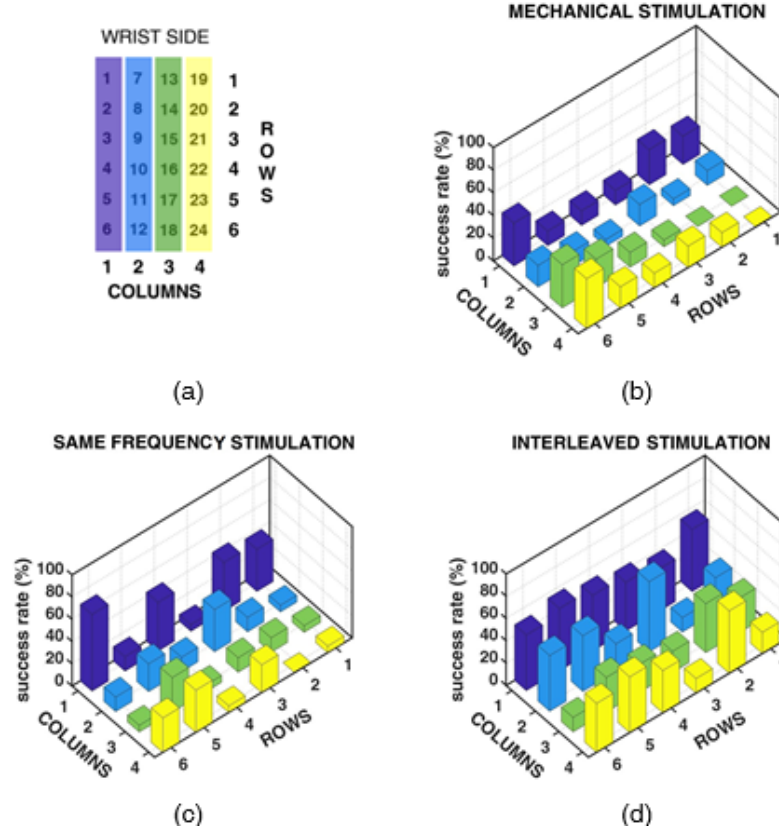


Figure 3.16: Success rates for the identification of each pad. The scheme reported in (a) illustrates the orientation of the matrix electrode with respect to the forearm.

can overcome some limitations of the real skin (e.g., low-density of receptors and thereby poor spatial localization over the forearm). Furthermore, this is a unique advantage of electrical stimulation, as this is the modality where the parameters are truly independent. In vibration motors, for example, the parameters are often mechanically coupled [127], [129] and in modality matched stimulation there is often only one parameter to modulate (e.g., the pushing force [106]).

The matrix electrodes as well as simultaneous modulation of stimulation parameters have been used previously but not in the same context. [120] demonstrated that the subjects were able to recognize the dynamic stimulus moving along the matrix electrode, i.e., different shapes (geometries and letters) were presented using conventional stimulation with a fixed frequency (corresponding to same-frequency condition of the present study). [119] used simultaneous modulation of location and frequency (mixed coding) to communicate clearly a number of levels of prosthesis grasping force using an array electrode (effectively only 5 pads). In the present study, however, it is showed how simultaneous modulation of frequency and intensity (dual-parameter modulation

in interleaved frequency condition) can be used to improve the spatial localization of a large number of closely spaced stimulation points (24 pads). The spatial localization has been investigated in [121], however, the stimulation was delivered using fixed frequency and large, conventional electrodes arranged in 2x4 matrix.

Since the aim of the present study was to improve stimuli localization, exploiting both frequency and intensity (dual-parameter modulation) to make the distinction between the columns as clear as possible has been decided. However, due to simultaneous modulation, it cannot be determined how much each of the parameters individually (frequency versus intensity) affected the localization. This question can be addressed by changing one parameter only, while controlling the other, which is relevant but outside the scope of the present study. Therefore, this will be the goal of future experiments. The difference in tactile sensation produced by 10 Hz-low intensity and 400 Hz-high intensity stimulation was clearly perceivable and thereby successfully assisting the participants in identifying the correct column and, in turn, the stimulated spot within the column. However, it was not easy to determine an effective modulation scheme and to this aim, extensive pilot tests have been performed. For example, one approach that has been tested in the pilot experiments was to associate different stimulation frequency to each column, e.g., 5, 10, 20, 50 Hz for columns 1, 2, 3, 4. It turned out that using increasing frequencies (including different ranges and different steps between them) was not effective for better pad recognition and this type of modulation has been abandoned. Therefore, the interleaved frequencies with substantial gap were selected (10 vs 400 Hz), but even in this case, the modulation of frequency had to be complemented with the interleaved intensity, in order to further increase the contrast between the columns. The frequency of 400 Hz was chosen as this was the maximum of the stimulator, but it is likely that the results would be similar with other high frequencies as well (e.g., 10 vs 100 Hz), as the sensations are similar.

In principle, the order of the trials might have an important influence over the success rate in recognizing the pad and optimally it should have been randomized. However, the order that was adopted in the present experiment due to technical constraints was in fact less favorable for the novel modality (dual-parameter modulation). The interleaved stimulation was tested in the first session and same frequency and mechanical stimulation were assessed in the second session, and yet the best results were obtained with the interleaved stimulation. This is further demonstration of the effectiveness of the dual-parameter modulation which resulted in the best performance although the participants were not yet acquainted with electrotactile stimulation.

The dual-parameter modulation has indeed shown to be an effective method for improving spatial localization of the active pad, allowing the electrotactile interface to even surpass the sensitivity of the skin to the natural mechanotactile stimulation on the forearm. The enhanced capability of distinguishing the single pad inside the column might therefore enable high-resolution contact localization inside a specific finger and the proposed high-resolution interface can be used as a research platform to explore different possible scenarios for high-bandwidth sensory transmission to the prosthesis user. An interesting example is transmitting high-resolution information on contact mechanics (e.g. force distribution) which might be required for advanced tasks as dexterous manipulation. For that, this multichannel interface is to be combined with the multichannel sensing systems such as electronic skins developed in robotics but now increasingly considered for prosthetic applications [126].

With the same high-resolution interface, other mappings could be of direct interest to communicate prosthesis variables using high resolution. The columns could represent different prosthesis dofs (hand open/close vs. wrist pro/supination) and the rows could be the levels of the feedback variable (e.g., the aperture or rotation angle). Such an interface would reliably communicate both the current dof (active column) and the variable value (active pad) to the prosthesis user.

However, a drawback of this method is that the spatial localization is improved at the expense of utilizing the two additional stimulation parameters (intensity and frequency). Therefore, they cannot be used anymore to convey feedback information through parameter modulation, as proposed in other approaches (e.g. increasing frequency/intensity to indicate higher grasping force and/or aperture [104, 116, 119]). Instead, the relevant prosthesis variables would need to be communicated spatially through an active pad, as explained before.

More than single information could also be transmitted at the same time. The next step in this research will be to investigate how well the subjects could perceive several electrotactile stimuli that are delivered simultaneously or sequentially along the columns (two or more active pads). In this context, each column/row of the matrix could represent a dedicated prosthesis variable (e.g., hand aperture and grasping force). If the subject could identify the active pads in each column, even when they are activated at the same time, this would allow transmitting several levels of different prosthesis variables concurrently. Such feedback would complement the capabilities of the state of the art regression methods for prosthesis control which allow operating several prosthesis degrees of freedom simultaneously.



## 3.6 Open issues

Sensory feedback information (such as a touch) should be reliable during prolonged periods of use, naturally interpretable and transferred without inducing pain. One of the research goal on sensory feedback is to explore methods for a successfully transmission of the stimulation to a prosthesis user.

Traditionally, researchers have implemented systems that presented the sensory feedback in a continuous fashion. However, continuous feedback yields to adaptation: after a certain period the stimulation is no longer or just barely perceived by the individual.

The adaptation time can be vary depending on the stimulus frequency and intensity over a certain location in electrocutaneous stimulation. In [130], Szeto and Lyman studied the adaptation time as a function of the stimulus frequency (i.d. using fixed stimulation intensity), and as a function of the stimulus intensity (e.d. using fixed stimulation frequency, and different waveform types). In [131], Szeto et al. demonstrated that adaptation do not occur by changing the waveform type. In [102], they also showed that pulse frequency has a large effect on adaptation. Little adaptation occurred when applying 10 pulses per second, while strong adaptation occurred within seconds when using 1000 pulses per second. Adaptation time was also evaluated comparing continuous and intermittent stimuli and focusing on the influence of the stimulation intensity. In [127], they found that higher adapatation (meaning less time for adaptating yourself to the stimulation feeling) occurs for a sensitivity (low) sensations than for a discomfort (high) one. In addition, they demonstrated intermittent stimuli show smaller adaptation.

The fact that higher frequencies cause shorter adaptation [102] can be managed by applying a specific novel stimulation protocol, in which the pulses are distributed sequentially over several closely positioned electrodes. The sequential stimulation of different electrodes over a reduced area gives the subject the impression of a localized stimulation at higher frequency than each electrode frequency.

Several experiments have been conducted in different sessions and days. Figure 3.17 schematically reported some of the tests. Experiments consisted of 8-12 sensation decay measurements. Each measurement lasted maximally 15 min and was followed by a pause of 5 min. The duration of the stimulation and the pause was selected based on the literature, as enough time to lead to adaptation and recovery, respectively. One, two or four pads out of the 16 pads were selected depending on the performed experiment at the right time. Then, the stimulation ran at fixed frequency (e.g. 10 Hz

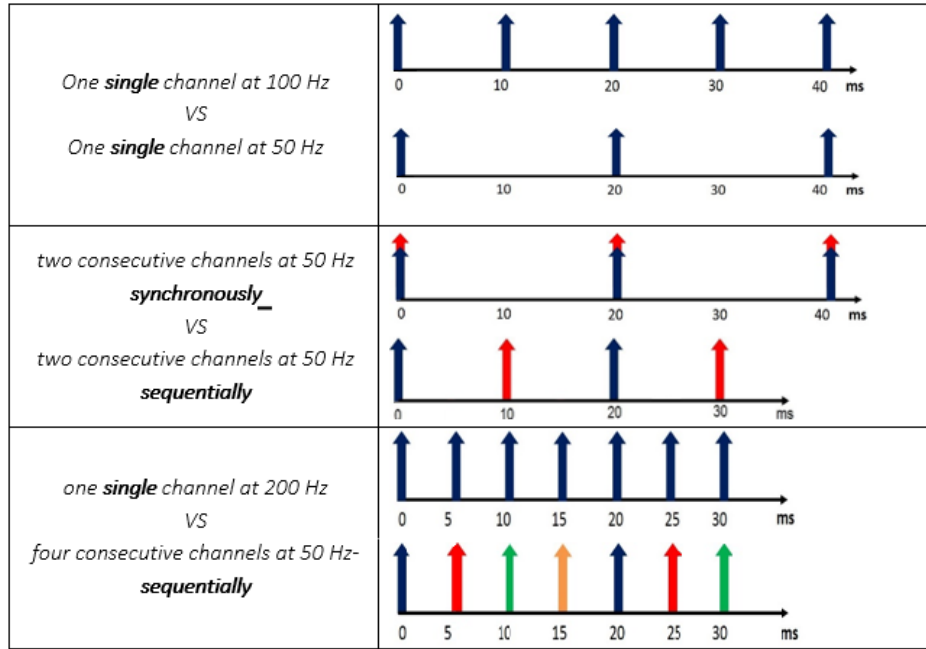


Figure 3.17: Study of the adaptation phenomenon testing electrocutaneous stimulation (synchronously or sequentially) over one single pads or more.

or 100 Hz) and the intensity of stimulation was increased by one-step (1 mA) at a time starting from zero. The participant was asked to identify the sensitivity (low) and discomfort (high) thresholds respectively. The threshold determination was necessary to compute the used stimulation level (corresponding to 50% of the whole available range between *discomfort* – *sensitivity* thresholds) during the experiment.

The subject was instructed to judge the sensation level after an auditory cue, every 30 s. This judgment was recorded using a commonly accepted Visual Analog Scale (VAS). VAS is a line of 10 cm with two boundaries, the left corresponding to no sensation, and right to discomfort. The subject was required to quantify the sensation level by inserting a cross on the line of the VAS scale. Afterwards, the subject was instructed to identify new sensitivity and discomfort thresholds after an auditory cue, every 60 s. To measure the adaptation phenomenon the same method employed in [127] have been exploited here for a fair comparison.

The main outcome of these tests were that the concept of higher adaptation for higher frequencies cannot be generalized. Many participants were experimentally tested, but a clear statement was not achieved by observing the obtained results. Besides, using the VAS scale was too difficult and subjective. The VAS scale seemed neither reliable measure of an absolute intensity (a reference seems to be needed to

give a number with a certain confidence) nor of a relative one (it seems that the needed reference value is something in between the last stimulation (stronger) and the memory of the initial one (more uncertain and vague). Personal perception also is highly variable and depend on many variables: electrode positions on the skin, muscle movements / activation during the test, attention (also affected by motivation), fatigue, expectations, physiological variables (e.g. hormones and autonomic nervous system), and emotions (e.g. anxiety and fear).

In order to increase the prosthesis utility, another aspect to consider is the prosthesis embodiment. A way to measure the embodiment effect is to conduct a test similar to the Rubber Hand Illusion test [132]: Artificial Rubber Hand Illusion (ARHI) test. Figure 3.18 shows the comparison of the setup between RHI and ARHI tests.

The experimenter will use a small paintbrush to stroke the artificial rubber hand, while the subjects hidden hand will be solicited by the corresponding electrostimulations. Each stroke will be about 2-3 cm long. During this phase, the subjects will be instructed to relax and observe the artificial rubber hand on the table. The ARHI test will consist of five trials. Each will last for 2/3 minutes using an irregular rhythm and between every trial will be 40 seconds. When the ARHI test is finished, subjects will complete a questionnaire. The questionnaire will indicate if subjects experienced an illusion in which they would feel the touch they were looking at, as if they owned the artificial rubber hand. To obtain behavioural evidence that the illusion happened, a pointing task would be required (adopted in [132]). Immediately before starting the experiment and after each trial, the participants will be required to close their eyes and point to where they feel their arm. A ruler mounted on the table will be used to measure the distance between the point indicated by subjects index finger and his real arm. The larger the distance is, the higher the number of subjects who feel the artificial rubber hand as their real hand. Obviously, the pointing drift should be from the real hand towards the artificial rubber one.

A first pilot study was evaluated, but further investigation are needed.

## 3.7 Conclusion

Human reaching and grasping is a complex motor task for which the biological controller needs to coordinate the kinematics and dynamics of many degrees of freedom (DOFs) concurrently. Yet, they are performed routinely, with a remarkable ease, effortlessly, numerous times during the activities of daily living (ADL). This remarkable performance requires a combination of feed-forward and feedback control,

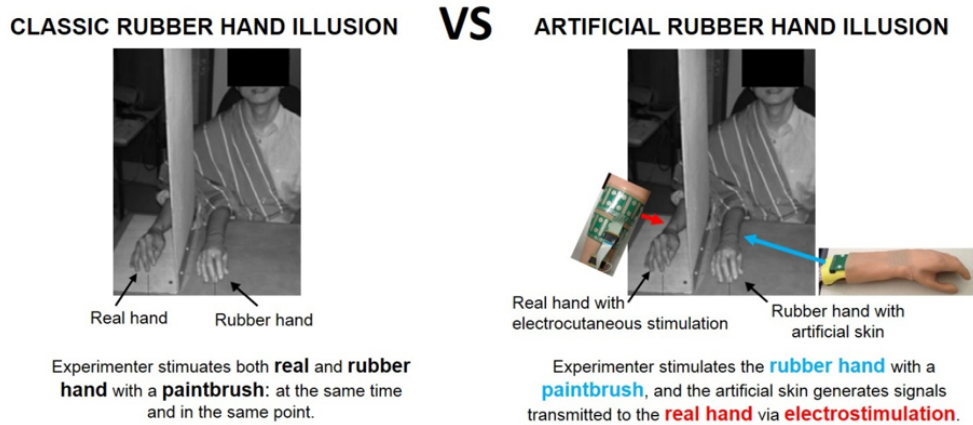


Figure 3.18: Rubber Hand Illusion versus Artificial Rubber Hand Illusion setup.

integrating sensory information from multiple sources and motor commands based on internal models of the environment and the body acquired through learning and development [133], [134].

Apart from a single recent example [3], none of today's prostheses have purposely designed sensory feedback. Body-powered prostheses provide awareness of the terminal device through the control cable. Battery-powered prostheses provide only incidental clues. Therefore prosthesis users largely rely on vision or audio feedback when it is possible. However as anticipated, it is known from the human motor control that bilateral communication through the provision of sensory feedback is also instrumental for learning and execution of the movements [75]. Therefore, to foster the prosthesis embodiment and utility, it is necessary to have a prosthetic system that not only responds to the control signals provided by the user, but also transmits back to the user the information about the current state of the prosthesis. For a user friendly control of these complex devices this new approach proposes a radically different control framework, mimicking characteristics of the biological control, such as the comprehensive perception of the environment, multimodal sensor fusion and cognitive-like processing.

Nowadays, many studies are focused on evaluating different methods for provide prosthesis with sensory feedback. Sensory feedback can be elicited invasively, by interfacing directly the nerves or non invasively, by providing feedback on the skin. Direct nerve stimulation is a viable means for eliciting proprioceptive and touch sensations, but the quality of the sensation is usually a foreign feeling, due to technological limitations of today's neural interfaces.

Sensory substitution has also been investigated in prosthetic research. Prevalent techniques have been vibrotactile and electrotactile. Although the sensation is substituted, these devices are low-cost and a low-power and can be actually be integrated in the prosthetic socket.

Moreover, modality-matched feedback is achievable with non-invasive mechanotactile devices; even though these are bulky, power-consuming and expensive.

In moving toward the new concept for prosthesis capable of being implemented for long-term use, the concept of providing natural, physiological feedback must be considered. The participants concentration cannot entirely be dedicated to interpreting the feedback signals provided as day-to-day activities incorporate varying levels of complexity with corresponding concentration required of the user. Besides, as many systems proposed in literature require training and sensory adaptation to interpret signals [101], an additional processing of information could increase the cognitive load and could have the potential to negate one of the largest benefits of sensory feedback reducing conscious attention. It is important for researchers to develop better measures for evaluating the usability of sensory feedback systems in day-to-day life, such as reporting on the naturalness of measured sensations as well as the amount of cognitive burden required.

Another consideration is regarding the timing of the sensory feedback. Delay from the sensory input should be in the order of milliseconds. A too prolonged stimulus could be made the feedback less effective and cause adaptation.

# Chapter 4

## Applications

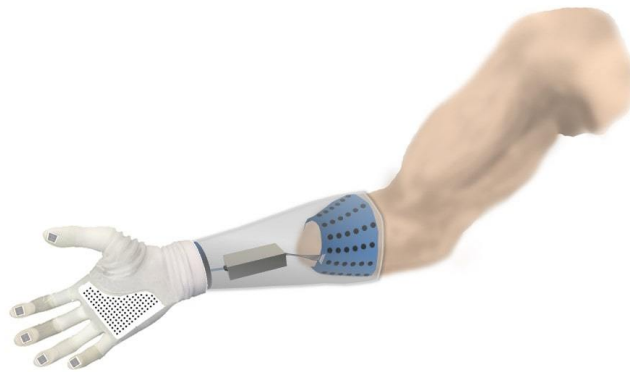


Figure 4.1: Ottobock Hand

### 4.1 Introduction

The results presented in this chapter were published in [120]. The candidate particularly contributed to the published work by:

- Developing of algorithms to process tactile data using Matlab.
- Developing of the experimental setup.
- Designing of the experimental protocol and pilot tests. on healthy subjects to evaluate the capabilities of the human brain to successfully interpret the elicited artificial tactile information.
- Designing a program in Matlab to automatize the experimental protocol, store the data and generate preliminary results.
- A comprehensive statistical analysis of the collected data using STATISTICA tool.

A myoelectric prosthesis can be used to compensate for the loss of motor function after a hand amputation. In this system, the electrical activity of the users muscles is recorded, processed and then decoded to estimate the user intention. Decoding is performed using simple heuristics, as in the classic two-channel control, or machine learning (pattern classification and regression) in multichannel systems [1], [135]. Typically, the same muscles that were used to accomplish a motor task before the amputation control analogous functions in the prosthesis, e.g., hand and wrist flexors and extensors control prosthesis closing and opening, respectively. This association provides an intuitive connection between the brain and the prosthesis, but this connection is unidirectional. However, a bilateral communication between the brain and the periphery, including both efferent and afferent information flow, is necessary for the human movement learning and execution [75]. Closing the prosthesis control loop by providing sensory feedback to the user is a key point in research on active prosthetics [136], [4] as well as an often cited requirement of the prosthesis users [137]. Nevertheless, apart from one recent example [3], there are still no commercially-available prostheses transmitting somatosensory information back to the user.

Artificial tactile feedback can be provided using a method known as sensory substitution [138] (see Chapter 3). The prosthesis is equipped with sensors measuring the system state (e.g., joint angles) as well as interaction with the environment (e.g., grasping force). To restore the feedback, the sensor data are read online, translated into stimulation profiles according to the selected coding method and the stimulation is delivered to the user invasively (e.g., neural [59] and brain stimulation [61]) or non-invasively (e.g., electro [102] and vibrotactile interfaces [101]). The information is transmitted by modulating the stimulation parameters.

The fidelity of the feedback that can be provided to the user depends on the available sensors and stimulation interfaces. Typically, current prostheses are equipped with only basic position and force sensors [139]. Importantly, some more sophisticated sensing elements are becoming available. BioTac [140] is a sensorized finger equipped with a matrix of pressure sensors across the fingertip, as well as a vibration and temperature sensor. Electronic skins integrating matrices of tactile (and other) sensing elements embedded into flexible structures have been fabricated (eg [141–145], references in [146]).

An electronic skin (e-skin) can be therefore integrated into a prosthetic device to endow the prosthesis with artificial cutaneous sensing by electromechanically converting touches into electrical signals to be delivered to the human subject. The information from multiple contact points can be used by the embedded prosthesis controller to

automatically evaluate the grasp stability, control grip formation and/or prevent slipping [147]. In the context of feedback to the user, such sensors are comprehensive sources of tactile signals, which can be utilized to provide an unprecedented level of afferent inflow, facilitating thereby the integration of the prosthetic device and the user body and mind, as proposed conceptually in [148–150], and stimulating the cognitive and psychological mechanisms related to body ownership [151]. Some recent sensors also implement biologically-inspired coding of the tactile information. For example, [152] describes a flexible electronic skin capable of quantifying the applied pressure and generate electrical signals to directly deliver such a sensory input in the form of electrical pulse trains mimicking nerve action potentials to a living nerve cell of the mouse brain.

Regarding the stimulation interfaces, non-invasive methods are still most common [4]. Many studies provided important insights regarding the role and advantages of feedback in prosthetics, but they utilized simple force and position sensors embedded into the prosthesis. These conventional stimulation configurations are not suitable for interfacing with the aforementioned advanced sensing systems. In order to transmit the information from a large number of sensing elements, available in the e-skin systems, the electrotactile interface needs to implement many stimulation channels integrated into a compact device.

The first prototype of such a distributed sensing and stimulation interface has been presented and the feasibility of communicating the tactile information to a human subject has been evaluated [120]. The system comprises an e-skin including an array of sensing elements (taxels), acquisition electronics and multichannel stimulator connected to flexible matrix electrodes placed on the forearm.

Although the concept of applying e-skin technology for somatosensory feedback in prosthetics has been proposed earlier [4], the present study is the first evaluation of an online system implementing the prospective communication link, i.e., the information transmission from e-skin to the human subject.

## **4.2 System to Close the Loop in Prosthetic**

### **4.2.1 System Architecture**

The tactile feedback system comprises the following components (Figure 4.2): 1) e-skin with 64 sensing elements (taxels), 2) custom-made electronics for the signal conditioning, 3) data acquisition card (NI cDAQ 9174, National Instruments, USA), 4) fully programmable multichannel electrotactile stimulator (MaxSens, Tecniaia,



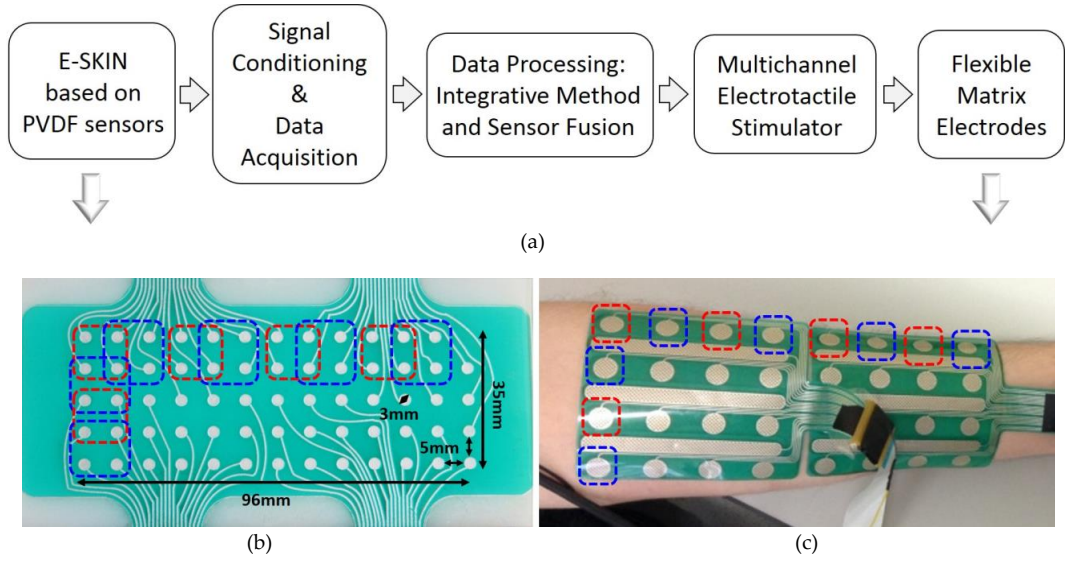


Figure 4.2: (a) System Architecture. The system comprises e-skin with 64 sensing points, charge amplifier assessing the response of each taxel to mechanical stimulus, data acquisition card to sample the signals, and multichannel stimulator with flexible matrix electrodes integrating 32 electrode pads to deliver the electrotactile stimulation to the subject. (b), (c) E-skin and Electrode Array: Sensor Fusion. The taxel signals are processed using integrative method and the signals from the neighboring taxels ( $2 \times 2$ ) are fused in order to map more taxels to less electrode pads (red and black dashed lines).

Spain) connected to two flexible matrix electrodes, and 5) a host laptop PC. The host PC runs the online control loop which was implemented in Matlab 2013b and operated as follows. The e-skin converted mechanical information (i.e. a specific way of touching) into a set of electrical signals (one signal per sensor). The taxel signals were sampled at the frequency of 1 kHz and the data were segmented into intervals of 50 ms for processing (as described below). The stimulation parameters were sent to the stimulator, which delivered the stimulation. The stimulation parameters were therefore refreshed at 20 Hz ( $1/\text{data window length}$ ).

The e-skin (Figure 4.2b) was a rectangular patch ( $12.8 \times 4.8$  cm) made of Polyvinylidene Fluoride (PVDF), which is a piezoelectric polymer directly converting mechanical stress into charge [153]. The intrinsic flexibility of this polymer in the form of a film together with its wide frequency bandwidth (1 Hz-1 kHz) and large measurable pressure range (50 Pa-1 MPa) make it a good candidate as functional constituent of a flexible e-skin measuring dynamic contacts. Sixty-four different sensors (taxels) were obtained by screen printing patterned electrodes [154] on a single commercial 100-μm thick PVDF sheet [155]. Taxels were circular (radius 1.5 mm) and arranged in a rectangular array ( $9.6 \times 3.5$ ), with the sensor pitch (centre-to-centre distance) of approximately 8 mm. In the present application, the PVDF sheet was placed on a rigid substrate and an elastic layer (Polydimethylsiloxane) was added on top for

stress transmission and sensor protection [156]. The charge is read from each taxel by a custom-made multi-channel charge amplifier (CA), converted into voltage and conditioned by a band-pass filter (details are reported in [157]). The analog outputs from the amplifier were connected to the analog inputs of the data acquisition card, which sampled the signals at 1 kHz and communicated the data to the host PC via a USB port.

The MaxSens stimulator whose details are explained in Chapter 3 (Section 3.4) was also employed for this study.

The stimulator was connected to two flexible matrix electrodes (Figure 4.2c). These electrodes were custom designed, each array consisting of 16 round electrodes with the radius of 0.5 cm, arranged in a  $4 \times 4$  geometry, with the total size of  $10.4 \times 9.1$  cm. The distance between two adjacent electrodes was approximately 1.5 cm, which is well above the two-point discrimination threshold for electrotactile stimulation ( $\sim 9$  mm on the forearm [126]). The round electrodes were the cathodes whereas the long bars in-between the cathodes served as the common anodes. It should be noted that the stimulation pulses were biphasic with symmetrical compensation, so that the current changed direction between the stimulus and the compensation pulse of the same width and amplitude. In this context, the terms anode and cathode refer to the physical design of the electrode and the consequent role of the pad (i.e., depolarizing vs. neutral) rather than polarity (i.e., electrical current direction). Namely, the surface of the anode pads was more than 10 times larger than the surface of the cathodes. The anodal current densities were therefore significantly lower and below the threshold required to excite the cutaneous afferents. Consequently, the tactile sensations were produced only below the cathodes, as confirmed by the psychometric assessment [158]. This configuration was selected in order to localize the current flow, from a cathode to its closest anode segment, eliciting focused tactile sensation.

The electrodes used in the present study were fairly large, as they were based on a design that was developed and tested previously [158]. Importantly, the technology for electrode production is flexible and allows custom solutions, as explained in previous paragraph. Therefore, an important future goal will be to optimize the design of the electrode (e.g., decrease the pad size, increase pad number and density etc.).

The intensity and quality of the electrostimulation (isolated touch, vibration, tingling) [159], [160] can be adjusted by modulating the pulse amplitude, width, and rate of delivery (parameter coding) as well as the location of stimulation by changing the active electrode (spatial coding) [102], [101].

### 4.2.2 Tactile data processing

Before starting the online loop, the baseline levels of the taxel signals were determined. Signals were recorded from the skin for at least 5 s with no mechanical interaction and the mean value of the collected samples was computed for each taxel. During the online operation of the system, the baseline was subtracted from each taxel output, centering the signals at zero level. The zero-mean taxel data were then multiplied by a calibration matrix that represented the scaling factors compensating for the variation of sensitivity across individual taxels.

The data were acquired from the skin in blocks of 50 samples corresponding to the data segments of 50 ms. After applying the calibration steps, as described above, the integrative processing was implemented. The taxels are piezoelectric sensors, capturing the dynamic aspects of the mechanical input to the skin. Figure 4.3a,c depicts the taxel signals responding to a step input in pressure, i.e., a finger pressing and then releasing the skin. As a response, the taxels generated two phasic burst outputs (Figure 4.3c, d), indicating the pressure and release events, while in-between there was almost no response apart from some wiggling due to small movements of the pressing finger. The contact event was indicated with a positive deviation whereas release generated a negative deflection of the taxel signal. Also, the figure demonstrates that the mechanical stimulus propagates across the skin (Figure 4.3a), activating neighboring taxels and even taxels distant with respect to the point of contact, with the response intensity decreasing with the distance.

Raw taxel data were processed in order to capture the static features of the mechanical input and also to suppress the aforementioned spurious activations. For the latter, a threshold was defined and the taxel signals below the threshold were set to zero. Considering that contact and release events elicited transient responses of opposite polarity (Figure 4.2c), an integrative method was implemented to estimate the static response: a mean value within the 50-ms data segment was computed for each taxel and the taxel means were added cumulatively across the time intervals.

More formally, let's define (i)  $S_i(k)$  as a sequence of vectors of  $N = 64$  elements (one element per taxel), representing taxel outputs at the time instant of  $k$  milliseconds, where  $k = 1, 2, \dots, 50$ , within the  $i$ th data segment (50 ms duration), and (ii)  $\bar{S}_i$  as a vector ( $N \times 1$ ) containing taxel data averaged across the  $i$ th data segment:

$$\bar{S}_i = \frac{1}{50} \sum_{k=1}^{50} S_i(k) \quad (4.1)$$

The mean tactile data from the present data segment  $\bar{S}_i$  were added to the cumulative mean  $\bar{S}_{i-1}$

of all the previous data segments yielding:

$$\bar{S}_i \leftarrow \bar{S}_i + \bar{S}_{i-1} \wedge \bar{S}_0 = 0 \quad (4.2)$$

The final output of this processing was a single value per taxel per data segment. The processed taxel signals are depicted in Figure 4.2(b-d). The raw signals are sampled at 1 kHz while the processed data are outputted at the sampling rate of 20 Hz (new value each 50 ms). During this time interval (Figure 4.2c vs Figure 4.2d), the signals rise as the pressure is applied, remain at the non-zero level while the pressure is maintained, and decrease back to zero as the pressure is released. The initial positive deflection (contact) increased the cumulative mean, which then remained constant during contact since the taxel signal was below the threshold. Finally, a negative deflection due to release decreased the cumulative mean. To counteract the fact that positive and negative deflections were not completely symmetric, the cumulative mean

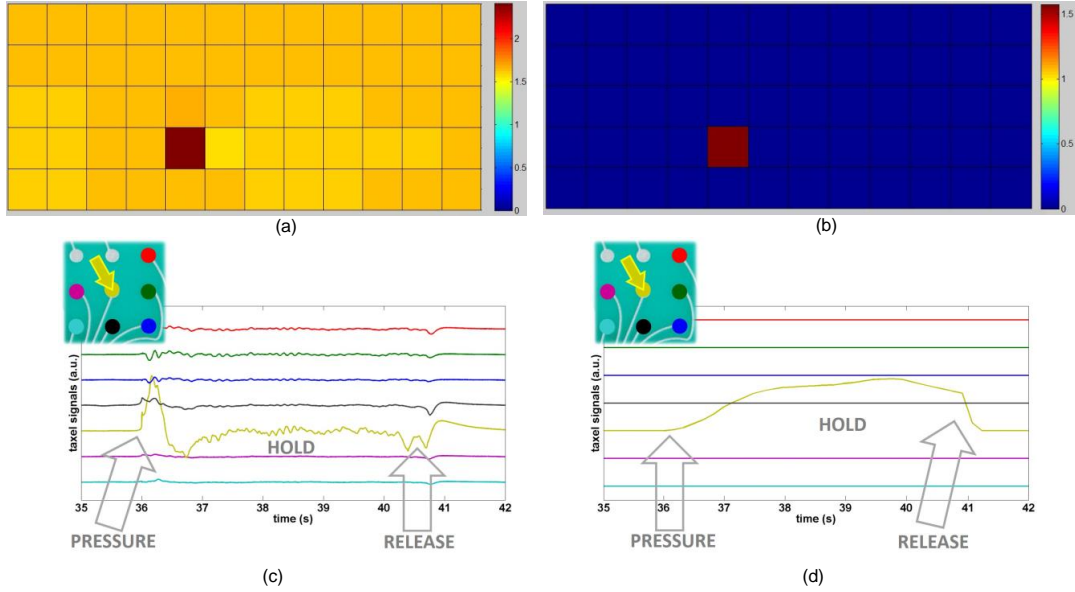


Figure 4.3: Electrical response (taxel signals) due to a pressure applied to the e-skin using a finger. The mechanical stimulus comprised a step input (pressure-hold-release) delivered to a single taxel. TOP: color map representing a snapshot of all the taxel signals while pressure is applied for (a) raw and (b) processed data. BOTTOM: signal traces for the stimulated taxel (inset, yellow arrow) and its neighbors for (c) raw and (d) processed data. The signal plots have the line color of the corresponding taxels (see inset). The signals are zero-mean but were shifted using offsets for the sake of plot readability. The contact and release events elicit positive and negative deflections in the raw signals, respectively. Processed signals are clean from spurious activations and capture also the static properties of the mechanical stimulus.

was further processed. The cumulative mean was forced to zero when below a fixed threshold until it exceeded the threshold due to a new contact event (negative values for the cumulative mean were not allowed). The response signal amplitude scaled with the applied pressure input.

### **4.2.3 Electrotactile spatial coding**

To map 64 tactile sensor outputs to 32 stimulation electrodes, tactile data from neighboring taxels were averaged in groups of  $2 \times 2$ . This resulted in 32 mean values that exactly corresponded to the number of stimulation channels. The mapping between taxel groups and stimulation electrodes is shown in Figure 4.2(Lower panel). The electrotactile channels were activated only if the corresponding mean value was above a predefined threshold, which has been previously chosen for each participant in pilot tests to guarantee a clear perception of the electrostimulation (details are reported in Sect. 3.2). Since the focus in the present study was on the recognition of spatially moving stimuli (see Experimental Protocol), the stimulation location was modulated while the intensity and frequency were kept constant. The stimulation frequency was set to 100 Hz based on our previous experience [161]. A single electrical pulse elicits a brief tactile sensation (transient touch). However, at the high rates, such as 100 Hz, the discrete sensations fuse into a continuous feeling of tingling or tickling, which can be easily and clearly perceived by the subjects. The intensity was set for each electrode and subject individually to just above the sensation threshold (see the Experimental Procedures), resulting in comfortable and localized tactile sensations.

## **4.3 Experimental Results**

### **4.3.1 Experimental Evaluation**

The experiments were conducted in eight healthy volunteers (three females, five males,  $28 \pm 5$  years). The study has been approved by the local Ethical committee of the Specialized Hospital for Rehabilitation and Orthopedic Prosthetics (approval number 1172). Before starting, the subjects signed an informed consent form.

The subjects were seated comfortably on a chair in front of a table. The forearm of the dominant arm was placed on the table surface, with the volar side oriented upwards. The electrode arrays were positioned on the volar aspect of the subject forearm. The first electrode was placed 1 cm below the elbow and the second right after it. The electrodes were then secured with medical tape to prevent movement and improve

contact. The electrostimulation was adjusted to be strong enough so that it is clearly perceivable, but without being uncomfortable or causing pain. First, electrotactile sensation thresholds were determined for each of the 32 channels individually using the method of limits [162]. The current pulse amplitude, starting at zero, was increased in steps of 0.1 mA until the subjects reported that they felt the electrostimulation. After determining the subjective stimulation threshold, the perception was additionally checked by repeatedly activating/deactivating the pad, and asking the subject to report when he/she felt the stimulation. If the subject was able to detect the change in state (on/off), the stimulation amplitude was accepted as appropriate, and the procedure was repeated for the next pad. Otherwise, the amplitude was increased until the subject could detect the transition. Furthermore, small adjustments in the current amplitudes were made to roughly equalize the tactile sensations across the electrode pads. The experimenter activated the pads in sequence from the first to the last and the subject reported if the sensation corresponding to a certain pad was too different from the others. In this case, the current amplitude would be increased/decreased, as needed. This was repeated several times until the subject reported that the elicited sensations across pads were indeed similar. This procedure lasted between 15 and 30 min. The experiment comprised blocks of different trials in which the experimenter applied a mechanical stimulus to the e-skin, while the system online translated this stimulus into stimulation profiles delivered to the subject (as described before).

The task for the subject was to perceive the stimulation and report to the experimenter the properties of the mechanical stimulus.

The *mechanical stimulus* was applied to the skin using a small roller to ensure contact reproducibility between different trials. To produce a natural and realistic motion, the experimenter moved the roller along the skin at a self-selected speed and pressure, aiming to be consistent across the trials.

Three categories of mechanical patterns have been tested (see Figure 4.4 and Figure 4.5): single lines (10 patterns), geometrical shapes (7 patterns) and letters (8 patterns). The lines and geometrical shapes were additionally organized in subcategories (see Figure 4.4).

The subject could not see the experimenter, nor the skin. A sheet of paper was placed in front of the subject with a schematic drawing of the electrode array. After receiving the electrical stimulation, the subjects were first asked to verbally estimate the perceived shape of the movement: longitudinal, transversal, diagonal line; square, rectangle or triangle; an exact letter (Figure 4.5). If the shape estimate was correct, the subjects were asked to retrace the perceived stimulus by moving their index finger

|                   |                       |                     |  |
|-------------------|-----------------------|---------------------|--|
| <b>CATEGORIES</b> | <b>LINEs</b>          |                     |  |
|                   | <b>SUBCATEGORIES</b>  | <b>LONGITUDINAL</b> |  |
|                   |                       | <i>lt</i>           | Longitudinal: <i>Top</i> , from left to right                |
|                   |                       | <i>lm</i>           | Longitudinal: <i>Middle</i> , from right to left             |
|                   |                       | <i>lb</i>           | Longitudinal: <i>Bottom</i> , from left to right             |
|                   |                       | <i>lb/</i>          | Longitudinal: <i>Bottom</i> , from left to middle            |
|                   |                       | <b>TRANSVERSAL</b>  |  |
|                   |                       | <i>ti</i>           | Transversal: <i>first</i> column, from top to bottom         |
|                   |                       | <i>to</i>           | Transversal: <i>fourth</i> column, from bottom to top        |
|                   |                       | <i>te</i>           | Transversal: <i>Eighth</i> column, from top to bottom        |
|                   |                       | <i>tt</i>           | Transversal: <i>Twelfth</i> column, from bottom to top       |
|                   |                       | <b>DIAGONAL</b>     |  |
|                   |                       | <i>drl</i>          | Diagonal: from bottom <i>Right</i> corner to top <i>Left</i> |
|                   |                       | <i>dlr</i>          | Diagonal: from bottom <i>Left</i> corner to top <i>Right</i> |
|                   | <b>GEOMETRIES</b>     |                     |  |
|                   | <b>SUBCATEGORIES</b>  | <b>TRIANGLE</b>     |  |
|                   |                       | <i>bt</i>           | Bottom pointing <i>Triangle</i>                              |
|                   |                       | <i>ltr</i>          | Left <i>Triangle</i>   |
|                   |                       | <i>rt</i>           | Right <i>Triangle</i>  |
|                   |                       | <b>RECTANGLE</b>    |  |
|                   |                       | <i>br</i>           | Bottom <i>Rectangle</i>                                      |
|                   |                       | <i>tr</i>           | Top <i>Rectangle</i>   |
|                   |                       | <b>SQUARE</b>       |  |
|                   |                       | <i>ls</i>           | Left <i>Square</i>   |
|                   |                       | <i>rs</i>           | Right <i>Square</i>  |
|                   | <b>LETTERS</b>        |                     |  |
|                   | <b>TOUCH MODALITY</b> | <i>L</i>            | L  |
|                   |                       | <i>T</i>            | T  |
|                   |                       | <i>N</i>            | N  |
|                   |                       | <i>X</i>            | X  |
|                   |                       | <i>Z</i>            | Z  |
|                   |                       | <i>J</i>            | Mirrored T   |
|                   |                       | <i>Λ</i>            | Mirrored V   |
|                   |                       | <i>Σ</i>            | Mirrored Z   |

Figure 4.4: Touch Modalities. Movement test patterns applied to the e-skin. Each line in this table corresponds to a single touch modality (pattern). S (Shape), T (Trajectory) and D (Direction): touch modality features.

across the schematic drawing. If this corresponded to the pathway of rolling when delivering the stimulus (and thereby applied electrical stimulation), it was deemed that the subjects correctly recognized the movement trajectory. Finally, if the subjects retraced the trajectory segments (lines) in the proper direction, as indicated by arrows in Figure 4.5, the movement direction was also correctly perceived.

During the first three blocks, after the subject reported the recognized movement features, the experimenter provided feedback about the correct features of the applied movement, in terms of shape, trajectory, and direction. The experimenter verbally described and also demonstrated the specific movement using a roller (reinforced learning phase). In the last two blocks, such information was not provided to the subject (test phase).



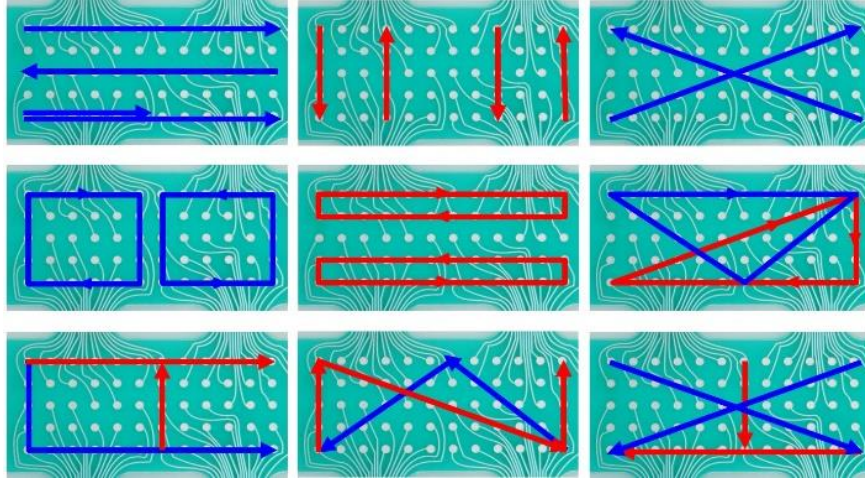


Figure 4.5: Example of movement test patterns applied to the e-skin by an experimenter using a roller. TOP: longitudinal (left), transversal (center), diagonal (right) single lines. MIDDLE: square (left), rectangular (center), triangle (right). BOTTOM: letters: T and L (left), N and mirrored V (center), X and mirrored T (right).

### 4.3.2 Data analysis

The main outcome measure was the success rate (SR) in recognizing a feature of the movement, i.e., movement shape, trajectory and direction. The SRs for the trajectory and direction were conditioned on the correct recognition of the shape, as described above. The SRs were computed per subject for the specific categories (lines, geometries, letters) and sub-categories, i.e., shapes (lines: longitudinal, transversal and diagonal; geometries: rectangle, square and triangle) of the tested mechanical stimuli. The letters were treated differently, with no subcategories, as each letter was peculiar and different from others. The SRs of all subjects were then averaged to obtain the overall mean SR and its standard deviation. The results were reported as mean  $\pm$  standard deviation in the text and the figures.

The data were tested for normality using Kolmogorov-Smirnov test. In case the test indicated normal distribution, one- and two-way repeated measure ANOVA was applied to assess statistically significant differences at the level of the group followed by Tukeys honest significant difference test for post hoc pairwise comparison.

If the distribution was not normal, the Friedman test was used instead of ANOVA, while the post hoc analysis remained the same. Following this scheme, one-way ANOVA or Friedman test (as appropriate) were used to compare the success rates in recognizing movement features across movement categories and sub-categories (see Figure 4.4). Finally, two-way ANOVA could be applied only in the case of geometries



(category), where the factors were subcategory (triangle, rectangle, and square) and movement feature (trajectory, direction). To evaluate the effect of training, the success rates in recognizing all the movement features simultaneously during training (blocks 1 to 3, pooled) and testing (blocks 4 and 5, pooled) were compared using two-way ANOVA with category (lines, letters, geometries) and phase (training, testing) as the factors. The threshold for the statistical significance was adopted at  $p < 0.05$ , and the statistical analysis was conducted in Matlab R2014a (MathWorks, US).

### 4.3.3 Results

There was no significant difference in performance between the reinforced learning and testing phase. There was no significant interaction nor main effect in the two-way ANOVA for the factors category and phase. Therefore, the data from both phases were pooled together and used for the subsequent analyses.

Figure 4.6 shows the overall mean SR across all subjects, movement features and categories. In general, the recognition of the dynamic stimuli was not an easy task for the subjects. The SRs for the correct identification of all the movement features simultaneously Figure 4.6a) were poor ( $66 \pm 15\%$ ). The SRs were similar for simple

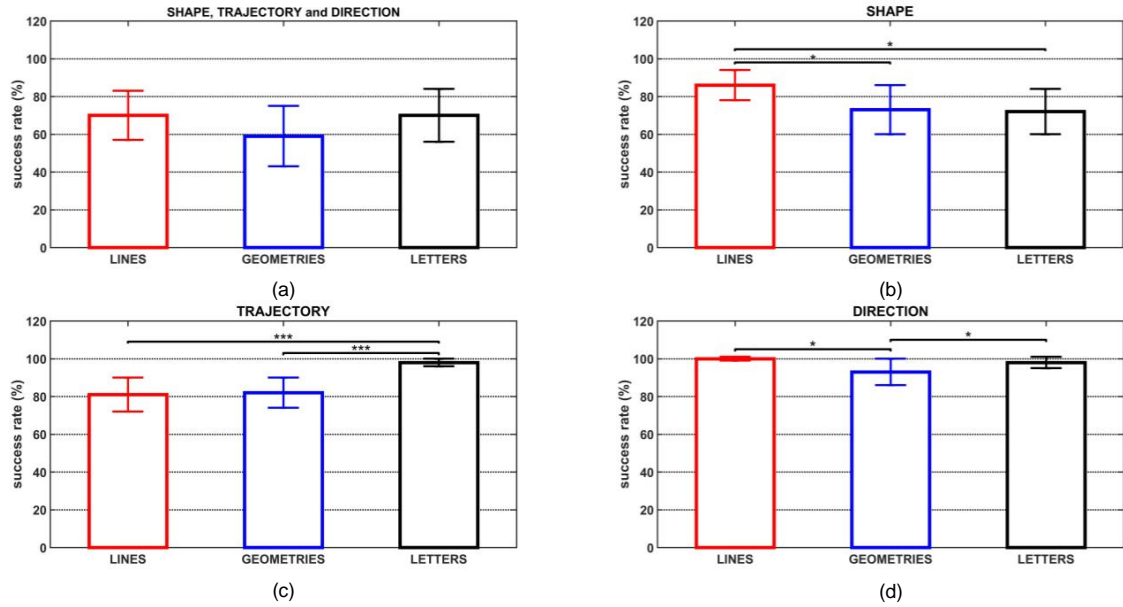


Figure 4.6: Overall success rate of recognizing either (a) all three movement features (shape, trajectory, direction) together, or (b) shape, (c) trajectory and (d) direction provided that the shape has been correctly predicted. This analysis is performed for single lines (RED bars), geometric shapes (BLUE bars) and letters (BLACK bars). Bars and stars indicated statistical significance (\*,  $p < 0.05$ ; \*\*,  $p < 0.01$ ; \*\*\*,  $p < 0.001$ )

movements (single lines) and more complex dynamic patterns comprising several segments (geometries and letters), with no statistically significant differences between. Importantly, the subjects were more successful when identifying the movement shape ( $77\pm13\%$ ), which means that they could interpret the semantic aspect of the movement (e.g., recognize a specific letter or shape). For the movement shape, the performance (Figure 4.6b) was better ( $p < 0.05$ ) for the lines ( $86\pm8$ ) than for the geometries ( $73\pm13$ ) and the letters ( $72\pm12$ ).

When the shape had been properly recognized, the subjects could retrace the full trajectory in  $87\pm10\%$  of cases (Figure 4.6c), and they could guess the movement direction almost perfectly ( $97\pm5\%$ ,  $p < 0.05$ ) (Fig. 4d). Interestingly, the movement trajectory was traced significantly better for the letters ( $98\pm2\%$ ) compared to the lines ( $81\pm9\%$ ,  $p < 0.001$ ) and the geometries ( $82\pm8\%$ ,  $p < 0.001$ ), while there were no significant differences between the latter two. Regarding the movement direction, the SR was lower for the geometries ( $93\pm7$ ) than for the lines ( $100\pm1\%$ ,  $p < 0.05$ ) and the letters ( $98\pm3\%$ ,  $p < 0.05$ ). The SR for the direction of the lines and the letters was similarly high (no statistical difference).

For the subcategories (transversal, longitudinal, and diagonal) of the line patterns (Figure 4.7 - red bars), there was no difference in performance when recognizing the shape (Figure 4.7a) and direction (Figure 4.7c). In particular, the subjects could identify the direction almost perfectly (SR  $\sim 100\%$ ). However, the trajectory tracing (Figure 4.7b) was significantly better for the diagonal lines ( $99\pm4\%$ ) compared to transversal ( $79\pm14\%$ ,  $p < 0.01$ ) and longitudinal lines ( $76\pm13\%$ ,  $p < 0.01$ ). The three best recognized patterns were the diagonal (drl), transversal (t), and longitudinal (l) line. Overall, the subjects were more successful (Figure 4.6 vs. c,  $p < 0.001$ ) in identifying the line direction ( $100\pm1\%$ ) than in tracing the trajectory ( $81\pm9\%$ ).

Regarding the geometries (Figure 4.7 blue bars), the movement shape was identified significantly better in triangles ( $81\pm14\%$ ) and squares ( $76\pm19\%$ ) compared to rectangles ( $56\pm19\%$ ,  $p < 0.01$ ). There was no significant interaction nor main effect in the two-way ANOVA for the factors subcategory and movement feature (trajectory and direction). As for the single lines, direction recognition was easier (Figure 4.6c vs. d), i.e., the overall SR was  $82\pm2\%$  for trajectory tracing vs.  $93\pm7\%$  for direction; but there were no significant differences between these two. The geometries with the highest SR (Figure 4.8b) were triangle (ltr) and square (ls).

The best recognized letters were Z, mirrored Z and L (Figure 4.8c). Note that those are the letters composed of the best recognized single lines (see Figure 4.8a). For the letters, contrary to lines and geometries, there was no statistically significant

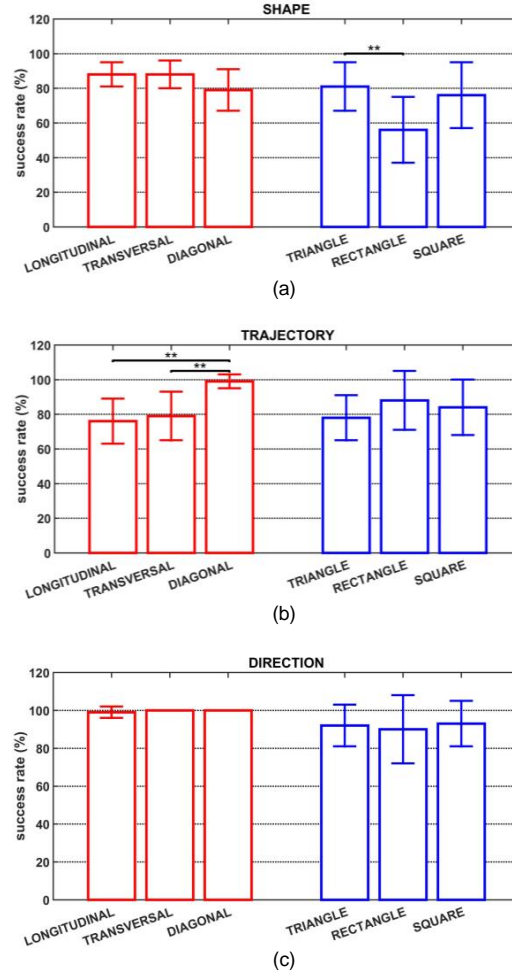


Figure 4.7: Success rate of recognizing the shape (a), trajectory (b), and direction (c), for each category of lines (longitudinal, transversal, and diagonal indicated by RED bars) and of geometries (triangle, rectangle and square indicated by BLUE bars). Bars and stars indicated statistical significance (\*,  $p < 0.05$ ; \*\*,  $p < 0.01$ ; \*\*\*,  $p < 0.001$ ).

difference between the trajectory tracing (Figure 4.6c,  $98 \pm 2\%$ ) and identification of direction (Figure 4.6d,  $98 \pm 3\%$ ).

## 4.4 Discussion

The ability of the human subjects to identify the direction of motion over the skin has been investigated in earlier studies [163], [164] to obtain insights into normal and pathological human sensory processing. Typically, an apparatus that could apply well controlled motion of the fine brush over the skin surface was used for the tests. The studies demonstrated that directional sensitivity depends on the body site (mechanoreceptor density), increases with longer trajectories, exhibits preferential range of movement velocities (decreases for too slow and too fast movements), and does not to depend on the movement orientation. The direction sensitivity was tested also

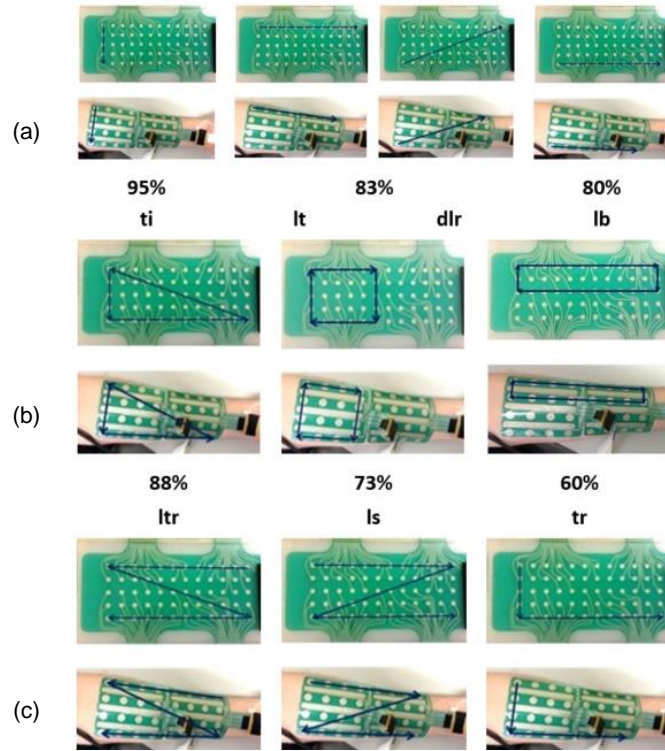


Figure 4.8: Three best recognized patterns: single lines (a), geometries (b), letters (c). For each panel, pictures at the top illustrate how each pattern was applied to the e-skin, pictures below indicate the electrodes activated during electrotactile stimulation to communicate the pattern to the subject, while the success rate and the textual code (Table 1) for the pattern are reported at the bottom.

by simulating movements using a matrix of tactile stimulators delivering predefined stimulation profiles [165]. The authors concluded that success rate in discriminating directions depends on the number of successively activated stimulation points (amount of sent information), rather than the distance between the first and last point.

In [166], the tactile motion patterns were simulated by controlling a 24 x 6 matrix of miniature probes indenting the skin. Moving patterns were produced by turning on and off the rows of the matrix in fast succession. The stimulation was delivered to the palm and fingers, since the intended application was the substitution of hearing using distributed tactile stimulation. The setup was used in a sequence of studies with microneurography [166–168] for the investigation of the basic properties of mechanoreceptors. However, the tactile profiles were preprogrammed, i.e., the setup did not include the sensing element such as the e-skin used in the present study.

The results obtained in the present experiment demonstrate that the novel system successfully translated the tactile signals into profiles of electrical stimulation using

the proposed online processing and control. The human subjects could successfully interpret the delivered multichannel stimulation to recognize the properties of tactile stimuli moving across the skin surface. The whole process was running online and thereby under realistic conditions, leading to variability in tactile signals and electrical profiles across the trials, e.g., occasionally activating taxels and electrodes that did not belong to the ideal trajectories (Figure 4.8). Importantly, the ability to recognize all the movement features correctly was similar across simple and more complex patterns, indicating robustness. Regarding the shape detection, the subjects could well identify lines with different orientation (longitudinal, transversal, diagonal) with no preference with respect to the specific orientation along the forearm. For more complex patterns, the ability to identify the shape decreased, but the subjects could still recognize quite reliably specific patterns comprising 3 to 4 connected lines (Figure 4.8). They could even precisely retrace the full movement trajectory in a reasonable number of cases, and recognize the direction of the movement along the skin with a high confidence. The performance in recognizing direction was consistently better than in trajectory tracing, overall and also for the lines and the geometries. The consistency in identifying the direction is in accordance with the results of previous studies [163–165], considering that our motion patterns were rather long and with multiple stimulation points (e.g., 8 electrode pads for longitudinal lines).

In general, the subjects were more successful with the patterns stretching across the full extent of the e-skin (and thereby matrix electrode) compared to the patterns localized to smaller areas. This is indicated by the fact that the performance was better for the letters than for the geometries in both trajectory tracing and direction identification. Furthermore, the subjects had difficulties when they needed to precisely localize one of several patterns with the same shape but different positions on the forearm. This is a likely reason for the worse performance of trajectory tracing for lines vs. letters, since the former category contains closely spaced longitudinal and transversal lines (Figure 4.8). Despite the fact that additional measurements are needed to extend this analysis to a larger number of subjects and investigate the learning effects through training, it can be concluded that this sensory feedback system is feasible and intuitive. There are several possible reasons why the training failed to improve the performance. The training could have been too short and/or dynamic to allow learning. There were many shapes (10 lines, 7 geometries and 8 letters) which were presented randomly and only three times per shape. In addition, there was a level of variability, i.e., tracking the same shape did not evoke exactly the same electrotactile activation, as explained before in this section. Finally, the subjects

were not allowed to look at the skin while the experimenter applied the pattern, and therefore he/she received only the tactile information. As demonstrated before [169], multisensory integration might facilitate the learning. All in all, optimal training paradigms for the presented tactile system are yet to be determined.

The tests employed in the present study are similar to those used for assessing graphesthesia [170], [171], which refers to the ability of recognizing numbers traced on the subject own skin. These tests are applied to assess the capability of the tactile sensor system to process differences in the position and direction of a moving tactile input and fuse that information into higher cortical representations. This assessment is conducted in healthy subjects as well as in patients to provide basic insights into the nature of sensory processing and/or diagnostic conclusions. The promising results obtained in the present study demonstrate that the developed skin substitute maintains the consistency of tactile perception and processing. In [170], the subjects recognized 10 numbers (0-9, size: 5 x 7 cm) presented by the experimenter moving his index finger along the skin of the dorsal side of the forearm with an SR of 89%. In another study [171], 12 letters were presented by moving a stylus over the palm of the subject resulting in the SR of 75%. In the present study, 8 letters (9x20 cm) were recognized (shape) with a success rate of approximately 72%. This is in agreement with the previous studies considering that the patterns were delivered using a non-ideal, online loop with electrotactile stimulation (compared to ideal, direct contact with the skin, as used in [170], [171]). The present study demonstrated that the subjects were able to recognize complex shapes, which is an encouraging result. However, the electrical profile was transmitted to the user by sequentially activating the pads along the trajectory. If the profile was presented in parallel, by simultaneously activating all the pads, the recognition would probably worsen [172]. The latter paradigm is likely to be more common during the practical application (e.g., a prosthetic hand grasping an object).

The experimental protocol in the present study was defined by considering the limitations of the available setup and the scope of the current work. These first results are encouraging considering the future applications of the proposed system in prosthetics. The fact that human subjects were able to perceive the tactile features of moving electrotactile stimuli can be of relevance for prosthesis control as well as for embodiment. As pointed out in [165], moving tactile stimuli can be an effective method for information transmission to the user, because mechanoreceptors respond stronger to this type of stimuli compared to simple (static) pressure. Also, such perception can be the basis for the exploration of the environment, as when the subject would

move the prosthetic hand across the object surface to assess texture by relying on artificial tactile stimulation, as demonstrated through simulations in [173]. Finally, the transmission of a rich, dynamic information between the subject and the prosthesis can be used to promote the embodiment of the artificial system. A similar set of mechanical test patterns, as those used in the present study, could be applied to the prosthesis allowing the subject to perceive, visually and through a tactile channel, a set of realistic interactions. This would be similar to the classic setup for the rubber hand illusion but including a variety of stimulation profiles. The richer stimulation could facilitate the illusion, though this needs to be tested in future experiments.

## 4.5 Conclusion

A system was implemented to transmit mechanical information from a multipoint tactile sensor (e-skin) to the human subject using multichannel electrotactile stimulation. The system was evaluated by assessing the ability of the human subjects to perceive the properties (shape, trajectory, direction) characterizing dynamic and versatile (lines, geometries, letters) mechanical interaction with the skin. To the best of our knowledge, this is the first development integrating an advanced tactile sensor with many sensing elements and an electrotactile stimulation unit with a flexible matrix of electrodes into an online system for the transmission of tactile data from artificial to natural skin (forearm).

However, the current study did not yet demonstrate the utility of the proposed technology in the real-life application, which is an important future goal. To this aim, the next step is to cover a myoelectric prosthetic hand with an e-skin in order to test the closed-loop system during functional tasks, as shown the example in Figure 4.9

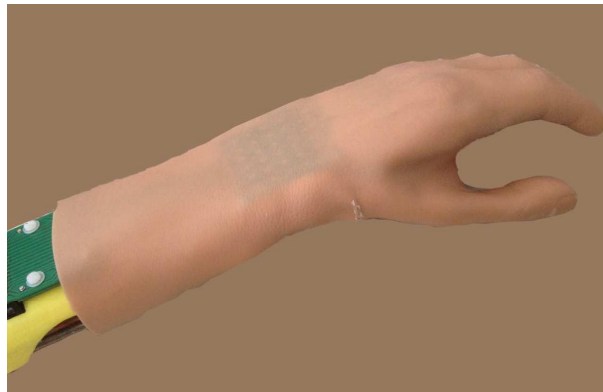


Figure 4.9: Rubber hand integrating the e-skin.

where the prosthetic hand was actually a rubber hand. The scope is to assess if a sophisticated tactile feedback would increase the utility of the device and/or facilitate the feeling of embodiment. An e-skin covering the prosthetic hand would provide sophisticated tactile sensing that could be used to react to the environment. For example, high sensitivity of the e-skin could be exploited to detect and thereby prevent incipient slip. This could be done automatically, through the prosthesis controller (as in Autograsp feature of Otto Bock Sensor Hand [174]), and/or by transmitting slip feedback to the user, so that he/she can react and manually tighten the grasp (as demonstrated in [175]).

Moreover, contrarily to the passive reception of the electrotactile stimuli, as in the present study, placing the skin on the prosthetic hand would allow the user to actively explore the environment. In this case, the tactile feedback would arise as an interaction between volitional commands and movements of the user and the objects with which he/she interacts. This is known to affect the haptic performance and is therefore an important factor that needs to be considered in the future [16,176].

Importantly, the e-skin used in the present setup is only an illustrative example of what can be done. The skin production technology is flexible and allows designing patches of different physical characteristics (e.g., size, shape, pad density). For the practical application, the e-skin will be dimensioned according to the prosthetic hand to which it will be applied. Ideally, the e-skin should be placed to cover the whole prosthesis, mimicking the skin of the natural hand (as proposed in [43]). The current work regards placing the prototype on the volar side of the Otto Bock Michelangelo Hand (e.g., finger pads, palm) to support stable grasping.



## Part III

# Circuits and techniques for the embedded implementation of tactile data processing and decoding



# Chapter 5

## Tactile data processing

### 5.1 Introduction

Before presenting the results achieved by the implementation of approximate techniques into Machine Learning algorithms for the classification of touch modalities ( i.e., the classification of attributes of the contacting objects such as roughness, textures, patterns and shapes), this chapter describes methods for processing tactile data. In Chapter 3 and 4, the electronic skin system processes tactile data as low level information, instead methods for processing high level information are considered in this chapter. Furthermore, the section better motivates why approximate techniques, introduced in the next Chapters, are needed in the hardware implementation of the touch modality classification algorithm.

An essential task of the electronic skin system is to process the tactile data and send information either to mimic human skin or to respond to the application demands. The electronic skin must be fabricated together with an embedded electronic system which has the role of acquiring the tactile data, processing, and extracting information.

Tactile data processing requires efficient methods to extract meaningful information from raw sensor data and deals with different kinds of information, which could be divided into two categories: low- and high- level information. Tactile data processing concerns lower level information as that is related to an accurate estimation of finger-object interaction such as contact location, area and duration, contact force intensity, direction and distribution, together with temperature. Alternatively, tactile data processing involves higher level information for the classification of attributes of the contacting objects e.g. roughness, textures, patterns, shapes, as well as data related

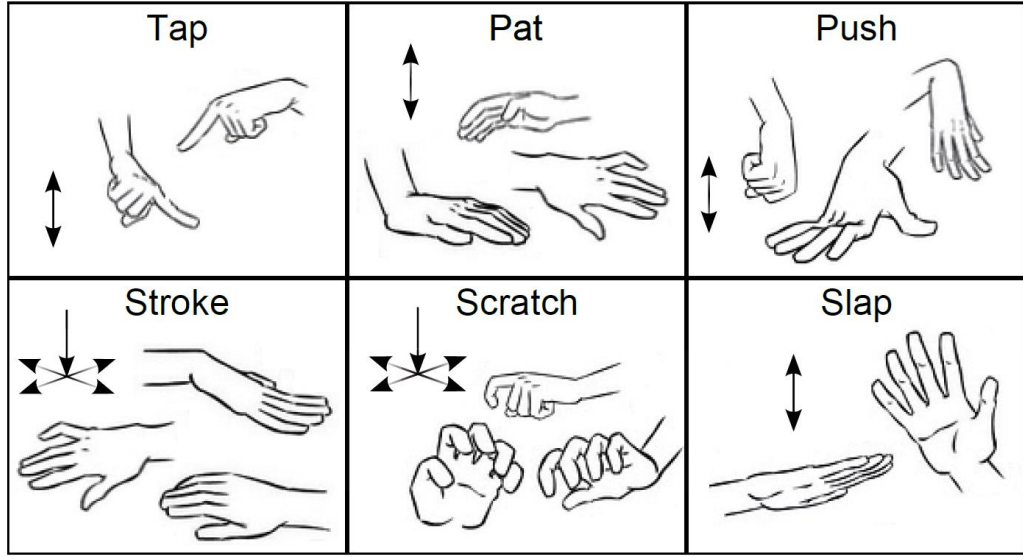


Figure 5.1: Several examples of touch modalities ( Source [183]).

to object movement on the cutaneous surface, e.g. slip detection, vibration, up to the discrimination of the touch modality.

Machine learning (ML) has increasingly been employed to provide solutions for difficult tasks, such as image and speech recognition. Besides, ML algorithms have been exploited to classify and interpret input touch modalities as they represent a powerful technology for tackling clustering, classification and regression problems in complex domains, e.g., in robotics to retrieve partial contact information on specific systems [55, 177–181]. In this thesis, the ML which is based on Tensorial Kernel approach and mentioned in [182], has been proposed to interpret touch modality in the e-skin system.

The rationale behind the choice of adopting ML-based methodologies is twofold. First, ML techniques can support predictive systems that make reliable decisions on unseen input samples [184]. This ability is especially appealing in the case of the interpretation of sensor data, as complex, non-linear mechanisms characterize the underlying phenomenon to be modelled and an explicit formalization of the input-output relationship is difficult to attain. ML technologies model the input-output function by a learning from examples approach (Figure 5.2); eventual implementations can vary according to different application scenarios, but all share a common probabilistic setting.

Second, the theoretical framework introduced in [185] allows one to extend every learning machine based on kernel methods to a tensor-based learning model. This

## Learning by examples

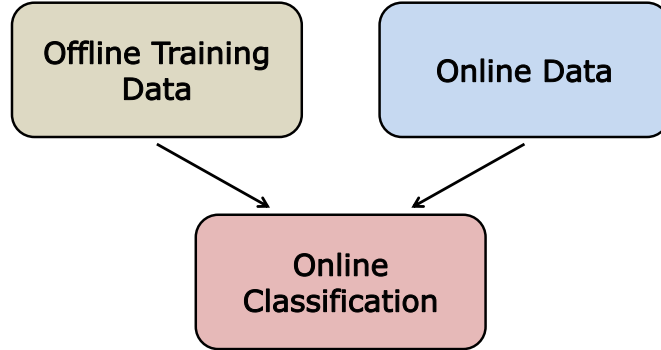


Figure 5.2: Machine Learning approach.

research exploits such aspect by proposing a pattern-recognition framework that involves the class of regularized kernel methods [184], which address the intricacies of the minimization problem embedded in empirical learning by controlling the complexity of the solution. This feature may indeed prove useful when coping with the issue of learning from small training set, as is the case for the interpretation of touch modalities.

## 5.2 Classification Algorithms in Machine Learning

In machine learning, classification is the problem of identifying to which of a set of categories a new observation belongs, on the basis of a training set of data containing observations (or instances) whose category membership is known. More specifically, given an  $n \times d$  training data matrix  $D$  (database  $D$ ), and a class label value in  $1 \dots k$  associated with each of the  $n$  rows in  $D$  (records in  $D$ ), create a training model  $M$ , which can be used to predict the class label of a  $d$ -dimensional record. Many data mining problems are directed toward a specialized goal that is sometimes represented by the value of a particular feature in the data. This particular feature is referred to as the class label. Therefore, such problems are supervised, where in the relationships of the remaining features in the data with respect to this special feature are learned. The data used to learn these relationships is referred to as the training data. The learned model may then be used to determine the estimated class labels for records, where the label is missing. The record whose class label is unknown is referred to as the test record.

To date, there is a lot of classification algorithms available, which one is superior to other depends on the application and nature of available data set. To name a few:

- **Logistic Regression** (LR), that is a machine learning algorithm for classification, the probabilities describing the possible outcomes of a single trial are modelled using a logistic function.
- **Naïve Bayes** (NB) algorithm based on Bayes theorem with the assumption of independence between every pair of features.
- **Stochastic Gradient Descent** (SGD): a simple and very efficient approach to fit linear models. It is particularly useful when the number of samples is very large.
- **Decision Tree** (DT) that produces a sequence of rules that can be used to classify data given a data of attributes together with its classes.
- **Support Vector Machine** (SVM) representing the training data as points in space separated into categories by a clear gap that is as wide as possible.

### 5.3 ML based on Tensorial Kernel Approach

Considering the processing of tactile data deriving from the e-skin, the problem is to interpret the sensor signals to discriminate between a set of stimuli that the system is expected to recognize. ML techniques may indeed face challenging assignments such as the discrimination of materials or the interpretation of touch modalities. To this purpose, one can reduce the overall complexity of the pattern recognition problem by splitting the modeling process into two tasks:

1. The definition of a suitable descriptive basis for the input signal provided by the sensor (or lattice of sensors), i.e., a feature-based description that lies in a feature space  $\mathcal{F}$ :

$$\phi(\mathbf{S}) \rightarrow \mathcal{F} \quad (5.1)$$

In (5.1),  $\mathbf{S}$  is the third-order tensor that characterizes sensor outputs and represents actually the composition of the two-dimensional geometry of the e-skin with the third-dimension for the time response of the sensor itself.

2. The empirical learning of a model for the non-linear function,  $\zeta$ , that maps the feature space,  $\mathcal{F}$ , into the set of tactile stimuli of interest:

$$\zeta : \mathcal{F} \rightarrow T \quad (5.2)$$

Here,  $T$  includes a finite number of stimuli; hence,  $\zeta$  in principle, implies a multi-class classification task.

In the literature, a wide range of ML-based techniques has been proposed to set up  $\zeta$ . However, taking into account the peculiarities of a tactile-sensing system the range of solutions that better fit shrink. The setup of  $\zeta$  is supported by a theoretical approach that can lead to a tensor-oriented kernel machine. The function  $\zeta$  is learned by using Support Vector Machine algorithm. Accordingly, the pattern recognition module can benefit from (5.1) a powerful machine-learning paradigm (Support Vector Machine); and (5.2) a suitable processing of sensors data.

### 5.3.1 Support Vector Machine

Support Vector Machines (SVM) is a very well-known method for the classification of both linear and nonlinear data. The SVM uses a nonlinear mapping to transform the original training data into a higher dimension. Within this new dimension, it searches for the linear optimal separating hyperplane. With an appropriate nonlinear mapping to a sufficiently high dimension, data from two classes can always be separated by a

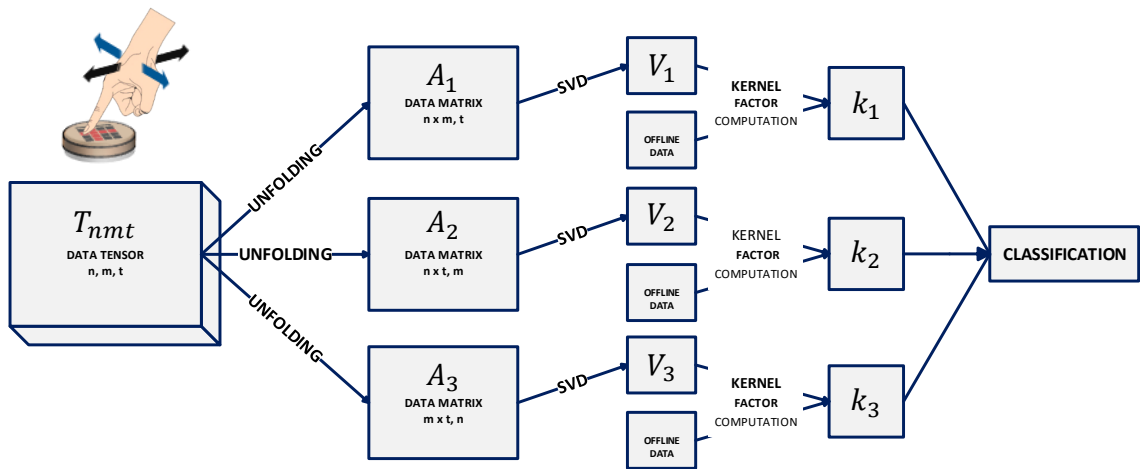


Figure 5.3: Online classification exploiting the Tensorial Kernel approach.

hyperplane. The SVM finds this hyperplane using support vectors (training tuples) and margins (defined by the support vectors).

In general, given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall (Figure 5.4).

The simplest case is a two-class problem, where the classes are linearly separable. The data set  $D$  is given as  $(X_1, y_1), (X_2, y_2), \dots$  and  $X \in \mathfrak{R}^{n \times d}$ , where  $X_i$  is the set of training tuples with associated class labels,  $y_i$ . Each  $y_i$  can take one of two values, either  $+1$  or  $-1$  (i.e.,  $y_i \in \{+1, -1\}$ ), and  $y = f(x)$  which approximates the relation between  $x$  and  $y$ . If we plot the product  $y\hat{f}$  i.e. the product between the true  $y$  and the estimated value  $\hat{f}$ , we are interested in obtaining the correct tag during classification, so the error function is given by the so called hard loss. When the signs are the same

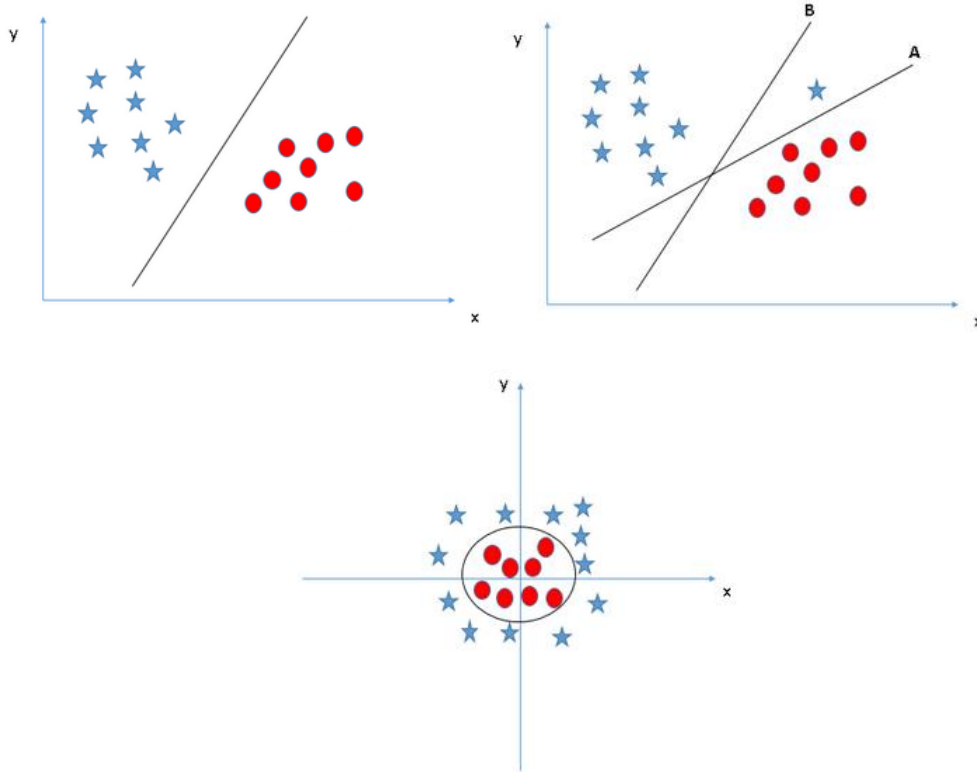


Figure 5.4: Examples of SVM classification method.



we have right prediction, else we have some error. The problem of this function is that it can't be easily minimized (NP problem).

Considering the target application, the SVM classification equation is defined as:

$$\hat{y} = f_{SVM}(x_j) = \sum_{i=1}^{N_P} \beta_i K(x_i, y_i) + b \quad (5.3)$$

where  $x_j$  represents the input,  $\hat{y}$  the predicted category ( e.g.,  $\pm 1$ ),  $\beta_i$  are the weights obtained during the training phase,  $N_P$  is the number of support vectors,  $b$  is the “bias” term, and  $K(x_i, y_i)$  is the kernel function. The SVM aims to compare training and new (on-line) data through the computation of the kernel function. For details concerning  $K(x_i, y_i)$  see Section 5.3.2.

### 5.3.2 Kernel Computation

In machine learning, kernel methods are a class of algorithms for pattern analysis, whose best known member is the SVM. The general task of pattern analysis is to find and study general types of relations (for example clusters, rankings, principal components, correlations, classifications) in datasets. In its simplest form, the kernel trick means transforming data into another dimension that has a clear dividing margin between classes of data.

The kernel function is given by:

$$\mathbf{K}(x, y) = \prod_1^z k^z(x, y) \quad (5.4)$$

where  $k^z(x, y)$  represents the kernel factors. As the target application uses a tensor of three-dimensions,  $z = 1, 2, \text{ and } 3$ . The kernel factors in turn are defined as:

$$k(x, y) = \exp\left(-\frac{1}{2\sigma^2}(I_n - \text{trace}(Z^T Z))\right) \quad (5.5)$$

Referring to [21], the  $Z$  in (5.5) can also be expressed as  $V_x^T V_y$ . Hence, the kernel factors in (5.6) becomes:

$$k(x, y) = \exp\left(-\frac{1}{2\sigma^2} \|V_x^T V_x - V_y V_y^T\|_F^2\right) \quad (5.6)$$

where  $\|\cdot\|_F$  is the Frobenius norm,  $V_x$  and  $V_y$  represents the tensorial inputs for the testing and training phase, respectively. The kernel computation is also intended as

the distance between the singular vector input matrices  $V_{x_i}$  with the corresponding singular vector training matrices  $V_{y_i}$ . Both  $V_x$  and  $V_y$  are obtained by using the Singular-Value Decomposition properly defined in Section 5.3.3.

Equations (5.4) and (5.5) show that the tensorial kernel function  $K(x, y)$  extends the conventional Gaussian kernels to tensor patterns.

### 5.3.3 Singular Value Decomposition

The Singular-Value Decomposition (SVD) factorizes a real  $m \times n$  matrix  $M$  into a product with the form:

$$M = USV^T \quad (5.7)$$

where:

- $U$  is a  $m \times m$  orthogonal matrix ( $U^T U = I_m$ ),
- $S$  is a  $m \times n$  rectangular diagonal matrix with non-negative numbers on the diagonal,
- $V$  is an  $n \times n$  unitary matrix ( $V^T V = I_n$ ).

The column elements of  $U$  and  $V$  matrices correspond to the left-singular and right-singular vectors of  $M$ , respectively. Instead, the  $diag(\sigma_0, \sigma_{n-1})$  elements of the  $S$  matrix are the singular values of  $M$ .

In the considered application, as the tactile data inputs originating from the e-skin are three-dimensional tensors while the SVD only works with matrices (two-dimensional inputs), in order to fill the gap between the SVD input and the e-skin outputs the Tensor Unfolding is explained in the next section.

### 5.3.4 Tensor Unfolding

The Tensor Unfolding main goal is processing tactile data, rearranging them in two-dimensional matrices, without losing any information. In this process, results are matrix representations of tensor: the first contains the columns information ( i.e., all the columns vectors are stacked one after the other), the second contains the rows information ( i.e., all the rows are stacked one after the other), and the last one represents the information brought along the third tensor dimension.

In general, assuming an  $N^{th}$  – order tensor  $A \in C^{I_1 \times I_2 \times \dots \times I_N}$  the matrix unfolding  $A_{(n)} \in C^{I_n \times (I_{n+1} I_{n+2} \dots I_N I_1 I_2 \dots I_{(n-1)})}$  contains the element  $a_{i_1 i_2 \dots i_N}$  at the position with row number  $i_n$  and column number equal to:

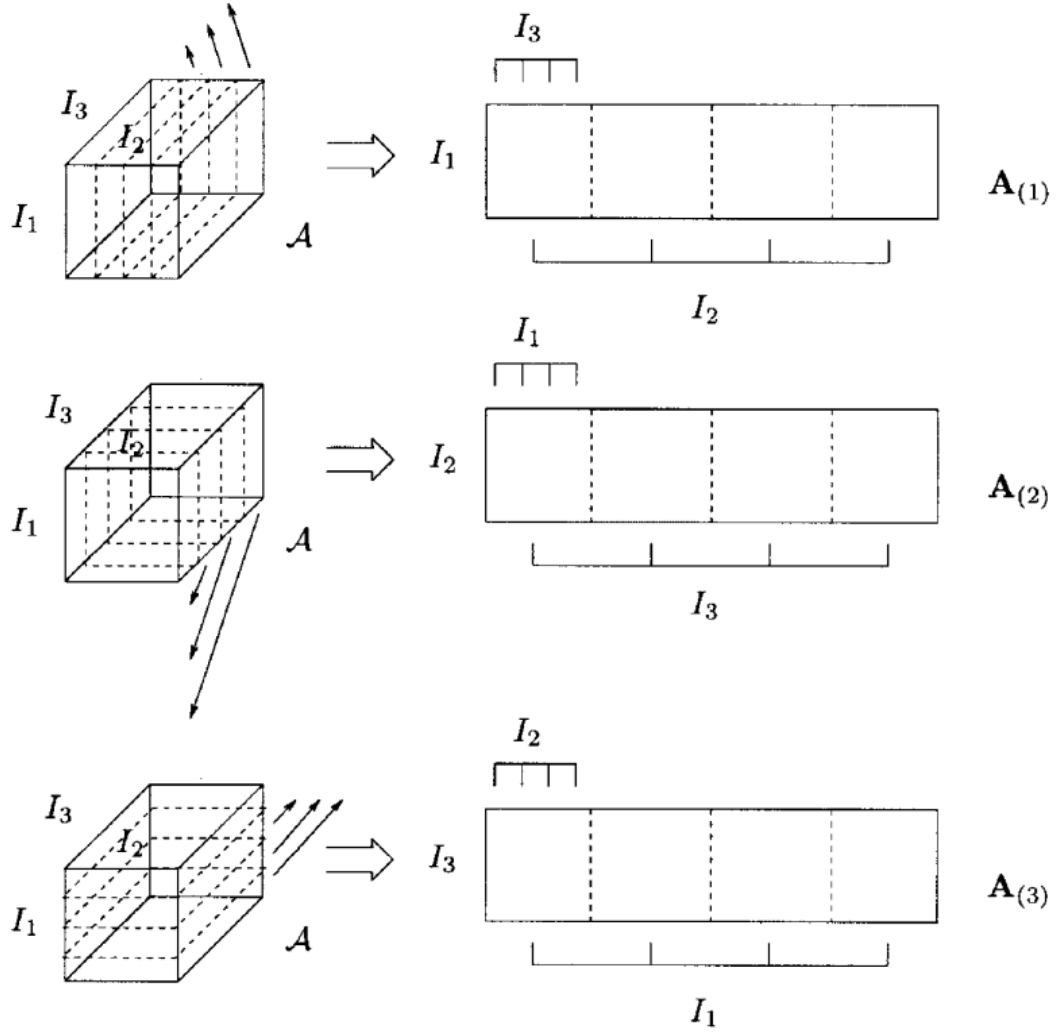


Figure 5.5: Unfolding of the  $(I_1 \times I_2 \times I_3)$ -tensor  $\mathcal{A}$  to the  $(I_1 \times I_2 I_3)$ -matrix  $A_{(1)}$ , the  $(I_2 \times I_3 I_1)$ -matrix  $A_{(2)}$ , and the  $(I_3 \times I_1 I_2)$ -matrix  $A_{(3)}$  (source [22]).

$$(i_{n+1}-1)I_{n+2}I_{n+3}\dots I_N I_1 I_2 \dots I_{n-1} + (i_{n+2}-1)I_{n+3}I_{n+4}\dots I_N I_1 I_2 \dots I_{n-1} + \dots + (i_N-1)I_1 I_2 \dots I_{n-1} + (i_1-1)I_2 I_3 \dots I_{n-1} + (i_2-1)I_3 I_4 \dots I_{n-1} + \dots + i_{n-1}$$

Thus, the third order tensor  $\phi \in C^{I_1 \times I_2 \times I_3}$  results three different matrices unfolding which are:  $A_{(1)} \in C^{I_1 \times (I_2 I_3)}$ ,  $A_{(2)} \in C^{I_2 \times (I_3 I_1)}$ , and  $A_{(3)} \in C^{I_3 \times (I_1 I_2)}$ .

Figure 5.5 shows the unfolding procedure for a three-dimensional tensor.

## 5.4 Computational Complexity

The computational complexity is defined as the number of operations performed by an algorithm. Table 5.1 reports an estimation of the number of operations needed for the different computational functions which are explained in Section 5.3.

The table presents a generic estimation where  $m$  and  $n$  are the dimensions of the unfolded matrix,  $N_c$  is the number of classes to be classified,  $N_t$  the number of training tensors, and  $N_{SV}$  is the number of support vectors obtained from the offline training stage. In [183], a Matlab script has been implemented to automatically generate the operation number needed according to the tensor dimension. They authors considered a case study for an input tensor  $\phi(8 \times 8 \times 20)$  noting that the complexity dramatically increases with the increase of the tensor dimension. The unfolding process results three different matrices ( $A_{(1)}$  ( $8 \times 160$ ),  $A_{(2)}$  ( $8 \times 160$ ),  $A_{(3)}$  ( $20 \times 64$ )) to which the SVD should be applied. Taking into account that the target task is to classify *three* input touch modalities, i.e.,  $N_c = 3$ , and the number of training tensors is set to 100, the number of operations that the algorithm must perform in real time is equal to 302 G-ops.

In [177], details regarding the dataset used as benchmark for the target application are given. Briefly, the database includes seventy participants. They have been involved in the experimental campaign and were asked to touch the tactile sensor array using three different predetermined modalities of touch, i.e. sliding the finger, brushing a paintbrush and rolling a washer as shown in Figure 5.6. Every participant was asked to complete two different actions for each of the three gestures. Thus, first the participant completed the gesture on the tactile sensor array moving horizontally over a random line; then, the same participant completed the same gesture over a vertical random line. The horizontal and vertical stimuli for each gesture were acquired separately. Each participant was allowed to complete every single touch within a time window of 7 seconds. No particular indications were given to the participants about the duration of the stimuli and the pressure level to apply. The overall experiment eventually resulted in a total of 420 patterns (70 participants, 3 gestures, 2 patterns for each gesture - horizontal and vertical stimuli).

|                                     | Number of Operations                               |
|-------------------------------------|--|
| <b>Singular-Value Decomposition</b> | $24m(n-1)[n^2(2n-1) + n^3 + 6]$                    |
| <b>Kernel Function</b>              | $[(2(2n-1)n^2 + (n+3)) + 2] \times N_c \times N_t$ |
| <b>Classification</b>               | $(1 + 3N_t) \times N_{SV}$                         |

Table 5.1: Computational complexity of the studied ML algorithms (source ).

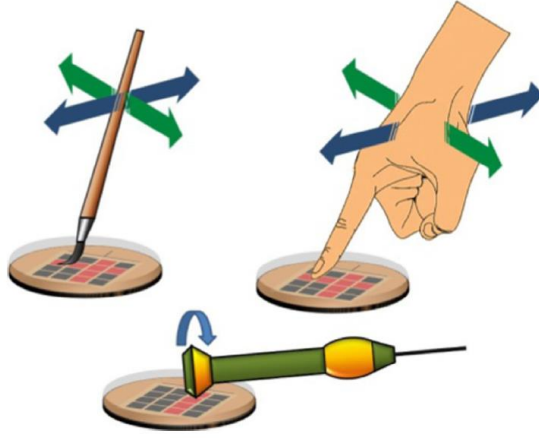


Figure 5.6: Tested touch modalities: brushing a paintbrush, sliding the finger, and rolling a washer (source [186]).

The analysis of the computational complexity of the tensorial kernel approach results the SVD as the most computational expensive algorithm of the overall approach: it represents about the 70% of the computational load [186]. Therefore, methods and architectures for the hardware implementation of the SVD have been investigated and evaluated to select the most suitable architecture for the target application. More specifically, three hardware implementations for the SVD have been proposed in [187] and reported with some further considerations in Section 5.5.

## 5.5 Hardware Implementation of SVD

Because of the crucial role it plays in a wide range of signal processing applications, efficient computation of the SVD is the subject of many publications. A VLSI Design of Singular Value Decomposition Processor for Portable Continuous-wave Diffusion Optical Tomography Systems is presented in [188], the design was implemented using

90nm CMOS process technology and simulation results verify the functionality of the JSVD design within the developed CW-DOT system. Highly parallel accelerators such as Graphic Processing Unit (GPUs) and multi-core platforms have been employed to explore parallel SVD implementations, although these works only achieved speedups when the input matrices have dimensions greater than 1000 [189, 190]. An FPGA implementation based on Jacobi method for singular value decomposition is presented in [191]; it introduced a floating-point Hestenes Jacobi architecture for SVD, which is capable of analyzing arbitrary sized matrices.

Although the most practical SVD algorithm is based on a Jacobi algorithm [192], this could be implemented by using different methods. This section briefly outlines three FPGA implementations of the one sided Jacobi algorithm for SVD based on different computation methods [187].

As the SVD is based on diagonalizing rotations since applying a sequence of rotation to the original matrix  $M$  allows to reach the diagonal matrix  $S$ , the general flow diagram of the SVD computation could schematically be represented as in Figure 5.7.

The Phase Solver (PS) is the block which computes the sine and cosine functions or the angle of rotation. The Rotation Blocks (RBs) consist on the pre- and post-rotation inside of the algorithm. To implement the PS and the RB two different methods each can be used: Jacobi or CORDIC method, and the one sided Jacobi rotation or CORDIC rotation, respectively (see Figure 5.8). Details about the SVD hardware implementation are given in Chapter 6 and Chapter 7, considering both FPGA and ASIC implementation.

Investigating both FPGA and ASIC implementation for the SVD is important. For example, the power consumption could be reduced by exploiting an ASCII implementation. Figure 5.9 illustrates the power consumption variation in terms of number of training tensors when the touch modalities are fixed to three, and Figure 5.10 in

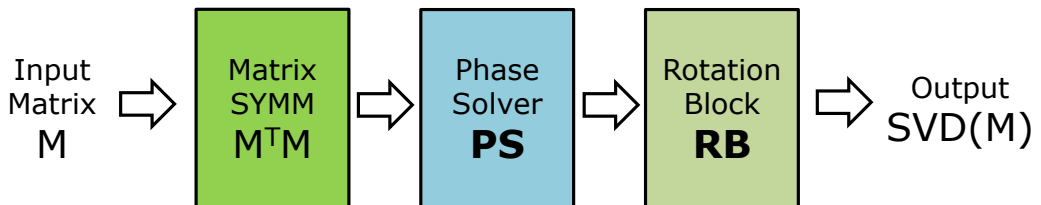


Figure 5.7: General flow diagram for the hardware implementation of the one sided Jacobi algorithm for SVD computation.

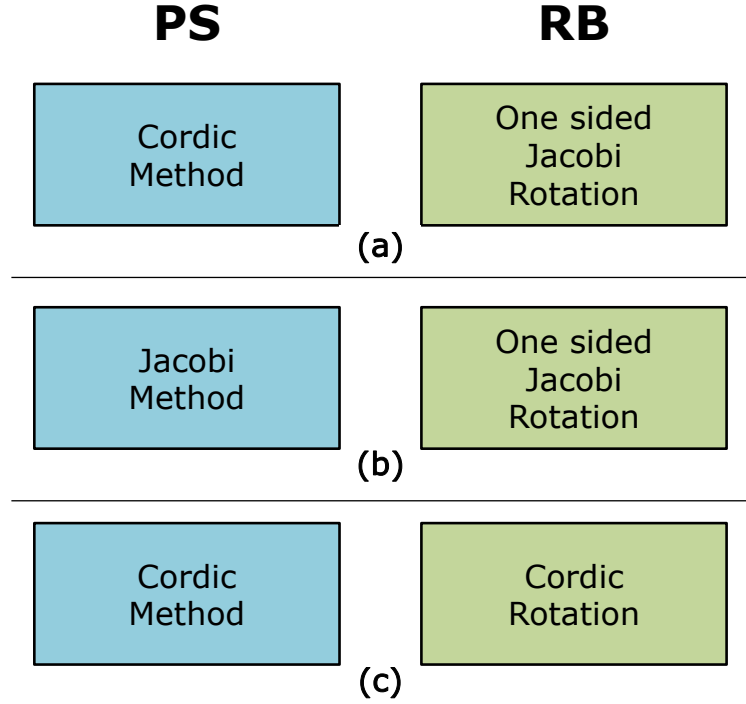


Figure 5.8: Architectures definition based on the different implementation methods (source [187]).

terms of number of touch modalities when the number of training tensors is set to 100. These results point out the power consumption issue which will be tackled in future work by proper circuit design strategies such as approximate computing [193]: it is used to reduce the silicon area and the power consumption by finding a trade-off between quality and performance of the systems. As estimated in Figure 5.9 and Figure 5.10, approximate computing may reduce the power consumption of the system by approximately 25% [193].

Another possible solution to fix the power consumption issue will be addressed by using ASIC implementation (e.g. using standard cell and a deep submicron) technology which may considerably reduce the power consumption of the overall system as estimated in Figure 5.9 and Figure 5.10 [194]. Power supply is a limiting factor in wearable devices whose form factor constrains battery size. Endowing the prosthesis with wearable harvesters that collect energy from the environment will represent a promising solution to achieve the long-life goal for truly wearable and self-powered devices [195].

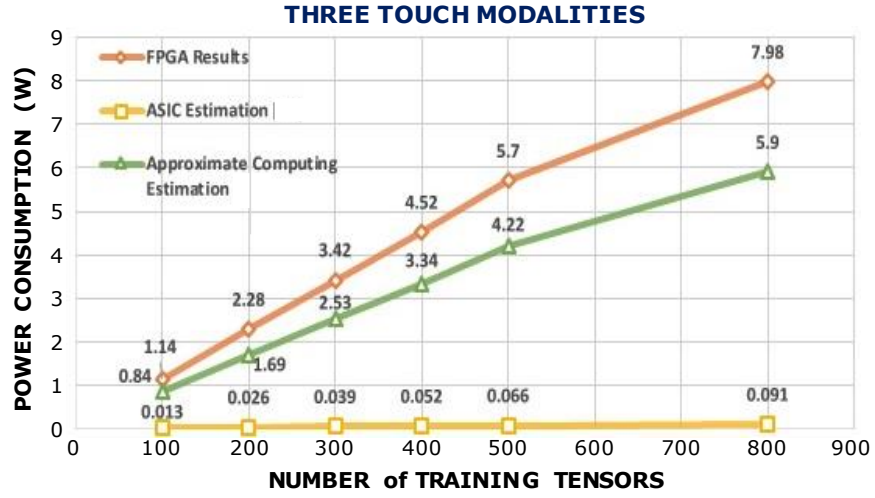


Figure 5.9: Power consumption versus number of training tensors - three touch modalities (source [196]).

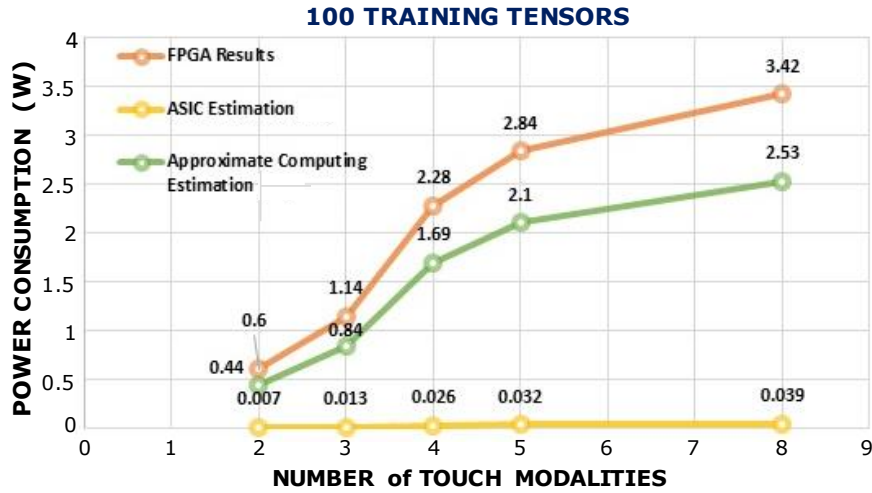


Figure 5.10: Power consumption versus number of touch modalities 100 training tensors (source [196]).

## 5.6 Conclusion

A key challenge nowadays is to provide wearable artificial systems with advanced tactile perception skills, facilitating dexterous exploration and interaction with the environment. Several efforts are underway to bridge the technological gap between artificial and biological skin, which both act as an interface to the external world.

An electronic skin (e-skin) can be used in prosthetics for effective capturing of touch,



and the acquired real-time tactile information can be delivered to the human subject through multichannel cutaneous electrostimulation. This would enable transmitting a comprehensive somatosensory feedback from the prosthesis to the user, thereby closing the loop in myoelectric prosthesis control and facilitating the sense of embodiment of the assistive device into the body scheme of the user [75, 136].

In the perspective of prosthesis utility and embodiment, human sense of touch can inspire research and technological solutions needed to support the required artificial tactile system functions (eg. spatial acuity, frequency bandwidth, sensory function).

On the other hand, wider scenarios can open on how to give back sensor data to the prosthesis user and how to help the human brain to successfully interpret the elicited artificial tactile information. Among open questions in prosthetics is which kind of information - whether *raw* data or *processed* - about a touched object should be sent back to the user. One approach would be to send the sensor signals directly to the user, who needs to meaningfully interpret this information. Alternatively, learning from robotics, sensor data can be locally processed at the body periphery (prosthesis socket with embedded electronics) and high level tactile information (e.g., texture properties, grasp stability) can be extracted and delivered to the user.

Machine learning (ML) methods provide powerful tools for data decoding, they address complex, nonlinear problems by exploiting a learning by examples approach [184]. Therefore, a ML algorithm have been chosen for processing of tactile data deriving from the e-skin.

In this Chapter, Machine Learning based on a tensorial kernel approach have been explained in details. After describing the employed algorithm for touch modality classification, the computational complexity of overall algorithm evaluated in [186] has been reported. In particular, the Singular-Value Decomposition (SVD) represented about 70% of the computational complexity of the overall approach. For this reason, methods and architectures for the hardware implementation of the SVD have been investigated in order to select an appropriate architecture suitable for the targeted application, i.e., electronic skin system for prosthetics.

The analysis conducted in [187] shows the requirements for tactile data processing unit with high level information extraction are very challenging and are still far from being achieved with the current methods. Therefore, methods and techniques to reduce hardware complexity and power consumption of the embedded electronic system have to be explored. More effort should be dedicated to the hardware architectures:

implementation issues should be firstly assessed in order to address an adequate trade-off to balance real time functionality and hardware resources utilization.

In this context, inexact circuit and approximate computing techniques have recently emerged as a promising approach to energy efficient design of digital systems. Using such methods could provide a solution to reduce the hardware complexity and the power consumption of the desired embedded electronic system [197,198].

The next Chapters are focused on applying to the proposed algorithms inexact and approximate circuit techniques.

# Chapter 6

## Approximate Techniques for Hardware Implementation

### 6.1 Introduction

This chapter regards approximate techniques and the importance that are assuming in the last years. More specifically, Chapter ISA aims at implementing approximate circuit techniques in the FPGA implementation of real-time tactile data processing for e-skin application. It focuses on the implementation of the Coordinate Rotation Digital Computer (CORDIC) algorithm, as it is used for several computing tasks such as Singula-Value Decomposition (SVD), the most computationally expensive algorithm for Machine Learning approaches that has ever been considered[199]. This first attempt of approximate CORDIC implementation on FPGA uses Inexact Speculative Adder (ISA) architectures [200], a circuit-level technique optimized for high-speed arithmetic computations. The results for the ISA implementation in the CORDIC were published in [201].

In this chapter, the Inexact Speculative Multiplier (ISM) architecture is also presented as it has been exploited for the SVD implementation, explained in Chapter 5.

The candidate contributed to this work by:

- Designing and implementing the ISM in the SVD algorithm.
- Designing and implementing the ISA in the CORDIC algorithm.
- Analyzing data.
- Considerations and conclusions.

Performance, density and energy efficiency of integrated circuits have been increasing exponentially for the last four decades following Gordon Moore's remarkable prediction. However, power and reliability pose several challenges to the future of technology scaling. Power has definitely emerged as a critical concern due to the poor scaling of  $V_{th}$ , while transistor miniaturization reaching atomic scale has led to tremendous Process-Voltage-Temperature (PVT) variations. Unfortunately, achieving low power and robustness against variability requires complex and conflicting design constraints.

For example, while power efficiency calls for voltage downscaling and minimization of hardware, robustness demands higher voltage, larger transistors and additional correction or redundancy. As a result, designers are being pushed to seek new energy-efficient computing techniques to meet the increasing demand of data processing.

The concept of *error tolerance*, i.e. accepting error in a design to save resources, is well known in many abstraction layers and is already implicit in digital signal processing as the representation of real numbers is approximated due to the finite number of bits. Built on these ideas, approximate computing [202] has emerged as a promising candidate to improve performance and energy efficiency beyond technology scaling. Designing approximate circuits explores a new trade-off, not only by accepting unreliability, but by intentionally introducing errors to save area and power and overcome the limitations of traditional circuit design.

With the exploding amount of data being processed in the cloud and on mobile devices, a wide range of applications can trade accuracy without compromising the functionality or the user experience. For example, in multimedia processing, due to the limited perceptual capability of humans, occasional errors such as dropping a particular frame or a small image quality loss often rarely affect a user's satisfaction.

The growing demand for statistical algorithms such as data mining, search and recognition represents another opportunity to compute in an approximate way as the outcome of those applications is not required to be a single golden result, but an adequate match.

To design approximate systems, several approaches have been investigated at different hardware levels, such as voltage frequency over-scaling [203] at physical level or significance based memory protection [204] at algorithmic level. At circuit level, an interesting approach is to perform computations using approximate arithmetic operators, such as adders and multipliers, allowing a controlled and limited amount of errors against significant power saving or performance increase.

|                     | <b>Approximation Technique</b>  |
|---------------------|---|
| <b>Program</b>      | Loop perforation, Code perforation<br>Thread fusion, Tunable kernels,<br>Patter reduction |
| <b>Architecture</b> | Approximate storage, ISA extensions,<br>Approximate accelerators                          |
| <b>Circuit</b>      | Imprecise logic, Voltage overscaling,<br>Analog computation, Precision scaling            |

Table 6.1: Approximate techniques at different layers (source [205])

In Table 6.1, a few representative approximate techniques spanning from application level to circuit level are shown. In general, at the hardware layer, a less accurate yet more energy-efficient circuit could be used for computation or the supply voltage could purposely be reduced for certain hardware components to trade-off energy and accuracy. At the software layer, certain computations and/or memory accesses that are not critical to the final quality of the application to achieve the same objective could be selectively ignored.

## 6.2 Approximate Techniques: An Overview

Approximate computing techniques present several challenges, which are shown below:

- Limited application domain. Due to their nature, some applications are not amenable to approximation, e.g., cryptography. Therefore, some approximation strategies are only valid in a certain range of an application.
- Correctness issues. Approximate computing techniques may prevent program termination (e.g. a matrix computation kernel where approximate computing can lead to an unsolvable problem), or lead to corrupt output which may not even be detected by the quality metric used [206].
- Finding application-specific strategies. A naive approximation approach such as uniform approximation is unlikely to be efficient. No strategy can be universally applied to all approximable applications. Hence, the approximation strategies need to be determined on a per-application basis by the user or a sophisticated program module [207].

- Overhead and scalability. Several approximate computing techniques may have large implementation overhead, for example, voltage scaling may require voltage shifters for moving data between different voltage domains [208]. Similarly, analog neural network implementations [209, 210] require conversion of signals between digital and analog domain.
- Providing high quality and configurability. Approximate computing techniques must maintain the Quality-of-Results (QoR) to a desired level and also provide tunable knob(s) to trade-off quality with efficiency [211]. If QoR falls below a threshold, the application may have to be executed precisely which increases design and verification cost and may even nullify the gains from approximation.

The strategies used for approximation consist of reducing their precision, skipping tasks, memory accesses or some iterations of a loop, performing an operation on inexact hardware, etc. These strategies are not mutually exclusive. Considering an application/algorithm/circuit, one or more than one of the strategies mentioned above could be used for working on approximable variables and operations once they have been identified. To note that finding approximable variables and operations is the crucial initial step in every approximate computing technique. While this is straightforward in several cases (e.g. approximating lower-order bits of graphics data), in other cases, it may require insights into program characteristics, or error-injection to find the portions that can be approximated with little impact on QoR. Closely related to it is the output monitoring step, which verifies adherence to the quality constraint and triggers parameter-adjustment or precise execution in case of unacceptable quality loss.

Several approximate computing techniques work by changing the precision (bit-width) of input or intermediate operands to reduce storage/computing requirements. Tian et al. [212] present a technique for scaling precision of off-chip data accesses for saving energy. They apply their technique to mixed-model based clustering problem, which requires accessing large amount of off-chip data. They note that in a clustering algorithm, a functional error happens only when a sample is assigned to a wrong cluster. Based on it, the precision can be lowered as long as the relative distances between clusters and samples are still in correct order, so that no functional error happens.

Some techniques use loop perforation approach which works by skipping some iterations of a loop to reduce computational overhead. Sidiroglou et al. [213] identify

several global computational patterns which work well with loop perforation, such as Monte Carlo simulation, iterative refinement and search space enumeration.

Memoization approach is also employed as approximate computing technique. It works by storing the results of functions for later reuse with identical function/input. By reusing the results for similar functions/inputs, the scope of memorization can be enhanced at the cost of possible approximation.

Other approximate methods selectively skip memory references, tasks or input portions to achieve efficiency with bounded QoR loss. Samadi et al. [214] present a SW-based ACT which works by identifying common patterns in data-parallel programs and using a specific approximation strategy for each pattern.

Approximate computing techniques could also exploit inexact/faulty circuits at architecture level. Kahng and Kang [215] present the design of an inexact adder. For an  $N$ -bit inexact adder,  $(N = k - 1)$  sub-adders (each of which is a  $2k$ -bit adder) are used to perform partial summations. The inexact adder avoids the carry chain to reduce critical-path delay and this can be used to improve performance and/or energy efficiency. When a carry input needs to be propagated to the result, the output of all (except the last) sub-adders becomes incorrect. With increasing  $k$  value, the probability of correct result increases but the dynamic power consumption and minimum clock period of the inexact adder also increase. Thus, by changing the value of  $k$ , the accuracy of the inexact adder can be controlled. Venkataramani et al. [216] present an approach for designing general inexact circuits based on register transfer level (RTL) specification of the circuit and QoR metric (such as relative error). Ganapathy et al. [217] present a technique for minimizing the magnitude of errors when using unreliable memories, which is in contrast to the ECC technique that actually corrects the errors.

Voltage scaling reduces energy consumption of circuits at the cost of possible errors [218]. For example, reducing SRAM supply voltage saves leakage energy but also increases probability of read upset (flipping of a bit during read operation) and write failure (writing a wrong bit) [219]. Chippa et al. [220] present a technique that uses approximation at multiple levels of abstraction. For example, for  $k$ -means clustering, at algorithm level, early termination and convergence-based pruning are used. At architecture level, both input and intermediate variables are represented and operated upon with scaled precision. This leaves some bit slices in the data path unutilized which are power-gated for saving energy. At circuit level, voltage overscaling is used, without scaling the clock frequency. The adder circuit is segmented into adders of smaller bit-width. Based on voltage scaling, carry propagation across segmentation

points is adaptively controlled and errors due to ignored carry values are reduced by using a low-cost correction circuit.

### 6.3 Inexact Speculative Adder (ISA)

Among numerous speculative adders[221], the Inexact Speculative Adder (ISA)[200] is a general and optimum architecture of speculative addition to improve speed, power efficiency and accuracy management thanks to a short speculative path and to an adaptable double-direction error compensation mechanism. This technique allows to precisely control mean and maximum errors. It has also shown significant benefits compared or combined with other low-power techniques or successfully integrated within bigger ASIC systems. In the case of FPGA, the ISA could be particularly interesting in order to overcome FPGA's hardware limitations, e.g. fixed number of Look-Up Tables (LUT) and interconnect constraints.

Additions are the most frequent arithmetic units used in digital systems. Hence, many have tried to improve their speed or power efficiency. For this purpose, some approximate adders have been built using the concept of carry speculation [197]. This is feasible as carry chain propagation typically does not cover the entire length of the adder, allowing to guess relatively accurately an internal carry based on a small number of preceding stages. As a result, the carry propagation chain, critical path of the circuit, can be sliced in multiple shorter paths executed in parallel, loosing up delay constraints over the whole circuit and enabling performance beyond theoretical bounds of exact adders.

Among numerous speculative adders[221], the Inexact Speculative Adder (ISA)[200] is a general and optimum architecture of speculative addition to improve speed, power efficiency and accuracy management thanks to a short speculative path and to an adaptable double-direction error compensation mechanism. This technique allows to precisely control mean and maximum errors. It has also shown significant benefits compared or combined with other low-power techniques [222–224] or successfully integrated within bigger ASIC systems[193]. In the case of FPGA, the ISA could be particularly interesting in order to overcome FPGA's hardware limitations, e.g. fixed number of Look-Up Tables (LUT) and interconnect constraints.

Figure 6.1: Block schematic of the Inexact Speculative Adder (ISA) [200]. Every speculative path comprises a carry speculation block (SPEC), an adder block (ADD) and a double-direction error compensation block (COMP).



Figure 6.2: Example of arithmetic computation in an ISA with 4-bit ADD, 2-bit SPEC, 1-bit COMP correction and 2-bit COMP reduction.

The general block schematic of the ISA is presented on Figure 6.1. It slices the carry chain in several speculative sub-paths executed in parallel, each of them consisting of a carry speculation block (SPEC), an addition block (ADD) and an error compensation block (COMP) that overlaps on two ADD blocks. For each of these paths, the functionalities of the blocks are the following:

- The SPEC block produces a speculated internal carry from a very short number of input bits. This is generally done with a carry look-ahead unit. If a carry propagation spans the entire SPEC block, it cannot predict exactly the carry and a wrong guess could lead to a speculative error. Since long propagation sequences are uncommon, the rate of erroneous speculations decreases with larger the SPEC block size.
- The ADD block computes a local sum from the carry speculated in the SPEC block.
- Without compensation, incorrect carry speculations could cause disastrous errors. The COMP block detects those incorrect speculations and compensates erroneous sum either by trying to correct a fixed number of bits in the current sum, or by balancing some bits in the preceding sum to limit its relative arithmetic value.

The achieved adder arithmetic is shown on Figure 6.2. Errors only occur in the speculative paths on the right. The COMP is triggered when the speculated carry differs from the carry-out of the prior sub-adder. The COMP's correction technique implements an incrementer or decrementer on a fixed group of LSBs of the current ADD block that fully corrects a missed carry. This technique fully resolves most speculative errors, as in the central path of Figure 6.2 for which the sum's LSB of the has been corrected. In the cases where the stages above correction bits are all in propagation modes, the sum bits cannot be corrected as it would cause an internal overflow. Thus, the COMP's reduction flips the MSBs of the preceding sum in order to reduce the arithmetic error as in the right path of Fig 6.2.

## 6.4 Inexact Speculative Multiplier (ISM)

Many works have tried to improve speed or power efficiency of adders[200, 225–227]. But the case of the multiplier circuits has been little studied, despite their much higher area, power consumption and delay.

The Error Tolerant Multiplier (ETM) [228], one of the first approximate multipliers, is based on the truncation of multiplier into an accurate multiplication part for the MSBs, and a low-cost non-multiplication part for the LSBs similarly as in the Error Tolerant Adder (ETA) [229]. This part uses OR gates as low-cost addition, and propagating in the right direction, it sets all lower-significance bits to the same value to balance the error if it happens. The Broken-Array Multiplier (BAM) [230] removes some carry-save adders of the array multiplier in both horizontal and vertical directions, improving circuit efficiency, but yet still suffering from high rate and mean of errors.

The Inexact Speculative Multiplier (ISM), a new approximate multiplier circuit derived from error-compensated speculative architectures, is introduced here.

Conventional parallel multiplier architectures are based on computing a set of partial products and summing them together. To be integrated in high-performance blocks such as a FPU, this process is generally pipelined with several stages. The ISM is based on a two-stage multiplier architecture. First, a Partial Product Multiplier generates and merges partial products with a compressor tree into two partial sums. Then, an Inexact Speculative Adder [200] adds them in a speculative way in the last stage. This approach strongly reduces the overall critical path, and with a retiming step, used for instance in the case of pipelining, it significantly relaxes the timing constraints, leading to smaller overall area and power consumption.

Sizing of the different speculative elements of the adder stage directly allows to trade worst-case and average errors in a delay-accuracy approach in the case of unsigned operation, as in [200]. In the case of two's-complement signed multiplication, a dynamic carry guess of the inverse of the expected sign is required on all speculative paths to avoid any sign error (i.e. a XNOR of the two operand's MSBs). Other parameters are selected in the same approach as for unsigned operation.

As the mantissa multiplier is in the critical path of the FPU circuit, even the slightest level of approximation can significantly relax the timing constraints. Moreover, the ISM error compensation and the FPU rounding unit both share the same philosophy that a few bits in one direction are equivalent to a single one at adjacent position. For instance, the FPU rounding would approximate the sequence '0.111' by '1.000', while the speculative error '0.000' instead of '1.000' would be compensated by '0.111'.

The ISM architecture has been studied for the Singular Value Decomposition (SVD) implementation mentioned in Chapter 5. As this work is still under-development, no measured results are available yet.

## 6.5 ISA in Machine Learning

In most robotic and biomedical applications, the interest for real-time embedded systems with tactile ability has been growing. For example in prosthetics, a dedicated portable system is needed for developing wearable devices. The main challenges for such systems are low latency, low power consumption and reduced hardware complexity. In order to improve hardware efficiency and reduce power consumption, approximate computing techniques have been assessed. This strategy is suitable for error-tolerant applications involving a large amount of data to be processed, which perfectly fits tactile data processing.

Considering the Machine Learning (ML) algorithm explained in Chapter 5, the creation of a real-time embedded data processing unit for e-skin is yet far from achievement. This ML approach requires the use of Singular Value Decomposition (SVD) algorithm, which is a computationally intensive[231] process. As one of the SVD architectural implementation has been based on the Coordinate Rotation Digital Computer (CORDIC) algorithm ([232]), the implementation of CORDIC on FPGA uses ISA architectures has been investigated.

### 6.5.1 CORDIC Algorithm

The CORDIC [232] is an iterative and particularly well parallelizable algorithm extensively used in digital signal processing. It only contains iterative Shift-Add operations to calculate a variety of functions, such as logarithmic, trigonometric and hyperbolic functions. It can be operated in vectoring mode or in rotation mode. The first produces a rotation of the input vector to the  $x$  axis while recording the angle needed for that rotation. The second, called *rotation* by Deprettere *et. al* [233], makes a rotation of the input vector by a specified angle. Despite the CORDIC can be operated in both modes, only the latter has been considered in this work.

The CORDIC rotation-mode algorithm starts by initializing the angle accumulator  $z$  with the requested rotation angle  $z_0$ . Then, depending of the sign of the angle after every iteration, a decision  $d_i$  is taken in order to decrease the angle accumulator

Figure 6.3: Architecture of the CORDIC in rotation mode.

magnitude. The equations in rotation mode are:

$$\begin{aligned} z_{i+1} &= z_i - d_i \arctan(2^{-i}) \\ x_{i+1} &= x_i - 2^{-i} d_i y_i \\ y_{i+1} &= y_i - 2^{-i} d_i x_i \end{aligned} \tag{6.1}$$

where (i)  $i = \{0, N - 1\}$ , (ii)  $N$  the number of iterations, and (iii)  $d_i = -1$  if  $z_i < 0$

and  $+1$  otherwise;

which implies that:

$$\begin{aligned} x_n &= A_n (x_0 \cos z_0 - y_0 \sin z_0) \\ y_n &= A_n (x_0 \sin z_0 + y_0 \cos z_0) \\ z_n &= 0 \\ A_n &= \prod \sqrt{1 + 2^{-2i}} \end{aligned} \tag{6.2}$$

where  $A_n$  is a gain depending on the number of iterations.

### 6.5.2 Hardware Implementation

The CORDIC algorithm architecture uses a single Shift-Add operation for each component:  $x$ ,  $y$ , and  $z$ . Each unit consists of a MUX (2:1 multiplexer), a shift register and an adder-subtractor. At the beginning of each CORDIC computation,  $x_0$ ,  $y_0$  and  $z_0$  values are given as inputs to the MUX. Then the computation proceeds using the values stored in  $Xreg$ ,  $Yreg$  and  $Zreg$ , respectively. In the ROM, the micro-rotation angles  $\arctan(2^{-i})$  are stored. The CORDIC algorithm is an iterative process varying according to the ROM input  $i$ . In the considered case, the assigned values of the variable  $i$  are from 0 to 29. To control the ROM addresses, the FSM tracks the shifting distance and enables the multiplexer signals.

In order to apply ISA within the CORDIC architecture, modifications have been needed. The new architecture is illustrated in Figure 6.3. As ISA normally works with unsigned numbers, the adder-subtractors for  $x$ ,  $y$  and  $z$  components have been substituted with conventional adders. Adding a MUX that considers the positive or the negative signal depending on a control signal makes possible to construct a circuit performing both addition and subtraction at the same time. This expedient has been

used to have the right working principles of the CORDIC algorithm as both addition and subtraction operations can occur depending on the sign of  $Zreg$ .

As the CORDIC in rotation mode has just been taken into account only the MUX selecting the angle accumulator  $z$  or the initialization input value  $z_0$ , the desired angle of rotation, has been considered. Figure 6.3 depicts a Xilinx Vivado simulation example underling the independence of  $x_0$  and  $y_0$  on the initial CORDIC operations.

According to (6.1) and (6.2), the results of the rotation-mode CORDIC computation are the cosine  $x_n$  and sine  $y_n$  of the input angle  $z_0$ . In particular, by setting:

$$y_0 = 0 \quad (6.3)$$

the equations in (2) are reduced to:

$$\begin{aligned} x_n &= A_n \cos z_0 \\ y_n &= A_n \sin z_0 \end{aligned} \quad (6.4)$$

and by setting:

$$x_0 = \frac{1}{A_n} \quad (6.5)$$

where  $A_n = 0.6073$ , the rotation produces the unscaled cosine and sine of  $z_0$ .

### 6.5.3 Experimental Results

The proposed CORDIC architecture has been modeled in VHDL and simulated using Xilinx Vivado (Figure 6.3). It has then been synthesized and implemented on a Xilinx ZYNQ-7000 ZC702 device.

This work aims at reducing resource utilizations (e.g. hardware complexity, power consumption, or latency) and analyzing computations accuracy of CORDIC and approximate CORDIC architectures. Although it would be advantageous to optimize the bigger number of speculative paths, this study solely considers limited cases of speculative paths with regular speculative structures (i.e. identical speculative paths). As first validation of the use of speculative arithmetic within an FPGA platform, over a hundred ISA architectures have been considered and eight of them have been selected. Table 6.2 lists the different ISA configurations of the approximate CORDIC implementations. The parameter value choice are based on the considered CORDIC data width of 32 bit.

In order to quantify the computation accuracy of the approximate designs, two metrics have been considered as in[200]. Both are built with the relative error ( $RE$ ), defined as:

| # | ISA configuration details                     |
|---|---|
| 1 | 2 paths, ADD = 16, SPEC = 0, COR = 6, RED = 0 |
| 2 | 2 paths, ADD = 16, SPEC = 1, COR = 2, RED = 4 |
| 3 | 2 paths, ADD = 16, SPEC = 2, COR = 1, RED = 3 |
| 4 | 2 paths, ADD = 16, SPEC = 3, COR = 0, RED = 2 |
| 5 | 4 paths, ADD = 8, SPEC = 0, COR = 1, RED = 2  |
| 6 | 4 paths, ADD = 8, SPEC = 1, COR = 3, RED = 0  |
| 7 | 4 paths, ADD = 8, SPEC = 2, COR = 5, RED = 0  |
| 8 | 4 paths, ADD = 8, SPEC = 3, COR = 0, RED = 0  |

Table 6.2: ISA configurations of the approximate CORDIC implementations

$$RE = \left| \frac{v_{approx} - v_{correct}}{v_{correct}} \right| \quad (6.6)$$

where  $v_{approx}$  and  $v_{correct}$  are the approximate and correct values of CORDIC computation, respectively. The two metrics used are the Root Mean Square (RMS) of the relative error ( $RE_{RMS}$ ), which is a well-known accuracy estimator, and the maximum relative error ( $RE_{MAX}$ ), that defines the worst-case accuracy.

Figure 6.4 shows the error characteristics and normalized costs of each approximate CORDIC implementation for sine and cosine outputs. Hardware costs, normalized to the conventional FPGA design, are expressed in terms of dynamic power and Power-Delay-Area Product (PDAP). Despite different  $RE_{RMS}$  and  $RE_{MAX}$  values between sine and cosine computing, the error characteristics follow the same trends. The persistent gap between cosine and sine errors is expected to come from the stimuli angles, ranging from 0 to 45° as required by the tactile data processing application.

Figure 6.4: Normalized costs and relative errors for both sine and cosine computations of the approximate CORDIC implementations.

Approximate circuits in the CORDIC design allows a dynamic power consumption saving up to 40% and a general improvement of PDAP of 58% at the cost of low errors with  $RE_{RMS}$  of 0.049% for sine and 0.003% for cosine computations. Though, some designs, as #5 and #6 display both bad accuracy and hardware characteristics, showing that small 8-bit ADD combined with low SPEC size does not provide a good enough addition for the convergence of the CORDIC. Configurations with 16-bit ADD blocks have lower circuit costs than 8-bit ones, this result contradicts the ASIC

intuitive results in [200], this is due to FPGA’s fixed LUT architecture. In effect, some ISA architectures might fit better the LUT implementations and interconnections to minimize their required number or delay.

## 6.6 Conclusion

Approximate computing has become a major field of research in the sense that it could significantly improve energy efficiency and performances of modern digital circuit. To date, many approximate techniques have been presented in the literature.

Inexact and approximate circuit design is a radical approach to trade this counterproductive quest for perfection for substantial gains in power, speed, and area. The primary challenge, however, is to determine *where* and *how* to let an error or an approximation occur in the circuits without compromising the functionality or the user experience. With ever-increasing amount of data being processed, a wide variety of applications could tolerate inaccuracies. For example, in multimedia processing, a small proportion of errors is not perceptible to humans, and in highly computational algorithms such as data mining, search or recognition, the outcome is not required to be a single result but an adequate match. A promising approach to design inexact circuits is to use speculation to trade circuit accuracy for better power and speed. Taking advantage of such circuits would help to realize extremely energy-efficient and high-performance DSPs and hardware accelerators at lower integration cost and with higher speed, data rate or duty-cycling.

This chapter has proposed a first attempt of approximate Coordinate Rotation Digital Computer (CORDIC) implementation on FPGA using Inexact Speculative Adders (ISA). The use of speculative arithmetic has allowed high performance and efficiency improvements of the CORDIC module, with up to 40 % power consumption reduction and up to 21 % delay reduction, offering overall cost reduction of up to 58 %. The approximate CORDIC has been characterized by its relative arithmetic error, showing negligible average and maximal errors, i.e., RMS relative errors being only 0.003 % for cosine computations and 0.049 % for sine computations.

This first FPGA implementation of approximate CORDIC, sub-task of the computationally expensive SVD required in tactile data processing, represents a successful preliminary investigation of approximate computing for real-time embedded prosthetics.

# Chapter 7

## Approximate Arithmetics for Hardware Implementation

### 7.1 Introduction

Before presenting the results achieved by implementing an approximate computing technique called Tunable Floating-Point (TFP), this technique is briefly outlined. The first section contextualize the general framework in which the herein presented project was developed. Furthermore, it better motivates the constraints and requirements of the hardware implementation.

In this chapter, two different algorithms have been exploited to evaluate the novel Tunable Floating-Point, reducing power consumption while still producing acceptable accuracy of results both algorithms.

The results presented in this chapter were published in [234, 235]. The candidate contributed to the published work by:

- Implementing the TFP technique in the tested algorithms at software and hardware level.
- Analyzing data.
- Considerations and conclusions.

In the last years, approximate computing has been undergoing a rapid growth as a promising approach to energy-efficient design of digital systems in a wide spectrum of domains ranging from digital signal processing, to robotics and machine learning. By relying on the ability of many systems and applications to tolerate some loss of accuracy, approximate computing techniques achieve improved energy efficiency. Approximate computing encompasses a broad spectrum of techniques that relax accuracy to improve



efficiency: both at hardware and software level (e.g. skipping computations, voltage overscaling, loop perforation, and quality-scalable or approximate circuits [201,208]). Moreover, there are some studies that combine multiple approximation techniques[236].

Machine Learning (ML) is arguably the hottest application field for arithmetic processors. ML algorithms execute a large number of operations, predominantly multiplications and additions, and require dedicated hardware to accelerate the execution.

Due to the huge size of the datasets to process in ML, the processing time and the energy necessary is very large [237]. To increase the power efficiency, one idea is that the computation migrates from double-precision (*binary64* in IEEE 754- 2008 standard [238]) to single (*binary32*) and half (*binary16*) precision. In [239], *Nurvitadhi et al.* report the precision scaling trend is particularly suitable for FPGA-based accelerators. A big advantage of FPGA-based accelerators is that the hardware can be tailored exactly to match the requirements of the application.

Previously, Nannarelli implemented multi-precision multipliers to increase the power efficiency of on-chip accelerators [240]. Then, the author addressed the design of an on-chip accelerator with Tunable Floating-Point (TFP) precision in [241]. That is, the precision of operands and results can be chosen for a single operation by selecting a specific number of bits for significand and exponent in the floating-point representation.

Numerous studies also report that employing different data formats in Deep Neural Networks (DNNs), the dominant Machine Learning approach, could allow substantial improvements in power efficiency considering an acceptable quality for results. As high classification accuracy comes at expenses of significant computation cost (area/time/energy) in DNNs [237], it may be advantageous to trade computational throughput for accuracy/quality. For example, in image processing, due to the limited perceptual capability of humans, occasional errors such as dropping a particular frame or a small image quality loss often rarely affect a users satisfaction. The precision and the data format requirements depend on the computation phases/workloads in DNNs. The training phase presents numerical challenges as it typically contains millions of parameters that are usually trained iteratively over a vast amount of data. In contrast, in the inference phase, input data have to go only once through the network and the required precision is usually lower than in the training phase [237].

Traditionally, neural networks (NNs) are trained in double or single-precision, a common practice in general scientific computing. However, to reduce the execution time (especially in memory transfers) and to increase the energy efficiency, the computation

is migrated from double-precision (*binary64*: 53-bit significand<sup>1</sup>, 11-bit exponent) in IEEE 754-2008 standard [238], to single (*binary32*: 24-bit significand, 8-bit exponent) and half (*binary16*: 11-bit significand, 5-bit exponent) precision.

Recent developments suggest that more efficient data formats could allow savings in power consumption at the cost of reduced precision in the results. Google has already made its way into production hardware with the Tensor Processing Unit (TPU) [242] designed for inference. The TPU supports the Brain-FP format consisting of 8-bit for the significand and 8-bit for the exponent. Moreover, some studies show that it is possible to achieve dramatic reductions in bit width from 32-bit all the way down to one bit (i.e., binary networks [243]).

Nvidia included Tensor cores specifically designed for ML in their latest generation of GPUs [244]. Each Tensor core can perform fully parallel  $4 \times 4$  matrix multiplication on *binary16* operands to produce *binary32* results.

Intel introduced the “Flexpoint” format for deep learning with the aim to replace the training done in *binary32*. The Flexpoint format is a blocked fixed-point format for tensors (matrices) consisting in a block of 16-bit significands sharing a 5-bit exponent [245].

## 7.2 Motivation

To thoroughly comprehend the advantages of the TFP precision, an example reported in [241] has been considered.

Nannarelli illustrated that the TFP unit is more power efficient than the *binary32* one, for multiple operations in reduced significand precision ( $m < 24$ ).

The experiment consisted in testing single and multiple multiplications. The operands exponent,  $e$ , has always been set to 8, instead of the operands significand  $m$  which was equal to 24, 20, 16, 14, 11, 8, 6. In particular, the test cases were:

- 1) Multiplication  $Z = X \times Y$ .
- 2) Multiplications (two)  $Z = (X \times Y) \times W$ .

The results for the total average power dissipation are reported in Table 7.1 and the trends are shown in Figure 7.1. The performance of the TFP-mult is almost the same for the two test cases, the plots are almost completely overlapped in Figure 7.1 (squares).

---

<sup>1</sup>It includes the integer bit. Significands are normalized in  $[1.0, 2.0)$  in IEEE 754.

| $P_{ave}$<br>$m$ | $Z = X \times Y$ |          |       | $Z = (X \times Y) \times W$ |          |             |
|------------------|------------------|----------|-------|-----------------------------|----------|-------------|
|                  | TFP-mult         | b32-mult | ratio | TFP-mult                    | b32-mult | ratio       |
| 24               | 15.51            | 14.24    | 1.09  | 15.72                       | 14.42    | 1.09        |
| 20               | 14.19            | 12.89    | 1.10  | 14.26                       | 16.63    | 1.05        |
| 16               | 11.14            | 10.26    | 1.09  | 11.10                       | 11.86    | <b>0.94</b> |
| 14               | 10.17            | 9.42     | 1.08  | 10.09                       | 11.26    | <b>0.90</b> |
| 11               | 8.58             | 7.91     | 1.08  | 8.54                        | 10.28    | <b>0.83</b> |
| 8                | 6.05             | 5.67     | 1.07  | 5.98                        | 8.52     | <b>0.70</b> |
| 6                | 5.08             | 4.72     | 1.08  | 5.07                        | 7.36     | <b>0.69</b> |

Table 7.1: Average power dissipation for TFP-mult and b32-mult (RTN) [241].

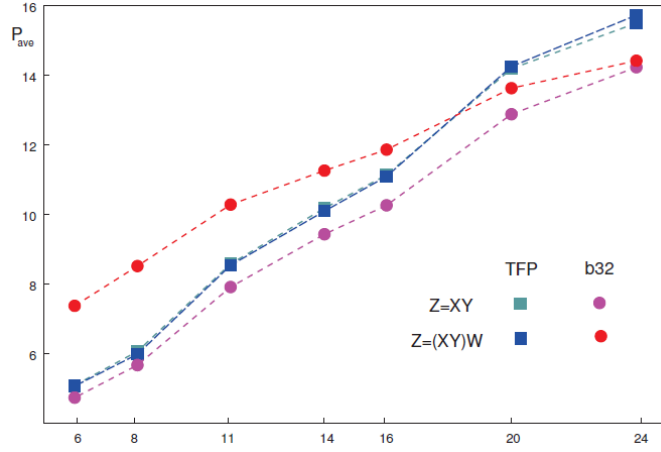


Figure 7.1: Trends of average power dissipation for comparison TFP-mult vs. b32-mult [241].

The trends show that if the number of multiplication increase the average power dissipation drastically drops decreasing the significand number bit for the multiplication operands. This behaviour begins to be distinct by the cross-over of the red/blue curves in Figure 7.1 at  $m \simeq 18$ , and by the ratios  $< 1.0$  in Figure 7.1. Therefore, implementing the TFP precision into very complex algorithms can make the computations more power efficient.

### 7.3 Tunable Floating-Point

The floating-point representation of a real number  $x$  is

$$x = (-1)^{S_x} \cdot M_x \cdot b^{E_x} \quad x \in \mathcal{R} \quad (7.1)$$

where  $S_x$  is the sign,  $M_x$  is the significand or mantissa (represented by  $m$  bits),  $b$  is the base ( $b = 2$  in the following), and  $E_x$  is the exponent (represented by  $e$  bits). The representation in the IEEE 745-2008 standard [238] has its significand normalized  $1.0 \leq M_x < 2.0$  and biased exponent:  $bias = 2^{e-1} - 1$ .

The dynamic range, defined as the ratio between the largest and the smallest (non-zero and positive) number [246], for binary floating-point (BFP) is defined as:

$$DR_{BFP} = (2^m - 1) \cdot 2^{2^e - 1}. \quad (7.2)$$

For example, for *binary32* (single precision,  $m = 24$ ,  $e = 8$ )

$$DR_{b32} = (2^{24} - 1)2^{2^8 - 1} \approx 9.7 \times 10^{83} \quad (7.3)$$

which is much larger than the dynamic range of the 32-bit fixed-point representation:  $DR_{FXP} = 2^{32} - 1 \approx 4.3 \times 10^9$  [246].

The Tunable Floating-Point (TFP) representation is a floating-point (FP) representation with arbitrary number of significand's and exponent's bits:  $m$  and  $e$ , respectively. Dynamic ranges are only considered from and below the *binary32* representation. These ranges are suitable for signal processing and ML applications. We support significand's bit-width from 24 to 4 and exponent's bit-width from 8 to 5. Table 7.2 shows the dynamic ranges for some TFP cases.

| $m$ | $e$ | $DR$                 | storage bits* | comment         |
|-----|-----|----------------------|---------------|-----------------|
| 24  | 8   | $9.7 \times 10^{83}$ | 32            | <i>binary32</i> |
| 11  | 8   | $1.2 \times 10^{80}$ | 19            |                 |
| 4   | 8   | $8.7 \times 10^{77}$ | 12            |                 |
| 24  | 5   | $3.6 \times 10^{16}$ | 29            | <i>binary16</i> |
| 11  | 5   | $4.4 \times 10^{12}$ | 16            |                 |
| 4   | 5   | $3.2 \times 10^{10}$ | 9             |                 |
| 8   | 8   | $1.5 \times 10^{79}$ | 16            | Brain-FP [242]  |
| 16  | 5   | $1.4 \times 10^{14}$ | 16            | Flexpoint [245] |
| 16  |     | 65,536               | 16            | 16-bit FXP      |

\* It includes the sign bit.

Table 7.2: Dynamic range for some FP formats.

By comparing TFP with FXP ranges in Table 7.2, the main advantage of FP over the fixed-point (FXP) representation is that the dynamic range is much larger than

the FXP for similar bits of storage. This is true especially for multiplication where the dynamic range increases quadratically. As for the significand's precision, the optimal bit-width depends on the application.

The TFP representation is normalized to have the conversions compatible with the IEEE 754-2008 standard [238]. Therefore, the implicit (integer) bit is not stored. Subnormals are flushed-to-zero in TFP. numbers when the exponent is less than  $-(E_{max} - 1)$ .

As for the rounding, TFP supports three types:

- **RTZ** Round-to-zero (truncation);
- **RTN** Round-to-the-nearest where half *ulp* (unit in the last position) is always added before the rounding.
- **RTNE** Round-to-the-nearest-even (on a tie) which is the default mode *roundTiesToEven* in IEEE 754-2008.

## 7.4 TFP Units

The algorithms and implementations for TFP addition and multiplication are not too different from those for FP operations. The main challenge to implement the adder and multiplier in TFP is to perform the rounding in a variable position.

For FP operations rounding, few additional fractional bits, called *guard bits*, are necessary to perform the different rounding modes. In particular, the guard bits are labeled as: *G* for guard, *R* for round, and *T* for sticky. These additional bits are arranged in the significand after the last bit (LSB), *L*, as Figure 7.2 shows. Considering FP operations (Figure 7.3), using the **RTZ** allows to truncate a value at bit *L*. For **RTNE**, after adding 1 in position *G* the correction of the bit in position *L* is needed if there is a tie. Namely,  $L = 0$  if  $G(R + T)' = 1$ . Implementing the **RTNE**, first the round value *rnd* is evaluated as  $rnd = G(L + R + T)$ , and then the *rnd* bit is added to the representation of *X*.

The main challenge to implement the adder and multiplier in TFP is that the rounding must be done in a variable position, namely extracting the *L*, *G*, *R* and *T* in variable positions. To do so, additional hardware has to be used. A bit-vector called Rounding Word (RW), in which all bits are zero except the one in the rounding position, is exploited. A simple decoder generate RW depending on *m*. In Figure 7.4 two example scenarios for *m*=24 (*binary32*) and *m*=21 are illustrated. The figure shows the result *S* and the rounding position (bit of weight  $2^{24}$  for *m*=24) marked by

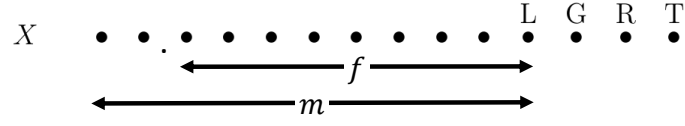


Figure 7.2: Representation of the significand  $m$ :  $f$  fractional bits; L last bit; G guard bit; R round bit; T sticky bit.

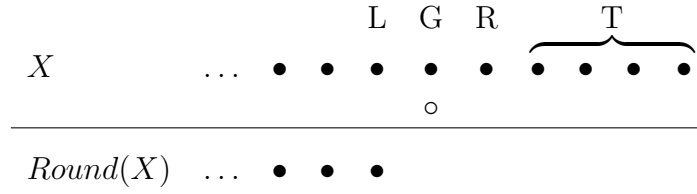


Figure 7.3: FP Rounding:  $\circ$  rounding position.

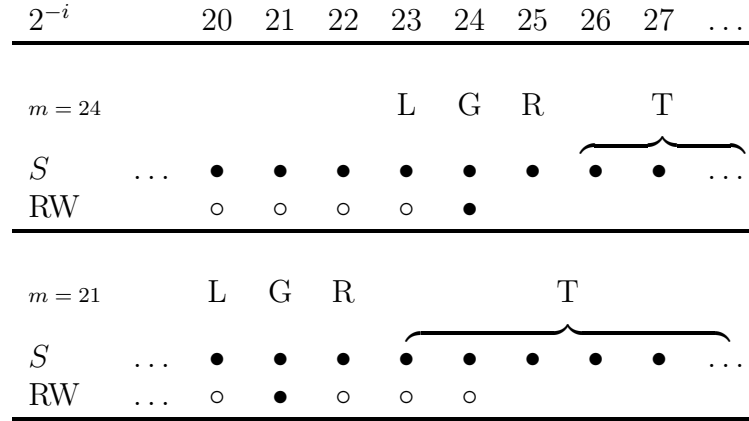


Figure 7.4: Rounding in variable position for  $m = 24$  and  $m = 21$ . RW holds the position of the rounding bit, marked as •

• in RW. Besides, a mask that is generated by the decoder has also to be added in the m-MSB to zero the  $m$  LSBs of the product. This mask can be generated by the decoder. For example for  $m = 11$ :

RW: 0000 0000 0001 0000 0000 0000

MASK: 1111 1111 1110 0000 0000 0000

More details are given in [241].

### 7.4.1 TFP Multiplier

The architecture of the TFP multiplier (TFP-mul) is sketched in Figure 7.5. The significand and exponent bit-widths  $m(= 5)$  and  $e(= 2)$  can be selected for the single operation by setting a 7-bit value in a control register.

The significand path consists of a radix-4 multiplier array with product in carry-save format  $(P_s, P_c)$  followed by rounding and normalization blocks. The rounding is performed speculatively and in a variable position by selecting a rounding word (RW) depending on the precision  $m$  required. The exponent path is depicted at left in Figure 7.5. More details are given in [241].

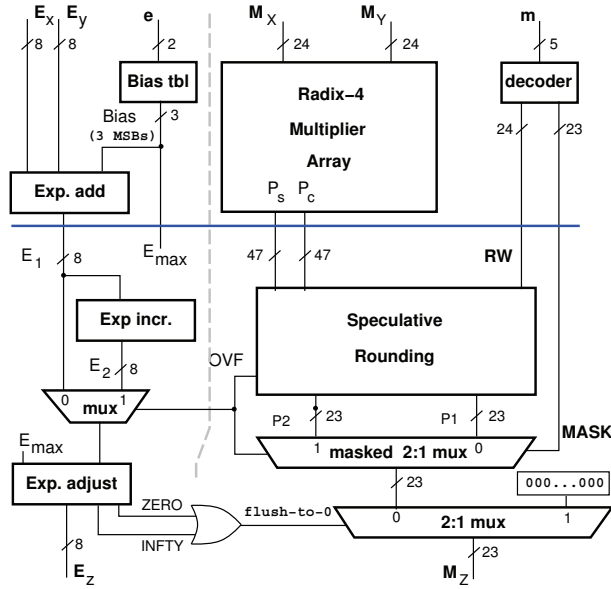


Figure 7.5: Architecture of TFP-mul.

### 7.4.2 TFP Adder

The architecture of the TFP adder (TFP-add) is derived from the “double-path” scheme of [246], and it is sketched for the significand path in Figure 7.6. Depending on the effective operation (addition or subtraction) and the exponent difference the operation is performed either in the “close” (at left) or the “far” (at right) path. Similarly to the TFP-mul, a decoder provides the rounding word and the mask to implement the TFP operation.

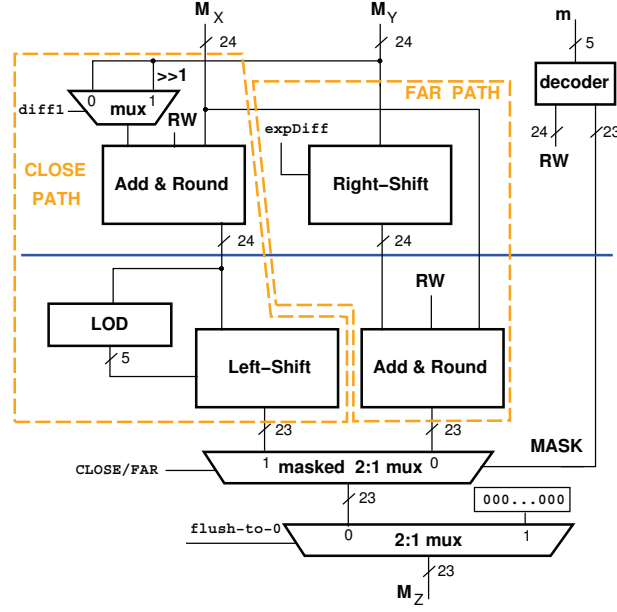


Figure 7.6: Architecture of TFP-add (significand only).

### 7.4.3 Hardware Implementation

For the implementation of the TFP multiplier and adder a 45 nm CMOS library of standard cells by using commercial synthesis and place-and-route tools (Synopsys) has been chosen.

A target throughput of 1 GFLOPS for the single TFP multiplier and TFP adder has been set. To reach this target, the architecture of Figure 7.5 and Figure 7.6 must be pipelined in two stages, with pipeline registers indicated by the horizontal blue lines in the corresponding figures.

For the power dissipation, traces (TFP vectors) have been generated from the simulator (see Section 7.5) by extracting the actual operands multiplied or added in the tested algorithms, and post-layout simulations through Synopsys VCS has been run for several cases.

## 7.5 TFP Simulation

Implementing a simulator to profile the applications and determine the acceptable precision and dynamic range was a fundamental step. A first version of the simulator consists in a library of C functions implementing TFP operations. The supported operations are: addition/subtraction, multiplication, division and square root. Each



operation is implemented with a standard FP algorithm by limiting the computation of the significand bits to  $m$  and by applying the specified rounding mode. However, all operands and results are rendered in double-precision (*double*) in C. Therefore, it is necessary to trap the cases when the unbiased 11-bit exponent exceeds the maximum  $e$ -bit exponent generating an infinity exception, and flushing the *double* to zero, when the unbiased 11-bit exponent is less than the minimum  $e$ -bit exponent. In the first version of the simulator, the algorithm under test is coded in the C main program by invoking the TFP operations with arbitrary  $m$ ,  $e$ . Each operation may have a different  $m$ ,  $e$ , i.e., precision and dynamic range can be changed in different parts of the algorithm. The simulator also executes the algorithm under test in *double* and provides the error in key points. Moreover, by setting a debug option, the error can be displayed for each TFP operation. Moreover, the simulator generates TFP vectors (either in format *binary32* or *binary16*) to be used to test the hardware implementation.

## 7.6 TFP for Deep Learning

Deep Neural Networks has emerged as the dominant Machine Learning (ML) algorithm showing remarkable success in many challenging application domains as image processing, speech recognition, and machine translation[247, 248].

Neural networks (NN) can be considered a simulation of the human nervous system. The human nervous system is composed of cells, referred to as neurons. Biological neurons are connected to one another at contact points, which are referred to as synapses. Learning is performed in living organisms by changing the strength of synaptic connections between neurons. Typically, the strength of these connections change in response to external stimuli. As in the case of biological networks, the individual nodes in artificial NN are referred to as neurons. These neurons are units of computation that receive input from some other neurons, make computations on these inputs, and feed them into yet other neurons. The computation function, called activation function, at a neuron is defined by the weights on the input connections to that neuron with an additional bias. The NN is also defined as a mathematical model that consists in a set of layers: input, hidden and output layers. An input layer represents a training or inference data set, a hidden layer generates computations and transfers information to the output layer that produces the results. If a NN consists in many hidden layers, it is called Deep Neural Network (DNN).

These models typically contain a very large number of parameters (weights  $w_i$  and bias terms  $b_i$ ) and are usually trained iteratively over vast amounts of data. The NN

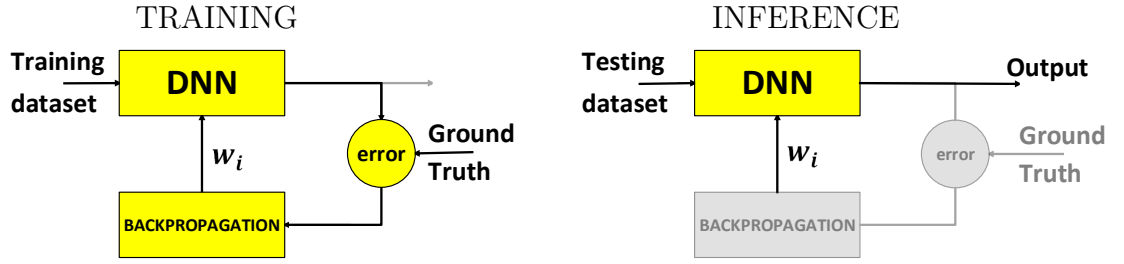


Figure 7.7: Neural network structure.

structure can be represented as in Figure 7.7. The training phase consists of feeding data to the network, forward propagating through the whole network, estimating whenever incorrect predictions are made by comparing predictions with *ground truth* (i.e. a target for the NN), computing and back-propagating weights through the network to minimize the error of incorrect predictions. At the beginning the error (defined as cost function) has a high value, then the neural network training adjusts the parameters of the neural network in order to attempt to minimize the value of the cost function step by step. The gradient descent algorithm has been chosen to reduce the error value. The gradient descent is an efficient optimization algorithm that attempts to find a local or global minima of a function and, in NN, enables a model to learn the gradient or direction that the model should take in order to reduce differences between actual  $Y$  and predicted  $\hat{Y}$  (Figure 7.8).

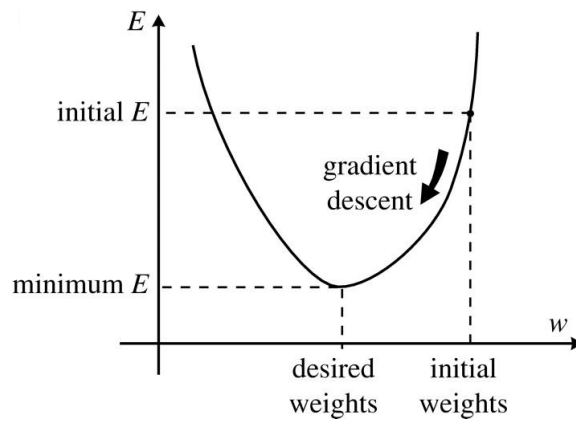


Figure 7.8: Gradient Descent Algorithm in Neural Network.

An epoch is defined as a full pass over the entire training data set. While the NN training can require hundreds of epochs before reaching the final parameter values, the inference phase consists of a single pass over the entire network. After training, the optimal parameter values are computed and the model is ready to classify new input data.

Figure 7.7 represents as the first stage in a NN starts by training a model to learn to predict outputs from some inputs, and then as the second stage deploys that model to predict outputs from a novel input data set.

Hence by changing these weights appropriately, the computation function can be learned, which is analogous to the learning of the synaptic strength in biological neural networks. The "external stimulus" in neural networks for learning these weights is provided by the training data. The key to the effectiveness of the neural network is the architecture used to arrange the connections among nodes. A wide variety of architectures exist, starting from a simple single-layer perceptron to complex multi layer networks. Neural Networks are networks of several interconnected units with a simple behaviour, most often without memory and capable of a simple input-to-output transformation.

The function of a node that defines the output of that node, or "neuron," given an input or set of inputs has called activation function. In NN, the most common non-linear activation functions are: (i) step function, (ii) logistic or sigmoid, and (iii) the Rectified Linear Unit (ReLU). Figure 7.9 schematically illustrates these activation functions. Nowadays, ReLU is very popular as it allows a easier learning for a NN. For this reason, ReLU has been used as activation function in part of the study that have been conducted.

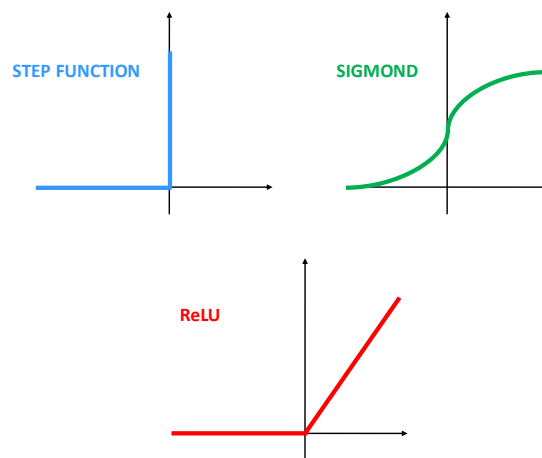


Figure 7.9: Example of activation functions.

### 7.6.1 Training in TFP

Figure 7.10 shows the architecture for the two-hidden-layers NN, which has been considered as reference example to illustrate the properties of TFP. Moreover, a data set corresponding to a cosine-trend curve with 200 points have been chosen (see Figure 7.11). The goal is to interpolate the function approximating the distribution of the points.

The conducted study shows the approximation errors, that depend on selecting a specific precision for the significand in the floating-point representation, for the training and inference stage of a NN. Results for NN training are presented.

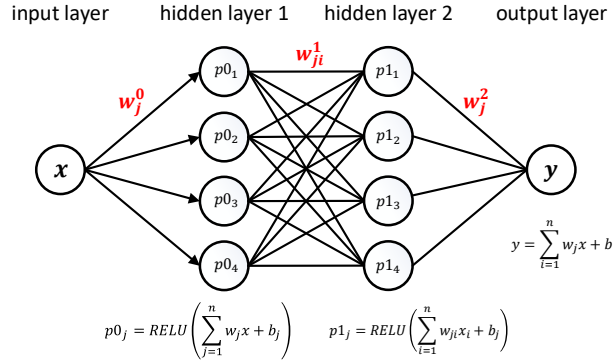


Figure 7.10: Neural network with two hidden layers (depth=2).

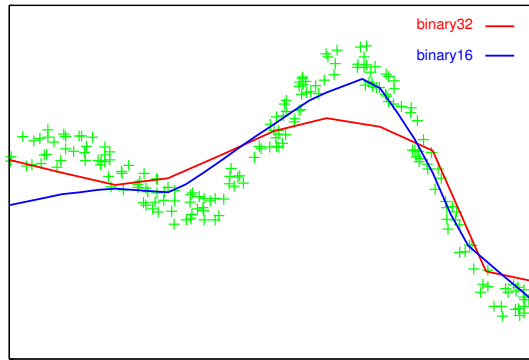


Figure 7.11: Training: interpolated functions by NN of Figure 7.10.

Table 7.3 reports the training for the test case *cosine* for several TFP precisions. The table lists the approximation relative error ( $\epsilon_{ave}$ ) obtained at the given epoch,

| $m$           | $e$ | $\epsilon_{ave}$ | epoch | $n_{op}$ | $P_{add}$                         | $P_{mul}$ | $P_{tot}$ | ratio |
|---------------|-----|------------------|-------|----------|-----------------------------------|-----------|-----------|-------|
| 24            | 8   | 0.13             | 212   | 6,127    | 5.84                              | 13.99     | 19.83     | 1.00  |
| 20            | 8   | 0.13             | 229   | 6,618    | 5.59                              | 12.35     | 17.94     | 0.90  |
| 16            | 8   | 0.13             | 214   | 6,188    | 5.24                              | 10.44     | 15.68     | 0.79  |
| 14            | 8   | 0.19             | 5     | 145      | 5.02                              | 9.68      | 14.70     | 0.74  |
| 11            | 5   | 0.12             | 5     | 145      | 4.70                              | 8.77      | 13.47     | 0.68  |
| 9             | 5   | 0.27             | 9     | 258      | 4.48                              | 7.69      | 12.17     | 0.61  |
| 7             | 5   | 0.27             | 4     | 115      | 4.27                              | 7.07      | 11.34     | 0.57  |
| 5             | 5   | 0.27             | 3     | 86       | 3.99                              | 6.13      | 10.12     | 0.51  |
| $\times 10^3$ |     |                  |       |          | $P_{ave}$ [mW] measured at 1 GHz. |           |           |       |

Table 7.3: Average error and average power dissipation for TFP training.

the number of the TFP operations executed, the average power dissipation (at 1 GHz for addition, multiplication and total), and the ratio among the total dissipated power for all the considered TFP precisions. The  $\epsilon_{ave}$  has been defined as:

$$|\hat{v} - v| \quad (7.4)$$

where  $\hat{v}$  and  $v$  are the approximate and *binary64* values of the NN computation, respectively.

The trends in Table 7.3 show that the power dissipation drops linearly as  $m$  is scaled. Scaling  $m$  in the NN allows to achieve a good power efficiency as a reduction up to 30 % and 50 % is reached comparing  $m=24$  with  $m=11$  and  $m=5$ , respectively. Table 7.3 also shows that the operands precision has an impact on the convergence rate. For *cosine* training, the *binary32* smallest error is obtained for a large epoch. From  $m = 14$  the NN converges very rapidly at epoch 5, and for *binary16* we obtain the lowest error. Figure 7.11 shows the curves approximated for the case of *binary32* (212 epochs) and *binary16* (5 epochs). The reason for this behavior is that, when  $m$  and  $e$  are reduced, small numbers are flushed to zero causing a sort of pruning in the NN. A lower precision may lead to faster convergence, but also to an excessive pruning resulting in the NN not converging.

### 7.6.2 Error Characterization

The neural network of Figure 7.10 is used to interpolate the function describing the distribution of the training points. Since for the test case of Figure 7.11 the function generating the training points is known, the error can easily be evaluated.

The **approximation error** is the error due to approximation done by the NN. Referring to Figure 7.12, assume there are a number of points (dark “+” in the

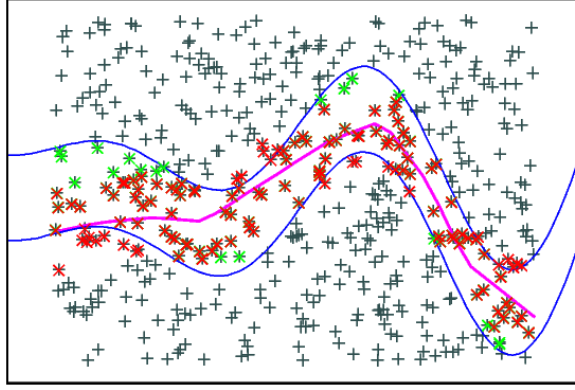


Figure 7.12: Approximation error for NN interpolated function.

figure) and we want to detect which ones are within a given distance from the ideal function/curve. This region is delimited in Figure 7.12 by the blue lines, and the points which fall in the region in the ideal case of the generating function are marked with a green “\*” (120 points).

However, when repeating the inference with the NN interpolated function (curve in magenta), a different set of points, marked with a red “\*” in Figure 7.12, lies within the given distance (129 points). By analyzing the results of the experiment, the points correctly identified by the NN lying in the region are 97, i.e., the intersection of the green and red sets. The miss-classified points are:

- 23 green \* which are inside the region, but not detected by the NN;
- 32 red \* which are detected inside the region by the NN, but they are actually outside.

In addition to the approximation error, there is also the **quantization error** due to the reduction in the precision of the operations. By repeating the experiment of points lying in the given region for different TFP precisions, the obtained results are reported in Table 7.4. In this case the reference set is the one obtained for inference with *binary32* precision.

In Table 7.4, the column marked  $(* \cap *)$  shows the points detected correctly for the given precision with respect to *binary32*; the column marked  $(*)$  reports the points lying in the region for *binary32*, but not detected in the given precision; the column marked  $(*)$ , vice versa, reports the points detected in the region for the given precision, but which are outside the region for *binary32*.

| $m$                   | $e$                  |                   | $Np_{TOT}$ | $(\textcolor{red}{*} \cap *)$ | $(\textcolor{red}{*})$ | $(*)$ |
|-----------------------|----------------------|-------------------|------------|-------------------------------|------------------------|-------|
| $\textcolor{red}{24}$ | $\textcolor{red}{8}$ | <i>binary32</i>   | 129        | —                             | —                      | —     |
| 20                    | 5                    | 129               | 129        | 0                             | 0                      |       |
| 11                    | 5                    | <i>binary16</i>   | 128        | 127                           | 2                      | 1     |
| 9                     | 5                    |                   | 129        | 127                           | 2                      | 2     |
| 8                     | 8                    | <i>Google BFP</i> | 131        | 124                           | 5                      | 7     |
| 8                     | 5                    | 131               | 124        | 5                             | 7                      |       |
| 7                     | 5                    | 132               | 121        | 8                             | 11                     |       |
| 6                     | 5                    | 129               | 116        | 13                            | 13                     |       |
| 5                     | 5                    | 129               | 97         | 32                            | 32                     |       |

Table 7.4: Quantization error for different TFP precisions.

The results of the error evaluation done for this specific example, show that even a small variation in precision may lead to a sizeable increase in the overall classification error. Therefore, finely tuning the precision, as in TFP, may significantly improve the results of NN inference with respect to fixed-precision formats.

### 7.6.3 Advantages of TFP for Deep Learning

The novelty of TFP format is its flexibility. To prove the flexibility advantage studies have been conducted on the inference phase as the inference usually requires lower precision than the training phase [237].

A NN with two hidden layers needs three different parameter levels (0, 1, 2). Consequently, weights and bias terms has been divided in  $w_j^0$ ,  $w_{ji}^1$ ,  $w_j^2$  and  $b_j^0$ ,  $b_j^1$ ,  $b^2$  (Figure 7.10). The precision of parameters has been reduced depending on the level belonging to. Each parameter level has been varied by selecting a specific number of bits for significand among 16 or 8 fixing  $e = 8$  in the floating-point representation. The flexibility of the employed TFP format for significand has been reported in the Table 7.5 together with results.

Results obtained by using *binary32* representation have been also included for comparing with the chosen TFP precision. Table 7.5 depicts the quantization error with respect the different TFP precisions, and also shows which parameters configurations can be more power efficient.

The maximum results quality and minimum dissipated power are achieved when  $m^0, m^1, m^2 = 16, 8, 8$ . These precisions allow to reduce power dissipation by 30% (ratio = 0.70) and to detect correctly 98% (127/129) points respect to *binary32*. Instead, the largest power reduction (40%, ratio= 0.60) is reached by exploiting the lowest

| $m^0$ | $m^1$ | $m^2$ | $Np_{TOT}$ | $(* \cap *)$ | $(*)$ | $(*)$ | $P_{add}$                | $P_{mul}$ | $P_{tot}$ | ratio |
|-------|-------|-------|------------|--------------|-------|-------|--------------------------|-----------|-----------|-------|
| 24    | 24    | 24    | 129        | 129          | —     | —     | 6.02                     | 14.80     | 20.82     | 1.00  |
| 16    | 16    | 16    | 129        | 125          | 4     | 2     | 5.31                     | 11.32     | 16.63     | 0.80  |
| 16    | 16    | 8     | 127        | 125          | 4     | 2     | 4.83                     | 10.37     | 15.20     | 0.73  |
| 16    | 8     | 16    | 127        | 125          | 4     | 2     | 4.94                     | 10.91     | 15.85     | 0.76  |
| 16    | 8     | 8     | 129        | 127          | 2     | 2     | 4.52                     | 9.99      | 14.51     | 0.70  |
| 8     | 16    | 16    | 126        | 123          | 6     | 3     | 5.16                     | 9.68      | 14.84     | 0.71  |
| 8     | 16    | 8     | 125        | 120          | 9     | 5     | 4.68                     | 8.59      | 13.27     | 0.64  |
| 8     | 8     | 16    | 126        | 123          | 6     | 3     | 4.75                     | 9.20      | 13.95     | 0.67  |
| 8     | 8     | 8     | 128        | 123          | 6     | 5     | 4.36                     | 8.21      | 12.57     | 0.60  |
|       |       |       |            |              |       |       | $P_{ave}$ [mW] at 1 GHz. |           |           |       |

Table 7.5: Quantization error and average power dissipation for flexible inference ( $e = 8$ ).

precisions (i.e.  $m^0, m^1, m^2 = 8, 8, 8$ ). For the given precisions only the 95% (123/129) points respect to *binary32* are correctly detected.

## 7.7 TFP for Embedded Machine Learning

The current challenge of the development of an embedded and real-time system for Machine Learning (ML) data processing relies on the efficient implementation and power requirements. An example would be prosthetic applications. ML paradigms [55, 177, 179, 181] have been used to retrieve information about object contacts as they are powerful methods for tackling clustering, regression or classification problems. ML algorithms execute a large number of operations, mainly multiplications which are the most power demanding arithmetic operations. In order to increase the power efficiency in processing ML algorithm, new design dimensions and tradeoffs need to be explored. In this context, many quality-scalable [201] or approximate circuits [227] have been presented in the recent years.

This section presents the implementation of the TFP representation [249] in the Singular-Value Decomposition (SVD) algorithm based on One-Sided Jacobi, as SVD is the most computationally expensive algorithm for ML approaches that has ever been considered [199] for real-time tactile data processing.

### 7.7.1 SVD Algorithm

The singular-value decomposition (SVD) factorizes a real  $m \times n$  matrix  $M$  into a product with the form:

$$M = USV^T \quad (7.5)$$



---

**Algorithm 1** Matrix M Symmetrization Algorithm.

---

```
/*If  $M$  satisfies condition  $nrow \neq ncol \rightarrow M_{symm} = M^T * M^*$ */  
for i=1 to ncol do  
  for ii=i+1 to ncol do  
    for j=i+1 to nrow do  
      tmp = tmp + M[j][i]*M[j][ii];  
    end for  
     $M_{symm}[i][ii] = tmp$ ;  
    tmp = 0;  
  end for  
end for
```

---

---

**Algorithm 2** Jacobi Rotation Algorithm.

---

```
for i=1 to i < j do  
  ...  
  for k=1 to ncol do  
    t =  $A_{ki}$ ;  
     $A_{ki} = t \cos \theta - A_{kj} \sin \theta$ ;  
     $A_{kj} = t \sin \theta + A_{kj} \cos \theta$ ;  
  end for  
end for
```

---

where  $U$  is a  $m \times m$  orthogonal matrix ( $U^T U = I_m$ ),  $S$  is a  $m \times n$  rectangular diagonal matrix with non-negative numbers on the diagonal, and  $V$  is an  $n \times n$  unitary matrix ( $V^T V = I_n$ ). The column elements of  $U$  and  $V$  matrices correspond to the left-singular and right-singular vectors of  $M$ , respectively. Instead, the  $diag(\sigma_0, \sigma_{n-1})$  elements of the  $S$  matrix are the singular values of  $M$ .

Despite different SVD algorithms exist, the most practical algorithm is based on a Jacobi algorithm [192]. In this work, the One-Sided Jacobi algorithm that calculates eigenvalues of a symmetric matrix has been considered.

The first step is to compute the product  $M^T M$  to symmetrize the matrix  $M$ . Then, the One-Sided Jacobi algorithm applies a sequence of rotations to  $M_{symm}$  in order to reach the diagonal matrix  $S$ . Considering  $M_{symm}$  as the  $m \times m$  matrix  $A_0$ , the Jacobi algorithm produces a sequence  $A_1, A_2$  which eventually converges to a diagonal matrix with the eigenvalues on the diagonal [14].  $A_{i+1}$  is obtained from  $A_i$  by the transformation given by the formula

$$A_{i+1} = J(i, j, \theta)^T A_i J(i, j, \theta) \quad (7.6)$$

where  $J(i, j, \theta)$  is a Jacobi rotation. The Jacobi rotation  $J(i, j, \theta)$  (eq. 3) is introduced, for an index pair  $(i, j)$  and a rotation angle  $\theta$ , as a square matrix that is equal to the identity matrix  $I$  plus four additional entries at the intersections of rows and columns

$i$  and  $j$ . The Jacobi rotations are calculated on every  $2 \times 2$  matrix to annihilate the off-diagonal terms of  $A_0$ .

$$J(i, j, \theta) = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \quad (7.7)$$

The symmetrization and the Jacobi rotation operations are described in Algorithm 1 and 2, respectively.

### 7.7.2 Simulation and Results

In order to implement the TFP simulator (in Section 7.5), the SVD algorithm based on One Sided Jacobi method (OSJ) has been coded in the C main program. The TFP operations with arbitrary  $m, e$  have been invoked in the operation reported in Algorithm 1 and 2. Each operation of addition and multiplication had a different  $m, e$ , i.e., precision and dynamic range could be changed in different parts of the algorithm. Figure 7.13 shows the simulation flow. The symmetrization (SYMM) of Algorithm 1, and the Jacobi Rotation (JR) of Algorithm 2 are implemented in TFP in the simulation. The simulator executes the algorithm in both *double* and TFP, and provides the error in key points (SYMM and SVD). Moreover, by setting a debug option, the error can be displayed for each TFP operation.

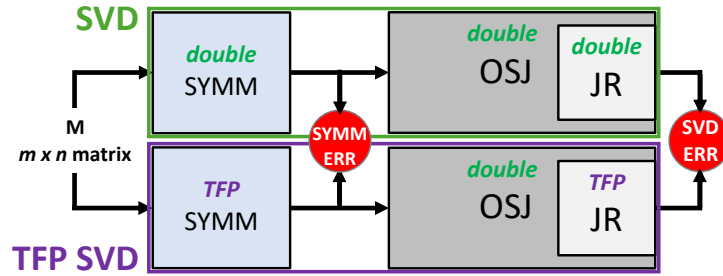


Figure 7.13: Simulation flow: SVD and TFP-SVD.

Multiplication is the operation that occurs more frequently in the analyzed SVD algorithm. In particular, the JR method consists of four multiplications, one addition and one subtraction, and is that these operations are repeated for multiple rotations. The number of rotations depends on  $n$ , the dimensions of the matrix which has been rotated, and  $n_{ITER}$ , the necessary iteration number to reach the diagonal matrix  $S$  through the OSJ.

As the JR have been defined as  $n(n - 1)/2$  transformations of the matrix which has been rotated, the crucial point of the OSJ algorithm is the stopping criteria, the convergence. If the OSJ iterations are stopped too early, the required accuracy may not be reached. In contrast, if the iterations are stopped too late, unnecessary operations would have been done increasing time and energy consumption. One criteria could be the evaluation of how close to zero are the off-diagonal elements of the original matrix.

The simulations have been run on a  $15 \times 10$  matrix with fractional random elements (input) with values in the range  $(0.0, 1.0)$ , and by keeping the exponent bit-width to  $e = 8$  and varying uniform significand bit-width to  $m = \{5, 7, 9, 11, 14, 16, 20, 24\}$ . Specifically, a  $15 \times 10$  matrix has been considered since sensors in [?] have the same size.

In Figure 7.14 and Figure 7.15, the average errors occurring in the SVD algorithm are reported under the three rounding modes: RTZ (blue), RTN (black), and RTNE (red).

Figure 7.14 shows the average error of the SYMM (dashed lines) and JR (continuous lines) algorithms when the TFP-precision has been used on each algorithm individually. Focusing on the average errors of the SYMM (addition and multiplication operations) (Figure 7.14-dashed lines), the RTZ leads to almost an order of magnitude larger error (i.e., factor 10, the scale is logarithmic on the y-axis). The RTZ also leads to a larger average error in the JR algorithm (Figure 7.14-continuous lines), for  $m > 9$  especially.

The error difference for RTN and RTNE is very similar. Figure 7.14 shows that for the RTN and RTNE simulations the average error with respect to the double-precision simulation is below 1% for  $m = 7$ . This error is probably acceptable for many applications including the prosthetic.

In Figure 7.15, the comparison between the whole SVD algorithm (based on OSJ) with double-precision and with TFP precision, where TFP-precision has been used for the SYMM and JR operations, has been analyzed. In particular, the average errors have been obtained by observing the errors between final  $V$  matrices for the SVD and TFP-SVD, respectively, when the algorithms reached the convergence point.

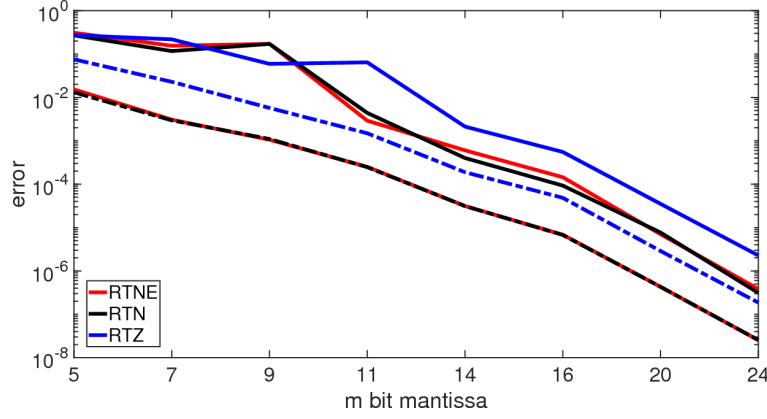


Figure 7.14: Average error for SYMM (dashed lines) and JR (continuous lines) algorithms under TFP rounding modes.

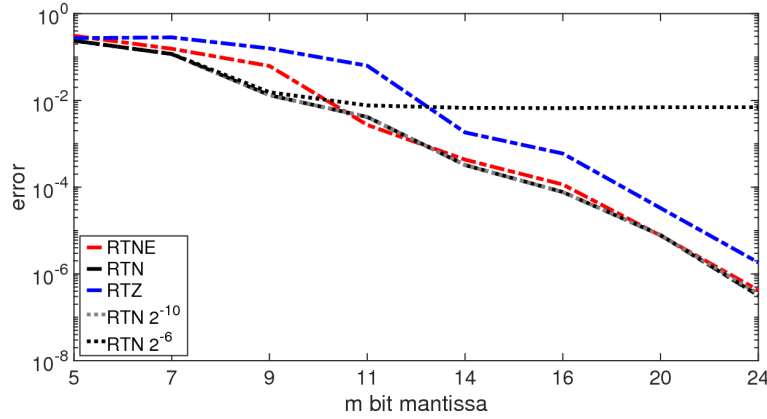


Figure 7.15: Average error for SVD algorithm under TFP rounding modes.

The overall average error are slightly larger than the errors obtained after the TFP-SYMM. This is due to the fact that the errors computed at the JR level included also the TFP symmetrization errors. Figure 7.15 depicts that for the RTN simulations the average error with respect to the double-precision simulation is below 1% for  $m = 9$  and for RTNE simulations is below 1% for  $m = 11$ . Also in this analysis, the RTZ results to be the worst rounding mode.

Simulations were also repeated for exponent bit-width  $e = 5$  and the result in errors were almost identical. As consequence, for this range of operands and operations the error mostly depends on the significand precision.

In all the simulations illustrated (double- and TFP- precision), eight was the

iteration number ( $n_{ITER}$ ) in order to respect the convergence criteria and annihilate only the upper off-diagonal elements of the rotated  $S$  matrix. Regarding the total number of JR ( $n_{ROT}$ ), in SVD and TFP-SVD algorithm the average  $n_{ROT}$  were 235 and 230, respectively. As there is almost no difference between the obtained  $n_{ROT}$  and the goal of this work is reducing power at the cost of accuracy results, another convergence criteria has been implemented. The new criteria has been defined by stopping SVD algorithm when the upper off-diagonal elements of the rotated  $S$  matrix are less than certain values (i.e.  $2^{-10}$  -  $2^{-6}$ ). In Figure 7.15, the trend of the average errors obtained by comparing SVD and TFP-SVD under the RTN mode for the three different convergence criteria (black and grey lines) has been depicted. Despite the number of iterations did not apparently change significantly in SVD and TFP-SVD algorithm (i.e. from 8 to 5), the variation of the rotation number changed considerably.

As the multiplication is the dominant operation in the considered algorithm, only the TFP multipliers have been estimated. Besides, the clock frequency was set to 100 MHz since the application target is a low-power embedded system.

Figure 7.16 depicts the results for the average power dissipation. The trends show that the power dissipation drops linearly as  $m$  is scaled. Using TFP multipliers allows power efficiency reductions up to 50 % and 70 % compared to *binary32* with  $m=11$  and  $m=5$ , respectively.

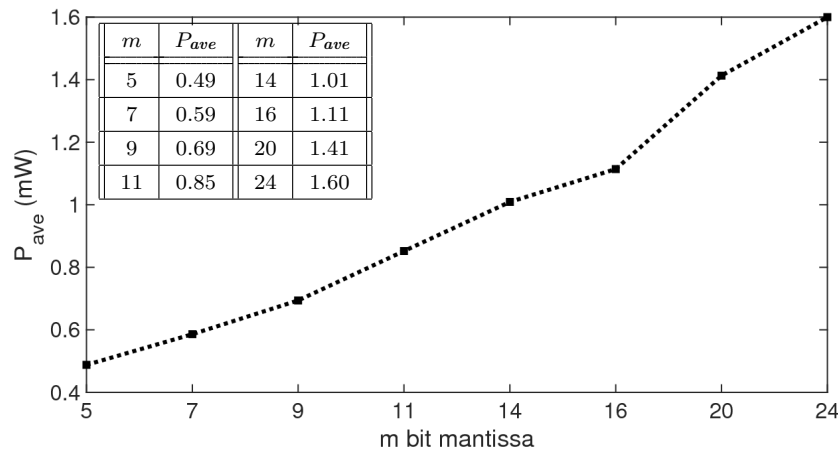


Figure 7.16: Average power dissipation ( $mW$ ) for TFP multiplier at 100 MHz.

Figure 7.17 illustrates an estimation of the energy for the JR (i) when the TFP-SVD convergence is to 0 (grey bars), (ii) when TFP-SVD convergence is to  $2^{-10}$  (yellow

bars) and (iii) to  $2^{-6}$  (brown bars). The energy to perform the JR required in the OSJ algorithm is defined as:

$$E_{JR} = P_{ave} \cdot t_{MUL} \cdot 4 \cdot n_{ROT} \quad (7.8)$$

where  $P_{ave}$  is the average power for each  $m$  expressed in Figure 7.16,  $t_{MUL}$  is the multiplier latency (20 ns), 4 indicates the number of multiplications in each JR, and the  $n_{ROT}$  values are reported in Table 7.6. Figure 7.17 shows a persistent gap among the TFP-SVD convergence to 0, to  $2^{-10}$  and to  $2^{-6}$ .

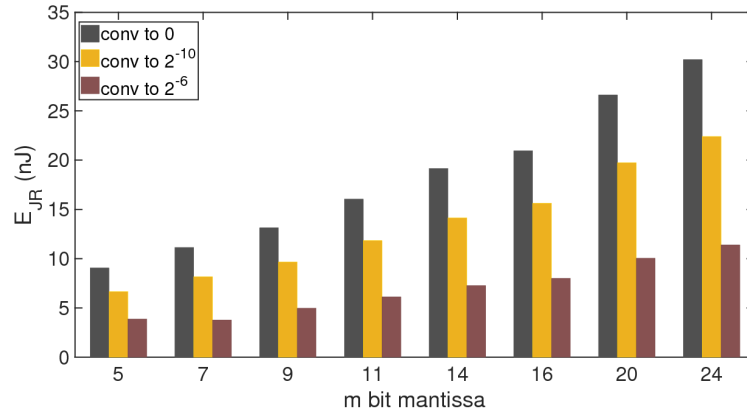


Figure 7.17: Energy to perform the JR required in the OSJ algorithm.

|                  |                  | m bits |     |     |     |     |     |     |     |
|------------------|------------------|--------|-----|-----|-----|-----|-----|-----|-----|
| conv<br>criteria | 0                | 5      | 7   | 9   | 11  | 14  | 16  | 20  | 24  |
|                  | 2 <sup>-10</sup> | 231    | 236 | 238 | 236 | 237 | 236 | 236 | 236 |
|                  | 2 <sup>-6</sup>  | 170    | 173 | 175 | 174 | 175 | 176 | 175 | 175 |
|                  | 2 <sup>-6</sup>  | 99     | 80  | 90  | 90  | 90  | 90  | 89  | 89  |

Table 7.6: Number of JR depending on the convergence criteria.

Observing Figure 7.15 and Figure 7.17 shows that the TFP-SVD is more convenient. Focusing on  $m=9$ , an error of  $10^{-2}$  could save up to 65 % of energy depending on the chosen convergence criteria (30 % for a convergence to  $2^{-10}$  and 65 % for  $2^{-6}$ ).

## 7.8 Conclusion

Today power efficiency is probably one of the most relevant aspects in the design of embedded digital systems.

The development of embedded and real-time systems for Machine Learning data processing is challenging (e.g. IoT) as relies on the efficient implementation and power requirements. Machine Learning algorithms execute a large number of operations, mainly additions and multiplications, which are the most power demanding arithmetic operations.

In order to increase the power efficiency in processing Machine Learning algorithm, new desing and trade-off need to be explored. Therefore, approximate computing techniques have been investigated. In fact, approximate computing has emerged as a promising approach to energy-efficient design of digital systems in many domains, including digital signal processing and machine learning.

Considering Machine Learning algorithms, an interesting approach is the use of Tunable Floating-Point (TFP) precision in computations, which allow to arbitrarily set the precision for each operation by selecting a specific number of bits for significand and exponent in the floating-point representation. By profiling and tuning the precision for a given algorithm, an efficient trade-off can be achieved to allow an acceptable target error while lowering the energy cost of computations.

This chapter has proposed the implementation of a Tunable Floating-Point technique in two different test cases:

- 1) Into a deep neural network with two-hidden-layers to achieve a lower power consumption implementing Tunable Floating-Point adder and multiplier. The TFP-unit worked at different precisions in the two phases of training and inference of neural networks by simply setting the required precision of operations. Experiments showed that TFP may (a) significantly improve the results of NN inference with respect to fixed-precision formats, and (b) be used in a novel flexible mode to save more power at cost of reasonable errors.
- 2) Into a Singular Value Decomposition algorithm based on One Sided Jacobi method implementing the Tunable Floating-Point multiplier. The use of Tunable Floating-Point representation has allowed high performance and efficiency improvements of the SVD, with up to 57 % power consumption reduction and up to 65 % energy, at the cost of a negligible 1 % average error, in the specific case  $m=9$  and  $e = 8$ . This TFP implementation into the SVD algorithm is sub-task of the computationally expensive Machine Learning algorithm used in tactile data processing [199], and represents a successful investigation of TFP for real-time embedded prosthetics.

# Part IV

## Outlook and Conclusion





# Chapter 8

## Outlook and Conclusion

This thesis presented an electronic skin system to close the loop in prostheses, restoring the sense of touch in prosthesis user.

Apart from a single recent example [3], none of today's prostheses have purposely designed sensory feedback. Integrating an advanced distributed sensing (electronic skin), electronic system for (a) signal conditioning, (b) data acquisition, and (c) tactile data processing and stimulation system into a myoelectric prosthesis would allow closing the prosthesis control loop by providing sensory feedback to the user.

A prosthetic system that not only responds to the control signals provided by the user, but also transmits back to the user the information about the current state of the prosthesis fosters the prosthesis embodiment and utility. It is an important step towards improving the quality of life of people with limb amputation and key point in research on active prosthetics.

In particular, this work showed a tactile feedback system, allowing the communication of a mechanical interaction from the electronic skin to prosthesis users, and the dedicated implementation of algorithms for processing tactile data originating from the e-skin.

In moving toward the new concept for prosthesis capable of being implemented for long-term use, the concept of providing *natural, physiological feedback* must be considered. As many systems proposed in literature require training and sensory adaptation to interpret signals [101], an additional processing of information could increase the cognitive load and could have the potential to negate one of the largest benefits of sensory feedback reducing conscious attention. It is important to develop sensory feedback systems reporting the naturalness of measured sensations as well as the amount of cognitive burden required.

Attempting to mimik the naturalness of a sensation, a system was implemented to directly transmit mechanical information from a multi-point tactile sensor to the human subject by using multichannel electrotactile stimulation. The system was evaluated by assessing the ability of the human subjects to perceive the properties (shape, trajectory, direction) characterizing dynamic and versatile mechanical interaction with the skin. This was the first development integrating an advanced tactile sensor with many sensing elements and an electrotactile stimulation unit with a flexible matrix of electrodes into an online system for the transmission of tactile data from artificial to natural skin (forearm).

However, the current study did not yet demonstrate the utility of the proposed technology in the real-life application, which is an important future goal. To this aim, the next step will be to cover a myoelectric prosthetic hand with an electronic skin in order to test the closed-loop system during functional tasks.

On the other hand, wider scenarios can open on how to give back sensor data to the prosthesis user and how to help the human brain to successfully interpret the elicited artificial tactile information. In fact, among open questions in prosthetics is which kind of information - whether *raw* or *processed* data- about a touched object should be sent back to the user. An approach is to send the sensor signals directly to the user, who needs to meaningfully interpret this information. Alternatively, learning from robotics, sensor data can be locally processed at the body periphery (prosthesis socket with embedded electronics) and high-level tactile information (e.g., texture properties, grasp stability) can be extracted and delivered to the user. Machine Learning algorithms have been exploited to classify and interpret input touch modalities as they represent a powerful technology for tackling clustering, classification and regression problems in complex domains, e.g., robotics.

An essential task of an electronic skin system for the restoration of the sense of touch is to process the tactile data and send information either to mimic human skin or to respond to the application demands. More specifically, Machine Learning based on Tensorial Kernal approach has been proposed to interpret touch modality in the electronic system to close the loop in prosthetics. The computational complexity of the employed Machine Learning algorithm has been assessed and showed that too many resources were needed. As the requirements for a tactile data processing unit with high-level information extraction are very challenging and still far from being achieved, methods and techniques to reduce hardware complexity and power consumption have been explored.

In the last few years, approximate computing has become a major field of research as it could significantly improve energy efficiency and performances of modern digital circuit. Inexact and approximate technique design is a radical approach to trade this counterproductive quest for perfection for substantial gains in power, speed, and area. The primary challenge, however, is to determine *where* and *how* to let an error or an approximation occur in the circuits without compromising the functionality or the user experience. With ever-increasing amount of data being processed, a wide variety of applications could tolerate inaccuracies.

Two encouraging approaches have been studied and implemented in the Machine Learning for the touch modality classification. The first one uses speculation to trade circuit accuracy for better power and speed. It was the first attempt of implementing an approximate Coordinate Rotation Digital Computer on FPGA with Inexact Speculative Adders. The use of speculative adders has allowed high performance and efficiency improvements showing negligible average and maximal errors.

The second approach uses Tunable Floating-Point precision in computations, which allow to arbitrarily set the precision for each operation by selecting a specific number of bits for significand and exponent in the floating-point representation. As well as the previous case, the Tunable Floating-Point representation has allowed to considerably reduce the power consumption at the cost of a insignificant errors into the Singular Value Decomposition algorithm.

These first implementations of the approximate techniques into the Coordinate Rotation Digital Computer and the Singular Value Decomposition algorithms were a sub-task of the computationally expensive Machine Learning algorithm used in tactile data processing, and represented a successful investigation for real-time embedded prosthetics. The next step will be to implement the approximate techniques into the overall Machine Learning algorithm for the touch modality classification in order to establish the maximum reduction of the computational algorithm complexity. If these methods allow to achieve better results, the future development will be to design an integrated circuit specific for the restoration of the sense of touch in upper limb prosthetics.

# Appendix A

## Ethical Committee Proposal

In this appendix, part of the application for the Ethical Committee approval is reported. For brevity reasons, not all the application documents have been listed.

The Ethical Committee approval has been necessary since the stimulator device is not a commercial product and since it directly communicates with experiment subjects injecting current through superficial electrodes wrapped around the subject's forearm.

This study was evaluated and approved on September 16<sup>th</sup>, 2016 (protocol number: P.R.172REG2016).

The candidate took care of obtaining the consent writing the proposal.

## DATI INTEGRATIVI ALLA RICHIESTA DI PARERE

|   |  |  |
|---|--|--|
| Codice Protocollo   | NON APPLICABILE  |  |
| Titolo (in italiano)  | TRASMISSIONE DI INFORMAZIONI TATTILI ATTRAVERSO ELETSTIMOLAZIONE   |  |
| <b>Tipologia di studio</b>  | Osservazionale <input type="checkbox"/>  | Interventistico con farmaco <input type="checkbox"/>     |
|   | Interventistico senza farmaco <input type="checkbox"/>   | Chirurgico <input type="checkbox"/>                      |
|   | Dispositivo <b>X</b>   | Traslazionale / Ricerca di base <input type="checkbox"/> |
|   | Profit <input type="checkbox"/>  | No-Profit <b>X</b>                                       |
|   | Fase I <input type="checkbox"/> Fase II <input type="checkbox"/> Fase III <input type="checkbox"/> Fase IV <input type="checkbox"/>  |  |
| SPERIMENTATORE PRINCIPALE<br>(indicare Nome Cognome Ente di appartenenza del PI e Unità Operativa)  | <b>DOTT. LUCIA SEMINARA</b><br>Dipartimento di Ingegneria Navale, Elettrica, Elettronica e delle Telecomunicazioni – DITEN<br>UNIVERSITA DEGLI STUDI DI GENOVA   |  |
| n. EudraCT<br>(se applicabile)  | NON APPLICABILE  |  |
| CENTRO COORDINATORE<br>(indicare le informazioni utili relative al coordinatore e al relativo centro: Nome, Cognome, Titolo, Telefono, FAX E-mail del PI, UO, indirizzo del Centro) | <b>COSMICLab</b> - <a href="http://www.cosmiclab.diten.unige.it">http://www.cosmiclab.diten.unige.it</a><br>Dipartimento di Ingegneria Navale, Elettrica, Elettronica e delle Telecomunicazioni – DITEN<br>UNIVERSITA DEGLI STUDI DI GENOVA<br><br>(1) INVESTIGATORE PRINCIPALE<br><b>Dott. LUCIA SEMINARA</b><br>+39 010 353 2775<br><a href="mailto:lucia.seminara@unige.it">lucia.seminara@unige.it</a><br><br>(2) CO-INVESTIGATORE<br><b>Dr.ssa MARTA FRANCESCHI</b><br>+39 010 353 2757<br><a href="mailto:marta.franceschi@edu.unige.it">marta.franceschi@edu.unige.it</a> |  |
| PROMOTORE   | <b>PROF. MAURIZIO VALLE</b>  |  |
| CRO   | NON APPLICABILE  |  |

|  |   |             |
|--|---|-------------|
| <b>DATI PER FATTURAZIONE</b><br>(Cognome, nome, numero telefono, email)<br>(se applicabile)                  | NON APPLICABILE   |             |
| <u>Indirizzo per la Fatturazione</u><br>(se applicabile)   | NON APPLICABILE   |             |
| <u>Parere Unico Centro Coordinatore allegato</u>   | Si <input type="checkbox"/>   | No <b>X</b> |
| <u>Numero di pazienti previsti per il centro</u>   | 50  |             |
| <u>Compenso per paziente completato</u>  | NESSUN COMPENSO   |             |
| <u>Data prevista di inizio sperimentazione</u>   | 2 MAGGIO 2016   |             |
| <u>Data prevista di fine sperimentazione al Centro</u><br>(approssimativamente)                              | MAGGIO 2020   |             |
| <u>Durata della sperimentazione in toto</u><br>(espressa in mesi)  | 48 MESI   |             |
| <u>Contatto/i</u><br><u>di riferimento per lo studio</u><br>(indicare Nome Cognome cellulare, fisso, e-mail) | <b>MARTA FRANCESCHI</b><br>010 353 2781<br>+39 340 9089187<br>marta.franceschi@edu.unige.it |             |

Dettagliato elenco dei documenti presentati in formato word (non lista Ia o Ib),  
comprensivo di date e versioni

| OGGETTO  | NOME FILE        | VERSIONE | DATA       |
|--|------------------|----------|------------|
| Modulo CERLiguria_Mod4_Study_info  | CEN a            | 1.0      | 22/04/2016 |
| Informativa paziente, consenso al trattamento dei dati personali, modulo di Consenso Informato | CEN bc           | 1.0      | 24/03/2016 |
| Lettera al medico di Medicina Generale   | CEN d            | 1.0      | 22/04/2016 |
| Mod1_Modulo di cessione in comodato d'uso di attrezzature sanitarie/informatiche               | CEN e<br>ADAPTED | 1.0      | 12/04/2016 |
| Informativa su differente formato del modulo di cessione                                       | CEN e INFO       | 1.0      | 24/03/2016 |
| Lettera di trasmissione/Richiesta di Autorizzazione  | GEN a            | 1.0      | 14/04/2016 |
| PROTOCOLLO in esteso   | GEN b            | 1.0      | 08/04/2016 |
| SINOSSI del protocollo   | GEN c            | 1.0      | 06/04/2016 |
| Scheda clinica per la raccolta dei dati  | GEN d            | 1.0      | 14/04/2016 |
| Manuale d'uso del dispositivo in lingua inglese  | GEN e            | 1.0      | 14/04/2016 |
| Approvazione Comitato Etico serbo (originale)  | GEN f            | 1.0      | 15/01/2016 |
| Approvazione Comitato Etico serbo (trad. ITALIANO)   | GEN f            | 1.0      | 19/04/2016 |
| Disponibilità ad attivare assicurazione  | GEN g            | 1.0      | 08/04/2016 |
| Cartella Selezione BIBLIOGRAFIA  | GEN h            | 1.0      | 22/04/2016 |
| Comunicazione attivazione studio al Direttore di Dipartimento                                  | GEN m            | 1.0      | 11/04/2016 |
| Mod8_Richiesta sperimentatore locale   | LOC a            | 1.0      | 22/04/2016 |
| Curriculum vitae breve   | LOC b            | 1.0      | 24/03/2016 |
| Mod3_Modello dichiarazione pubblica sul conflitto d'interessi                                  | LOC c            | 1.0      | 17/03/2016 |



## PROTOCOL

**Protocol Title:** Tactile information transmission through electrostimulation

**Protocol Version:** 1.0

**Protocol Date:** 08 April, 2016

**Principal Investigator:** Prof. Maurizio Valle

**Research Team:** Marta Franceschi, Lucia Seminara, Luigi Pinna

### I. Abstract

Myoelectric prostheses are successfully controlled using muscle electrical activity, thereby restoring lost motor functions. The association between the muscle activity and the prosthesis functions provides an intuitive connection between the brain and the prosthesis; however, this connection is unidirectional. A bilateral communication between the brain and the periphery is necessary for human movement learning and execution [1], but the somatosensory feedback from the prosthesis to the user is still missing.

The idea to provide a sensory feedback to an amputee controlling upper limb prosthesis is not a new concept [2]. This was quite a popular research topic during 60's and 80's within the field of the so-called sensory substitution, where the goal is to substitute a missing sense by using an alternative intact sense. With respect to the aforementioned previous research (in 60's, 70's and 80's), many new technologies are now available to implement the feedback interface. They offer more capabilities and they are also better in terms of ergonomics and practicality. For example, there are smaller motors, multichannel electronic stimulators, more powerful processors, and better sensors built within the hand.

Out of many commercially available systems, there is only one recently presented device integrating a simple vibrotactile feedback of grasping force. Integrating an advanced distributed sensing (artificial skin) and stimulation system into a myoelectric prosthesis would allow closing the prosthesis control loop by providing sensory feedback to the user. A prosthetic system that not only responds to the control signals provided by the user, but also transmits back to the user the information about the current state of the prosthesis fosters the prosthesis embodiment and utility. It is an important step

towards improving the quality of life of people with limb amputation and key point in research on active prosthetics [3], [4].

## II. Current study

In this study, we present a prototype of a distributed *sensing* (artificial skin) *and stimulation* interface. The system comprises an artificial skin including a matrix of sensing elements (taxels), acquisition electronics and multichannel stimulator connected to flexible electrode matrix placed on the forearm. Electrotactile stimulation is conducted by delivering low level electrical current pulses through specific electrodes placed on the skin, to stimulate skin afferents and activate the tactile sense.

The envisioned future setup would integrate a prosthetic hand covered by artificial skin, miniature stimulators, and a matrix of stimulation electrodes integrated into the prosthetic socket.

In this framework, our overall scope is to study **what are the best methods to deliver artificial tactile information (recorded by artificial skin) to the prosthesis user through multichannel electrocutaneous stimulation and to test the effectiveness of the prosthetic system embodiment.**

While testing the prosthetic system on amputees is the final goal, at present we study how this works on healthy and voluntary subjects.

In details, our goals are detailed in the following:

**1) Studying methods to transmit tactile information by electrostimulation**, in order to foster the participant recognition of the tactile stimulus.

Consequently, we will:

a) **optimize electrostimulation parameters for each participant**

Touch information should be reliable during prolonged periods of use, naturally interpretable and transferred without inducing pain. However, it is known that the perceived sensation decays within a certain period (minutes) due to central adaptation, both if the stimulation is applied continuously and intermittently [5], [6] and [7].

Therefore, the first step in our research is to **optimize the stimulation to reduce adaptation**. The fact that higher frequencies cause shorter adaptation [8] can be managed by applying a specific novel stimulation protocol, in which the pulses will be distributed sequentially over several closely positioned electrodes. The sequential stimulation of different electrodes over a reduced area gives the subject the impression of a localized stimulation at higher frequency than each electrode frequency.

The experiments will study the sensation decay during classic and novel stimulation protocols:

- the adaptation time as a function of the stimulus **frequency** (fixed amplitude, fixed waveform type), applying the stimulus **sequentially** to different stimulation electrodes (one channel after the other), i.e., the novel protocol;
- the adaptation time as a function of the stimulus **frequency** (fixed amplitude, fixed waveform type) applying the stimulus **synchronously** to different stimulation electrodes (one channel after the other), i.e., the classic protocol;
- the adaptation time as a function of the stimulus **amplitude** (fixed amplitude, fixed waveform type) in different stimulation electrodes and protocols.

b) evaluate the **capabilities of the human brain to successfully interpret** the elicited **artificial tactile information**.

The task will be to perceive and interpret the delivered electrical stimuli in order to extract the properties of the mechanical stimulus applied onto the e- skin. More specifically, from electrostimulation the subjects will try to recognize spatial and temporal features (shape, direction and trajectory) of the mechanical stimuli moving over the e-skin surface.

Consequently, methods for processing tactile data acquired from the skin will be tested as well as the translation of this processed information into patterns of electrical stimulation to be delivered to the subject.

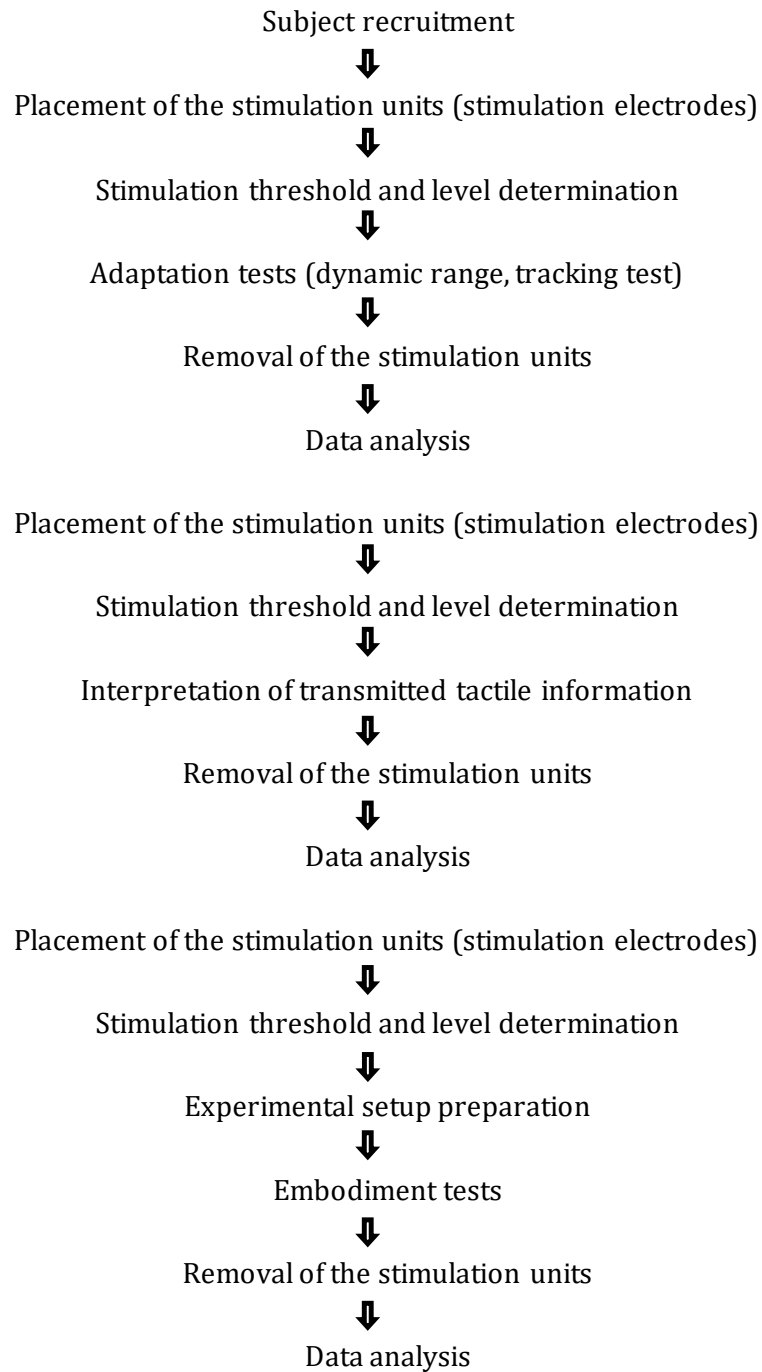
## 2) Testing and evaluating the effectiveness of embodiment of the prosthetic system (artificial skin + electrostimulation).

**To date, different experiments have been conducted showing as humans are able to feel a rubber (or robotic) hand [9-11] as if it was their hand.** Rubber hand illusion is based on a three-way interaction between vision, touch and proprioception, and may supply evidence concerning the basis of bodily self-identification.

We propose a modification of the original experiment: **we will integrate both an artificial skin into the rubber hand and electrostimulation** to convey artificial tactile information to the subject.

Once we have defined the best electrostimulation methods to transmit tactile data, is the participant able to recognize that anything happened on the skin (e.g. somebody touched) and exactly perceive what happened (e.g. a specific object is sliding from gripping), without watching it? This would mean that the prosthesis is identified by the participant him/herself as a part of his/her body (=embodiment).

### Flow chart of the project



### III. Experiment description (Design of experiments)

The overall experiment takes place on three days. Each experimental session lasts for about 2.5 hours, except for the last session that will last about 30 minutes. The days do not have to be consecutive. A researcher from the research team will conduct the experimental tasks in these three sessions.

#### **Session 1: Touch/tactile information transmission modality tests.**



**Fig. 1**

#### Step 1: PREPARATION

The subjects will seat comfortably on a chair in front of a table. The forearm of the dominant arm will be placed on the table surface, with the volar side oriented upwards. An electrode matrix will be positioned on the volar aspect of the subject forearm. The matrix will be placed at one third of the forearm length distal to the elbow. The electrode matrix will be then secured with medical tape to prevent movement and improve contact. This phase will last about 5 minutes.

#### Step 2: THRESHOLD DETERMINATION

Electro tactile sensation thresholds will be determined for all employed stimulation electrodes (out of 16), individually, at fixed frequencies (in the 10-400 Hz range). For each electrode and at each frequency, the participant is asked to identify the sensation and discomfort thresholds. The sensation threshold is defined as the lowest current amplitude at which the stimulation is perceivable. Similarly, the discomfort threshold is defined as the lowest current amplitude at which the subject evaluates the sensation as uncomfortable. In order to determine sensation and discomfort thresholds, the current pulse amplitude, starting at zero, will be increased in steps of 0.1 mA until the subjects will report that

they feel the stimulation for the first time and that the stimulation becomes uncomfortable, respectively. This phase will last between 10 and 20 minutes.

### Step 3: STIMULATION LEVEL DETERMINATION

As previous studies have already tested the effects of low, medium and high stimulation levels on adaptation time and their results have indicated high adaptation with low stimulation level and low adaptation with high stimulation level (significant changes in the sensation and marginally or no significant changes in the discomfort thresholds after the sensation decay measurements) [7], we have decided to test the medium stimulation level (exactly in between sensation and discomfort thresholds). However, each stimulation electrode will have its own medium stimulation level, computed as the mean of the interval between the sensation and discomfort thresholds, and the sensation elicited using the medium level may be electrode specific. To minimize this effect, adjustments will be made to equalize the tactile sensations across the electrodes. It is important that the sensations over the different channels are similar because this will improve the comfortability. This phase will last few minutes.

### Step 4: ADAPTATION TESTS

Experiments will consist of 8-12 sensation decay measurements. Each measurement will last maximally 15 min (900 s) and will be followed by a pause of 5 min (300 s). The duration of the stimulation and the pause was selected based on the literature, as enough time to lead to adaptation and recovery, respectively [5]. During each trial, the subject will be instructed to judge the sensation level after an auditory cue, every 30 s. This judgment will be recorded using a commonly accepted Visual Analog Scale (VAS) [12]. VAS is a line of 10 cm with two boundaries, the left corresponding to no sensation, and right to discomfort. The subject is required to quantify the sensation level by inserting a cross on the line of the VAS scale.

During every sensation decay measurement, it will be evaluated if the subject can detect the end of the stimulation. To this purpose, the subject will be instructed (i) that the stimulation may be stopped randomly by the experimenter, and (ii) that he/she should indicate a “stop” when she/he feels that the stimulation has stopped.

During each sensation decay measurement, the stimulus amplitude and frequency will be fixed. Each sensation decay measurement will be done twice (at each frequency) in a random order.

This phase duration will depend on the participant adaptation time, but never exceeds 2 hours.

## **Session 2: Haptic tests.**



**Fig. 2**

### **Step 1: PREPARATION**

The subjects will seat comfortably on a chair in front of a table. The forearm of the dominant arm will be placed on the table surface, with the volar side oriented upwards. Two electrode matrices will be positioned on the volar aspect of the subject forearm. The first electrode will be placed with its center at one third of the forearm length distal to the elbow and the second right after it, towards the hand. The electrode matrices will be then secured with medical tape to prevent movement and improve contact. This phase will last about 5 minutes.

### **Step 2: THRESHOLD DETERMINATION**

Electro tactile sensation thresholds will be determined for all employed stimulation electrodes (out of 32), individually, at fixed frequencies (in the 10-400 Hz range). For each electrode and at each frequency, the participant will be asked to identify the sensation threshold (see Session 1). This will allow fixing the stimulation amplitude for each electrode, corresponding to a higher value with respect to the sensation threshold, so that the elicited tactile sensation will be comfortable and clearly

recognizable. Furthermore, adjustments will be made to roughly equalize the tactile sensations across the electrodes. This phase will last between 15 and 30 min.

### Step 3: PATTERN RECOGNITION

The experiment will comprise blocks of trials in which the experimenter will apply a mechanical stimulus to the e-skin, and stimulation profiles will be delivered to the subject accordingly.

The task for the subject will be to perceive the stimulation and report to the experimenter the properties of the mechanical stimulus.

The mechanical stimulus will be applied to the skin using a small roller to ensure contact reproducibility between different trials. To produce a natural and realistic motion, the experimenter will move the roller along the skin at a self-selected speed and pressure, aiming to be consistent across the trials.

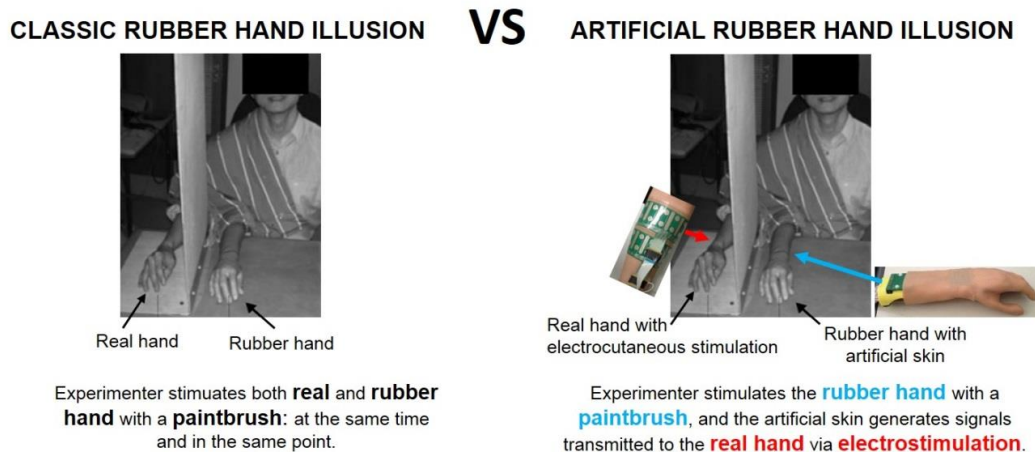
Three categories of mechanical patterns can be tested: single lines, geometrical shapes and letters. Each set will be presented to the subjects in five blocks and within each block the test patterns will be applied in a random order.

The subject could not see the experimenter nor the skin. A sheet of paper will be placed in front of the subject with a schematic drawing of the electrode matrix. After receiving the electrical stimulation, the subjects will be first asked to verbally estimate the perceived shape of the movement: longitudinal, transversal, diagonal line; a specific geometry (e.g. square, rectangle or triangle); an exact letter. If the shape estimate is correct, the subjects will be asked to retrace the perceived stimulus by moving their index finger across the schematic drawing. If this corresponds to the actual rolling pathway, it will be deemed that the subjects correctly recognized the movement trajectory. Finally, if the subjects retrace the trajectory segments (lines) in the proper direction, the movement direction will be also correctly perceived.

This phase will last about 90 min.



### Session 3: Embodiment tests.



**Fig. 3**

#### Step 1: PREPARATION

The subjects will seat comfortably on a chair in front of a table. The right forearm will be placed on the table surface, with the dorsal side oriented upwards. A stimulation electrode matrix will be positioned on the dorsal aspect of the subject forearm, with the extremity over the wrist. *The electrode will be then secured with medical tape to prevent movement and improve contact.* This phase will last about 5 minutes.

#### Step 2: THRESHOLD DETERMINATION

See Step 2, Session 2.

#### Step 3: ARTIFICIAL RUBBER HAND ILLUSION SETUP

A standing screen will be positioned beside the subject's arm to hide it from the subject's view. A life-sized *artificial* rubber model (integrating the artificial skin already tested in Session 2) of a right arm will be placed on the table directly in front of the subject. The subject will sit with eyes fixed on the artificial hand.

### Step 3: ARTIFICIAL RUBBER HAND ILLUSION TESTS

The experimenter will use a small paintbrush to stroke the artificial rubber hand, while the subject's hidden hand will be solicited by the corresponding electrostimulations. Each stroke will be about 2-3 cm long. During this phase, the subjects will be instructed to relax and observe the artificial rubber hand on the table.

The ARHI test will consist of five trials. Each will last for 2/3 minutes using an irregular rhythm and between every trial will be 40 seconds.

When the ARHI test is finished, subjects will complete a questionnaire. The questionnaire will indicate if subjects experienced an illusion in which they would feel the touch they were looking at, as if they owned the artificial rubber hand.

To obtain behavioural evidence that the illusion happened, a "pointing task" would be required (adopted in [13]).

Immediately before starting the experiment and after each trial, the participants will be required to close their eyes and point to where they feel their arm. A ruler mounted on the table will be used to measure the distance between the point indicated by subject's index finger and his real arm. The larger the distance is, the higher the number of subjects who feel the artificial rubber hand as their real hand. Obviously, the pointing drift should be from the real hand towards the artificial rubber one (Fig. 3).

#### IV. Duration

Date of study: 2 May 2016 – May 2020

Recruitment start: 2 May 2016

Date of termination of recruitment: March 2020

Date of study termination: May 2020

#### V. Risk-benefit analysis

Both direct mechanical and electrical stimulation have been used in a number of previous studies and also within working prototypes to provide the sensory feedback [14-17]. There are no known side effects of the electrical stimulation. This is generally perceived as a gentle pinch that is typically judged as quite comfortable by the subjects. In the current study, the intensity of electrical stimulation is very low (I: 4-5 mA, PW: 100-500 us), much lower than that applied for therapeutic or orthotic purposes (e.g., functional electrical stimulation and therapy). In the latter case, the stimulation can cause skin

damage if not properly applied. However, if standard practices are followed (new electrodes, good skin-electrode contact), it is usually considered as highly unlikely for the skin burns to happen [18]. In our case, due to the very low intensity that is employed in this study, the risk for skin burns is truly minimal. As a precaution, we will be checking the skin-electrode contact multiple times throughout the experiment. Some subjects may find the sensations due to electrical stimulation as quite uncomfortable. To minimize the discomfort, stimulation parameters will be adjusted by relaying on the constant feedback from the patient, i.e., the patient will indicate if the stimulation is uncomfortable and the stimulus intensity will be decreased accordingly. Furthermore, the parameter adjustment will always start from the minimum value and then gradually increase so that a subject can get used to the sensory consequences of the stimulation. Finally, before applying the stimulation, the skin will be checked for eventual small injuries (e.g., abrasions, scratches), and the stimulation will not be applied over.

The outcome of the current study will potentially improve the quality of life for amputees significantly, leading to the development of a closed loop controlled prosthetic and/or potentially a therapeutic device. Therefore, the proposed study has potentially a significant social and economic impact, and its benefit certainly outweighs the limited risks it involves.

## **VI. Subject recruitment**

### **a. Number of subjects and duration of the study**

The study will recruit a total of 50 able-bodied subjects. Subjects could participate in more than one experimental session. Subjects who will participate in two or three experimental sessions will do it on two or three separate days (respectively). The third session will last about 30 minutes, while the other session will individually last from 2.5 to 3 hours. The days have to be not consecutive.

### **b. Subject Selection**

All subjects will be recruited through the University of Genova.

#### **i. Inclusion criteria**

Healthy subjects aged between 18 and 65 years will be included in the study. No gender preference is applied. The subject has to have legal capacity to give the consent at his/her own will.

#### **ii. Exclusion criteria**

Subjects having any illness (active diagnosis) and/or receiving any medication, taking drugs or suffering from alcohol abuse will be excluded from the study. If the subject is taking part in other medical, clinical or scientific experiments/studies he/she will also be excluded.

## VII. Medical screening

### a. Personal data

Only the necessary personal data for the identification of the subjects during the experiment will be collected. All personal data collected during the experiment, such as name, age, and gender, will be strictly confidential, and will not be disclosed to a third party other than the patient and the experimenter. When the experimental results are reported, the identity of the subjects will be replaced with an ID (numerical code). The connection between the ID and the subject will only be present in a separate, dedicated document (i.e., a key list). The list will contain subject names and IDs, and double copies will be kept in locked cabinets in the office of the chief investigator and/or people of the research team. It will not be given to a third party, and it will be kept stored for 10 years. A key list will never be present in the electronic form. In case of experiment interruption, personal data for subject identification will be deleted.

### b. Measurement parameters and laboratory parameters

The following parameters will be varied during the tests: 1) intensity and frequency of the electrostimulation of each employed electrode, 2) electrostimulation type (synchronous – sequential).

During the tests, we will measure the following variables: 1) sensation and discomfort thresholds, 2) sensation levels (VAS), 3) success rate of stimulus identification on the artificial skin transmitted through electrostimulation; 4) the distance between the perceived hand/arm and the real one; 5) BOTVINICK and COHEN questionnaire.

## VIII. Materials and Methods

Our experiments will employ a stimulation device (IntFES Ver 2 - MaxSens, Tecnia, Spain) that will generate electrostimulation profiles to be transmitted to the subject through a matrix of electrodes applied to the participant's forearm.

Electrostimulation is an electric pulse applied locally, via the nervous tissue, which stimulates the muscles and causes a physiological contraction. The electrostimulation created by IntFES Ver 2 - MaxSens replicates the natural process controlled by the brain.

Electrostimulation is a technique which has been practiced for many years and no major problems have been reported to date. There is no associated risk so long as you follow the safety instructions. Occasionally harmless rashes may appear on areas of sensitive skin. If this occurs, remove the electrodes and wash the area with soap and water. The rashes will disappear after a few minutes.

The IntFES Ver 2 - MaxSens stimulator comprises a single stimulation unit that can generate biphasic symmetric current-controlled pulses and distribute these pulses in time and space over the multi-pad electrode, thus providing multichannel stimulation interface. The stimulator is fully programmable and the stimulation parameters can be adjusted online by sending text commands from the host PC via a Bluetooth connection. The current amplitude can be changed in the range 0-5 mA in increments of 0.1 mA, the pulse width from 50 to 1000  $\mu$ s in increments of 10  $\mu$ s, and the frequency resolution is 1 Hz with the maximum rate of 400 Hz. The output channel of the stimulator is connected to a multiplexer which can route the generated pulse to one of 64 stimulation channels.

The stimulator can at maximum be connected to four flexible electrode matrices. Each electrode matrix has its own geometry and size. Each electrode matrix consists of 16 round electrodes. The round electrodes are the cathodes whereas the long bars in-between the cathodes served as the common anodes. This configuration is selected in order to localize the current flow, from a cathode to its closest anode segment, eliciting focused tactile sensation. The stimulation electrodes were made of a polyester layer, an Ag/AgCl conductive layer, and an insulation coating covering the conductive leads.

To improve the electrode-skin contact, conductive hydrogel pads (AG702, Axelgaard, DK) were placed on the top of the electrode pads. This design allowed electrode bending and provided a close contact of the electrode with the skin.

Our experiments will also employ an artificial skin that provides information about mechanical contact over its surface, through a high density matrix of piezoelectric polymer (Polyvinylidene fluoride - PVDF) tactile sensors. When a pressure is applied on the artificial skin, tactile information from the sensor matrix is sent to the host PC via USB port (processing software). Processed signals are sent to a stimulator and the subject receives the tactile information through electrostimulation. PVDF was selected since it has wide frequency bandwidth (1Hz-1kHz), large measurable pressure range (50Pa-1MPa), flexibility, conformability and low cost. Due to its frequency bandwidth, only dynamic contacts can be measured while static ones are not recorded. A rigid substrate is currently used as a support and a 2.5 mm thick Polydimethylsiloxane (PDMS) elastomer layer is used for stress transmission and sensor protection [19]. In between, a commercial [20] PVDF sheet integrates on both sides 64 ad-hoc screen-printed electrodes [21] organized in a rectangular matrix.

## IX. References

- [1] J. C. Rothwell, M. M. Traub, B. L. Day, J. A. Obeso, P. K. Thomas, and C. D. Marsden, "Manual motor performance in a deafferented man.," *Brain*, vol. 105 (Pt 3), pp. 515–42, Sep. 1982.
- [2] D. S. Childress, "Closed-loop control in prosthetic systems: historical perspective," *Annals of Biomedical Engineering*, vol. 8, no. 4-6, pp. 293-303, 1980.
- [3] J. Ning, S. Dosen, K.-R. Muller, and D. Farina, "Myoelectric Control of Artificial Limbs—Is There a Need to Change Focus? [In the Spotlight]," *IEEE Signal Process. Mag.*, vol. 29, no. 5, pp. 152–150, Sep. 2012.
- [4] C. Antfolk, M. D'Alonzo, B. Rosén, G. Lundborg, F. Sebelius, and C. Cipriani, "Sensory feedback in upper limb prosthetics.," *Expert Rev. Med. Devices*, vol. 10, no. 1, pp. 45–54, Jan. 2013
- [5] K. A. Kaczmarek, "Electrotactile adaptation on the abdomen: Preliminary results," *IEEE Trans. Rehabil. Eng.*, vol. 8, no. 4, pp. 499–505, Dec. 2000.
- [6] A. Higashiyama and T. Tashiro, "Magnitude estimates for electrical pulses: Evidence for two mechanisms," *Perception Psychophys.*, vol. 45, pp. 537–549, 1989.
- [7] D. G. Buma, J.R. Buitenweg, P.H. Veltink, "Intermittent stimulation delays adaptation to electrocutaneous sensory feedback", *IEEE Trans. Neural Syst. Rehabil. Eng.* 2007 Sep; 15(3):453-41.
- [8] A. Y. J. Szeto and F. Saunders, "Electrocutaneous stimulation for sensory communication in rehabilitation engineering," *IEEE Trans. Biomed. Eng.*, vol. 29, no. 4, pp. 300–308, Apr. 1982.
- [9] Botvinick, M. & Cohen, J. 1998 Rubber hands 'feel' touch that eyes see. *Nature* 391, 756.
- [10] G. L. Moseley, N. Olthof, A. Venema, S. Don, M. Wijers, A. Gallace, and C. Spence, "Psychologically induced cooling of a specific body part caused by the illusory ownership of an artificial counterpart", *Journal List, Proc Natl Acad Sci U S A*, v.105(35); 2008 Sep 2, PMC2529116
- [11] B. Rosén, HH Ehrsson, C. Antfolk, C. Cipriani, F. Sebelius, G. Lundborg, "Referral of sensation to an advanced humanoid robotic hand prosthesis", *Scand J Plast Reconstr Surg Hand Surg*, 2009;43(5):260-6.
- [12] M. E. Wewers and M. K. Lowe, "A critical review of visual analogue scales in the measurement of clinical phenomena," *Res. Nursing Health*, vol. 13, pp. 227–236, 1990.
- [13] HH. Ehrsson, B. Rosén, A. Stockselius, C. Ragnö, P. Köhler, G. Lundborg, "Upper limb amputees can be induced to experience a rubber hand as their own", *Brain*, 2008 Dec;131:3443-52.

- [14] N. Miljković, N. Malešević, V. Kojić, G. Bijelić, T. Keller, DB Popović, "Recording and assessment of evoked potentials with electrode arrays", *Med. Biol. Eng. Comput.* 2015 Sep; 53(9):857-67.
- [15] K. A. Kaczmarek, J. G. Webster, P. Bach-y-Rita, and W. J. Tompkins, "Electrotactile and vibrotactile displays for sensory substitution systems," *IEEE Transactions on Bio-Medical Engineering*, vol. 38, no. 1, pp. 1-16, Jan. 1991.
- [16] R. Mann and S. Reimers, "Kinesthetic sensing for EMG controlled Boston Arm," *IEEE Transactions on Man-Machine Systems*, vol. 11, no. 1, pp. 110-&, 1970.
- [17] R. E. Prior, J. Lyman, P. A. Case, and C. M. Scott, "Supplemental sensory feedback for the VA/NU myoelectric hand. Background and preliminary designs.," *Bulletin of prosthetics research*, 1976.
- [18] P. L. L. B. (Author), et al PT, *Neuro Muscular Electrical Stimulation: A Practical Guide* (4th Edition) [STUDENT EDITION], 4th ed. Los Amigos Research Institute, Inc., 2000.
- [19] L. Seminara, L. Pinna, M. Valle, L. Basiricò, A. Loi, P. Cosseddu, A. Bonfiglio, A. Ascia, M. Biso, A. Ansaldo, D. Ricci, G. Metta, "Piezoelectric polymer transducer arrays for flexible tactile sensors," *IEEE Sensors Journal*, vol. 13, no. 10, 2013, pp. 4022-4029.
- [20] Measurement Specialties Inc. Available: <http://www.meas-spec.com/default.aspx>
- [21] SPES Medica, Genoa, Italy. Available: <http://www.spesmedica.com>

# Appendix B

## IntFES v2 Stimulator

The IntFES v2 stimulator prototype (Tecnalia, San Sebastian, ES) has been adapted to be used for electrotactile feedback applications in the MaxSENS project ([104,250,251]). Specifications of the IntFES v2 stimulator are reported in Table B.1.



Table B.1: Technical Characteristics of MaxSens Stimulator

|                              |                        |
|------------------------------|------------------------|
| Display                      | TFT 320x240            |
| Touch Panel Type             | Resistive              |
| Battery                      | 2xLi-Ion 3.7V 1400 mAh |
| Charger                      | 12V DC                 |
| Bluetooth                    | 2.1                    |
| Bluetooth range              | 10m                    |
| CAN Bus                      | 1.0/2.0                |
| Number of electrodes         | 1 to 4                 |
| Channels for each electrode  | 1 to 16                |
| Pulse instensity (amplitude) | 1 to 5mA               |
| Pulse width                  | 50 o 1000us            |
| Pulse rate                   | 1 to 400Hz             |
| Number of pulses             | 1 to INF               |
| Delay of stimulation         | 0 to 50000ms           |
| Stimulation voltage          | 95V                    |
| Rising Factor                | 0 to 99 no. of pulses  |
| Falling Factor               | 0 to 99 no. of pulses  |
| Time between Plets           | 1 to 99ms              |
| Size                         | 96mm/154mm/34mm L/W/H  |

# Bibliography

- [1] M. A. Oskoei and H. Hu, “Myoelectric control systemsA survey,” *Biomedical Signal Processing and Control*, vol. 2, no. 4, pp. 275 – 294, 2007.
- [2] D. S. Childress, “Closed-loop Control in Prosthetic Systems: Historical Perspective,” *Annals of Biomedical Engineering*, vol. 8, no. 4, pp. 293–303, 1980.
- [3] *Vincent Evolution 2 Hand*, available at <http://vincentsystems.de/en/prosthetics/vincent-evolution-2/>.
- [4] C. Antfolk *et al.*, “Sensory feedback in upper limb prosthetics,” *Expert review of medical devices*, vol. 10, no. 1, pp. 45–54, 2013.
- [5] M. I. Tiwana, S. J. Redmond, and N. H. Lovell, “A review of tactile sensing technologies with applications in biomedical engineering,” *Sensors and Actuators A: physical*, vol. 179, pp. 17–31, 2012.
- [6] A. Chaudhuri, *Fundamentals of sensory perception*. Oxford University Press Don Mills, 2011.
- [7] R. S. Dahiya *et al.*, “Tactile sensingfrom humans to humanoids,” *IEEE transactions on robotics*, vol. 26, no. 1, pp. 1–20, 2010.
- [8] Z. Kappassov, J.-A. Corrales, and V. Perdereau, “Tactile sensing in dexterous robot hands,” *Robotics and Autonomous Systems*, vol. 74, pp. 195–220, 2015.
- [9] B. A. Kent, J. Lavery, and E. D. Engeberg, “Anthropomorphic control of a dexterous artificial hand via task dependent temporally synchronized synergies,” *Journal of Bionic Engineering*, vol. 11, no. 2, pp. 236–248, 2014.
- [10] A. Hamed *et al.*, “Advances in haptics, tactile sensing, and manipulation for robot-assisted minimally invasive surgery, noninvasive surgery, and diagnosis,” *Journal of Robotics*, vol. 2012, 2012.

- [11] M. T. Francomano, D. Accoto, and E. Guglielmelli, "Artificial sense of slip: a review," *IEEE Sensors Journal*, vol. 13, no. 7, pp. 2489–2498, 2013.
- [12] R. Dahiya *et al.*, "Directions toward effective utilization of tactile skin: A review," *IEEE Sensors Journal*, vol. 13, no. 11, pp. 4121–4138, 2013.
- [13] H. Yousef, M. Boukallel, and K. Althoefer, "Tactile sensing for dexterous in-hand manipulation in robotics: a review," *Sensors and Actuators A: physical*, vol. 167, no. 2, pp. 171–187, 2011.
- [14] J. Loomis *et al.*, "Cognitive processes performances," *Tactual Perception*, vol. 2, p. 31, 1986.
- [15] M. S. Graziano and M. M. Botvinick, "How the brain represents the body: insights from neurophysiology and psychology," *Common mechanisms in perception and action: Attention and performance XIX*, pp. 136–157, 2002.
- [16] R. L. Klatzky and S. J. Lederman, "Touch," 2003.
- [17] 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," *Experimental brain research*, vol. 56, no. 3, pp. 550–564, 1984.
- [18] R. S. Johansson and A. Vallbo, "Tactile sensibility in the human hand: relative and absolute densities of four types of mechanoreceptive units in glabrous skin." *The Journal of physiology*, vol. 286, no. 1, pp. 283–300, 1979.
- [19] Å. Vallbo and K.-E. Hagbarth, "Activity from skin mechanoreceptors recorded percutaneously in awake human subjects," *Experimental neurology*, vol. 21, no. 3, pp. 270–289, 1968.
- [20] S. Najarian, J. Dargahi, and A. A. Mehrizi, *Artificial tactile sensing in biomedical engineering*. McGraw Hill Professional, 2009.
- [21] N. Jamali and C. Sammut, "Material classification by tactile sensing using surface textures," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE, 2010, pp. 2336–2341.
- [22] J. Dargahi *et al.*, *Tactile Sensing and Displays: Haptic Feedback for Minimally Invasive Surgery and Robotics*. John Wiley & Sons, 2012.

- [23] J. Tegin and J. Wikander, "Tactile sensing in intelligent robotic manipulation—a review," *Industrial Robot: An International Journal*, vol. 32, no. 1, pp. 64–70, 2005.
- [24] G. Schwartz *et al.*, "Flexible polymer transistors with high pressure sensitivity for application in electronic skin and health monitoring," *Nature communications*, vol. 4, p. 1859, 2013.
- [25] F. W. Clippinger, "A Sensory Feedback System for an Upper-limb Amputation Prosthesis," *Bulletin of Prosthetics Research Fall 1974*, pp. 247–258, 1974.
- [26] B. Crone *et al.*, "Large-scale complementary integrated circuits based on organic transistors," *Nature*, vol. 403, no. 6769, p. 521, 2000.
- [27] S. P. Lacour, S. Wagner, Z. Huang, and Z. Suo, "Stretchable gold conductors on elastomeric substrates," *Applied physics letters*, vol. 82, no. 15, pp. 2404–2406, 2003.
- [28] D.-H. Kim *et al.*, "Materials and noncoplanar mesh designs for integrated circuits with linear elastic responses to extreme mechanical deformations," *Proceedings of the National Academy of Sciences*, vol. 105, no. 48, pp. 18 675–18 680, 2008.
- [29] T. Someya *et al.*, "Conformable, flexible, large-area networks of pressure and thermal sensors with organic transistor active matrixes," *Proceedings of the National Academy of Sciences*, vol. 102, no. 35, pp. 12 321–12 325, 2005.
- [30] C. Metzger *et al.*, "Flexible-foam-based capacitive sensor arrays for object detection at low cost," *Applied Physics Letters*, vol. 92, no. 1, p. 013506, 2008.
- [31] I. Graz *et al.*, "Flexible ferroelectret field-effect transistor for large-area sensor skins and microphones," *Applied Physics Letters*, vol. 89, no. 7, p. 073501, 2006.
- [32] S. C. B. Mannsfeld *et al.*, "Highly sensitive flexible pressure sensors with microstructured rubber dielectric layers," *Nature materials*, vol. 9 10, pp. 859–64, 2010.
- [33] B. C. K. Tee *et al.*, "An electrically and mechanically self-healing composite with pressure- and flexion-sensitive properties for electronic skin applications," *Nature nanotechnology*, vol. 7 12, pp. 825–32, 2012.

- [34] Z. Yu *et al.*, “Intrinsically stretchable polymer light-emitting devices using carbon nanotube-polymer composite electrodes,” *Advanced Materials*, vol. 23, no. 34, pp. 3989–3994.
- [35] D. J. Lipomi *et al.*, “Stretchable organic solar cells,” *Advanced Materials*, vol. 23, no. 15, pp. 1771–1775.
- [36] T. Sekitani *et al.*, “Flexible organic transistors and circuits with extreme bending stability,” *Nature materials*, vol. 9 12, pp. 1015–22, 2010.
- [37] J. A. Rogers, T. Someya, and Y. Huang, “Materials and mechanics for stretchable electronics,” *Science*, vol. 327 5973, pp. 1603–7, 2010.
- [38] M. L. Hammock *et al.*, “25th anniversary article: the evolution of electronic skin (e-skin): a brief history, design considerations, and recent progress,” *Advanced materials*, vol. 25, no. 42, pp. 5997–6038, 2013.
- [39] P. Mittendorfer and G. Cheng, “Integrating discrete force cells into multi-modal artificial skin,” in *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*, Nov 2012, pp. 847–852.
- [40] F. Bossuyt, T. Vervust, and J. Vanfleteren, “Stretchable electronics technology for large area applications: Fabrication and mechanical characterization,” *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 3, pp. 229–235, 2013.
- [41] J. van den Brand *et al.*, “Flexible and stretchable electronics for wearable health-care,” in *Proceedings of the European Solid-State Device Research Conference*, 2014, pp. 206–209.
- [42] J. A. Rogers, T. Someya, and Y. Huang, “Materials and mechanics for stretchable electronics,” *Science*, vol. 327, no. 5973, pp. 1603–1607, 2010.
- [43] J. Kim *et al.*, “Stretchable silicon nanoribbon electronics for skin prosthesis,” *Nature Communications*, vol. 5, p. 5747, 2014.
- [44] V. J. Lumelsky, M. S. Shur, and S. Wagner, “Sensitive skin,” *IEEE Sensors Journal*, vol. 1, pp. 41–51, 2001.
- [45] K. Li, Y. Fang, Y. Zhou, and H. Liu, “Non-invasive stimulation-based tactile sensation for upper-extremity prosthesis: a review,” *IEEE Sensors Journal*, vol. 17, no. 9, pp. 2625–2635, 2017.

- [46] R. Puers, “Capacitive sensors: When and how to use them,” *Sensors and Actuators A: Physical*, vol. 37-38, pp. 93 – 105, 1993.
- [47] H. Muhammad *et al.*, “A capacitive tactile sensor array for surface texture discrimination,” *Microelectronic Engineering*, vol. 88, no. 8, pp. 1811 – 1813, 2011, proceedings of the 36th International Conference on Micro- and Nano-Engineering (MNE).
- [48] M. yuan Cheng *et al.*, “Article a polymer-based capacitive sensing array for normal and shear force measurement,” 2010.
- [49] M. Wegener, W. Wirges, and R. Gerhard-Multhaupt, “Piezoelectric polyethylene terephthalate (petp) foams specifically designed and prepared ferroelectret films,” *Advanced Engineering Materials*, vol. 7, no. 12, pp. 1128–1131.
- [50] “Tactile sensor arrays using fiber bragg grating sensors,” *Sensors and Actuators A: Physical*, vol. 126, no. 2, pp. 312 – 327, 2006.
- [51] G. M. Krishna and K. Rajanna, “Tactile sensor based on piezoelectric resonance,” *IEEE Sensors Journal*, vol. 4, no. 5, pp. 691–697, Oct 2004.
- [52] R. S. Dahiya, M. Valle, and G. Metta, “System approach: A paradigm for robotic tactile sensing,” *2008 10th IEEE International Workshop on Advanced Motion Control*, pp. 110–115, 2008.
- [53] F. Vidal-Verdú *et al.*, “A large area tactile sensor patch based on commercial force sensors,” in *Sensors*, 2011.
- [54] Y. Ohmura, Y. Kuniyoshi, and A. Nagakubo, “Conformable and scalable tactile sensor skin for curved surfaces,” in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, May 2006, pp. 1348–1353.
- [55] D. Goger, N. Gorges, and H. Worn, “Tactile sensing for an anthropomorphic robotic hand: Hardware and signal processing,” in *Robotics and Automation (ICRA), IEEE Conference*, 2009.
- [56] P. Mittendorfer and G. Cheng, “Humanoid multimodal tactile-sensing modules,” *IEEE Transactions on Robotics*, vol. 27, no. 3, pp. 401–410, June 2011.

- [57] B. Peerdeman *et al.*, “Myoelectric Forearm Prostheses: State of the Art from a User-centered Perspective.” *Journal of Rehabilitation Research & Development*, vol. 48, no. 6, 2011.
- [58] H. P. Saal and S. J. Bensmaia, “Biomimetic approaches to bionic touch through a peripheral nerve interface,” *Neuropsychologia*, vol. 79, pp. 344–353, 2015.
- [59] S. Raspopovic *et al.*, “Restoring natural sensory feedback in real-time bidirectional hand prostheses,” *Science translational medicine*, vol. 6, no. 222, pp. 222ra19–222ra19, 2014.
- [60] S. Micera *et al.*, “Decoding Information from Neural Signals Recorded using Intraneural Electrodes: toward the Development of a Neurocontrolled Hand Prosthesis,” *Proceedings of the IEEE*, vol. 98, no. 3, pp. 407–417, 2010.
- [61] G. A. Tabot *et al.*, “Restoring the sense of touch with a prosthetic hand through a brain interface,” *Proceedings of the National Academy of Sciences*, p. 201221113, 2013.
- [62] R. S. Johansson and J. R. Flanagan, “Coding and Use of Tactile Signals from the Fingertips in Object Manipulation Tasks,” *Nature Reviews Neuroscience*, vol. 10, no. 5, p. 345, 2009.
- [63] M. Botvinick and J. Cohen, “Rubber Hands feelTouch that Eyes See,” *Nature*, vol. 391, no. 6669, p. 756, 1998.
- [64] S. Shimada, K. Fukuda, and K. Hiraki, “Rubber Hand Illusion under Delayed Visual Feedback,” *PloS one*, vol. 4, no. 7, p. e6185, 2009.
- [65] R. Johansson and I. Birznieks, “First spikes in ensembles of human tactile afferents code complex spatial fingertip events,” *Nature neuroscience*, vol. 7, pp. 170–7, 03 2004.
- [66] S. Banzi, E. Mainardi, and A. Davalli, “Analisi delle strategie di controllo per protesi di arto superiore in pazienti con amputazioni transomerale o disarticolati di spalla,” pp. 4290–300, 2005.
- [67] U. P. Services, “Upper limb amputations.”
- [68] Ziegler-Graham *et al.*, “Estimating the prevalence of limb loss in the united states: 2005 to 2050,” vol. 89, pp. 422–9, 04 2008.

- [69] A. M. Agur and A. Dalley, “Grant’s Atlas of Anatomy. 13th,” 1999.
- [70] J. T. Belter *et al.*, “Mechanical Design and Performance Specifications of Anthropomorphic Prosthetic Hands: a Review,” *J Rehabil Res Dev*, vol. 50, no. 5, pp. 599–618, 2013.
- [71] D. J. Atkins, D. C. Heard, and W. H. Donovan, “Epidemiologic Overview of Individuals with Upper-limb Loss and their reported Research Priorities,” *JPO: Journal of Prosthetics and Orthotics*, vol. 8, no. 1, pp. 2–11, 1996.
- [72] E. A. Biddiss and T. T. Chau, “Upper Limb Prosthesis Use and Abandonment: a Survey of the last 25 years,” *Prosthetics and Orthotics International*, vol. 31, no. 3, pp. 236–257, 2007.
- [73] G. Lundborg and B. Rosen, “Sensory Substitution in Prosthetics.” *Hand Clinics*, vol. 17, no. 3, pp. 481–8, 2001.
- [74] M. Rakić, “The Belgrade Hand Prosthesis,” in *Proceedings of the Institution of Mechanical Engineers, Conference Proceedings*, vol. 183, no. 10, 1968, pp. 60–67.
- [75] J. Rothwell *et al.*, “Manual Motor Performance in a Deafferented Man,” *Brain*, vol. 105, no. 3, pp. 515–542, 1982.
- [76] R. S. Johansson and K. J. Cole, “Sensory-motor coordination during grasping and manipulative actions,” *Current opinion in neurobiology*, vol. 2, no. 6, pp. 815–823, 1992.
- [77] E. L. Graczyk *et al.*, “The neural basis of perceived intensity in natural and artificial touch,” *Science Translational Medicine*, vol. 8, no. 362, pp. 362ra142–362ra142, 2016.
- [78] J. S. Schofield *et al.*, “Applications of Sensory Feedback in Motorized Upper Extremity Prosthesis: a Review,” *Expert Review of Medical Devices*, vol. 11, no. 5, pp. 499–511, 2014.
- [79] S. Micera, J. Carpaneto, and S. Raspopovic, “Control of Hand Prostheses using Peripheral Information,” *IEEE Reviews in Biomedical Engineering*, vol. 3, pp. 48–68, 2010.
- [80] A. Benvenuto *et al.*, “Intrafascicular thin-film multichannel electrodes for sensory feedback: Evidences on a human amputee,” in *Conf Proc IEEE Eng Med Biol Soc*, vol. 2010, 2010, pp. 1800–1803.



- [81] K. Horch *et al.*, “Object Discrimination with an Artificial Hand using Electrical Stimulation of Peripheral Tactile and Proprioceptive Pathways with Entrafascicular Electrodes,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 19, no. 5, pp. 483–489, 2011.
- [82] P. M. Rossini *et al.*, “Double Nerve Intraneural Interface Implant on a Human Amputee for Robotic Hand Control,” *Clinical Neurophysiology*, vol. 121, no. 5, pp. 777–783, 2010.
- [83] G. S. Dhillon and K. W. Horch, “Direct Neural Sensory Feedback and Control of a Prosthetic Arm,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 13, no. 4, pp. 468–472, 2005.
- [84] G. Dhillon *et al.*, “Effects of Short-term Training on Sensory and Motor Function in Severed Nerves of Long-term Human Amputees,” *Journal of Neurophysiology*, vol. 93, no. 5, pp. 2625–2633, 2005.
- [85] R. R. Riso, “Strategies for Providing Upper Extremity Amputees with Tactile and Hand Position Feedback—moving Closer to the Bionic Arm,” *Technology and Health Care*, vol. 7, no. 6, pp. 401–409, 1999.
- [86] A. Anani and L. Körner, “Discrimination of Phantom Hand Sensations elicited by Afferent Electrical Nerve Stimulation in Below-elbow Amputees,” *Medical Progress through Technology*, vol. 6, no. 3, pp. 131–135, 1979.
- [87] B. T. Nghiem, I. C. Sando, R. B. Gillespie, B. L. McLaughlin, G. J. Gerling, N. B. Langhals, M. G. Urbanek, and P. S. Cederna, “Providing a sense of touch to prosthetic hands,” *Plastic and reconstructive surgery*, vol. 135 6, pp. 1652–63, 2015.
- [88] T. A. Kuiken *et al.*, “The use of targeted muscle reinnervation for improved myoelectric prosthesis control in a bilateral shoulder disarticulation amputee,” *Prosthetics and orthotics international*, vol. 28, no. 3, pp. 245–253, 2004.
- [89] J. B. Hijjawi *et al.*, “Improved myoelectric prosthesis control accomplished using multiple nerve transfers,” *Plastic and reconstructive surgery*, vol. 118, no. 7, pp. 1573–1578, 2006.
- [90] J. S. Hebert *et al.*, “Novel targeted sensory reinnervation technique to restore functional hand sensation after transhumeral amputation,” *IEEE Transactions*

- on *Neural Systems and Rehabilitation Engineering*, vol. 22, no. 4, pp. 765–773, 2014.
- [91] T. A. Kuiken *et al.*, “Redirection of cutaneous sensation from the hand to the chest skin of human amputees with targeted reinnervation,” *Proceedings of the National Academy of Sciences*, vol. 104, no. 50, pp. 20 061–20 066, 2007.
  - [92] T. A. Kuiken *et al.*, “Targeted reinnervation for enhanced prosthetic arm function in a woman with a proximal amputation: a case study,” *The Lancet*, vol. 369, no. 9559, pp. 371–380, 2007.
  - [93] J. S. Hebert *et al.*, “Updates in targeted sensory reinnervation for upper limb amputation,” *Current Surgery Reports*, vol. 2, no. 3, p. 45, 2014.
  - [94] B. Stephens-Fripp, G. Alici, and R. Mutlu, “A Review of Non-Invasive Sensory Feedback Methods for Transradial Prosthetic Hands,” *IEEE Access*, vol. 6, pp. 6878–6899, 2018.
  - [95] C. Cipriani, M. D’Alonzo, and M. C. Carrozza, “A miniature vibrotactile sensory substitution device for multifingered hand prosthetics,” *IEEE transactions on biomedical engineering*, vol. 59, no. 2, pp. 400–408, 2012.
  - [96] L. A. Jones and N. B. Sarter, “Tactile displays: Guidance for their design and application,” *Human factors*, vol. 50, no. 1, pp. 90–111, 2008.
  - [97] C. E. Stepp, Q. An, and Y. Matsuoka, “Repeated training with augmentative vibrotactile feedback increases object manipulation performance,” *PloS one*, vol. 7, no. 2, p. e32743, 2012.
  - [98] H. J. Witteveen *et al.*, “Vibro-and electrotactile user feedback on hand opening for myoelectric forearm prostheses,” *IEEE transactions on biomedical engineering*, vol. 59, no. 8, pp. 2219–2226, 2012.
  - [99] M. A. F. Ismail and S. Shimada, “robothand illusion under delayed visual feedback: Relationship between the senses of ownership and agency,” *PloS one*, vol. 11, no. 7, p. e0159619, 2016.
  - [100] J. Cohen *et al.*, “A closed-loop tactor frequency control system for vibrotactile feedback,” in *CHI’05 Extended Abstracts on Human Factors in Computing Systems*. ACM, 2005, pp. 1296–1299.

- [101] K. A. Kaczmarek *et al.*, “Electrotactile and vibrotactile displays for sensory substitution systems,” *IEEE Transactions on Biomedical Engineering*, vol. 38, no. 1, pp. 1–16, 1991.
- [102] A. Y. Szeto and F. A. Saunders, “Electrocutaneous stimulation for sensory communication in rehabilitation engineering,” *IEEE Transactions on Biomedical Engineering*, no. 4, pp. 300–308, 1982.
- [103] N. Jorgovanovic *et al.*, “Virtual grasping: closed-loop force control using electrotactile feedback,” *Computational and mathematical methods in medicine*, vol. 2014, 2014.
- [104] M. Štrbac *et al.*, “Integrated and flexible multichannel interface for electrotactile stimulation,” *Journal of neural engineering*, vol. 13, no. 4, p. 046014, 2016.
- [105] K. Choi *et al.*, “Two-channel electrotactile stimulation for sensory feedback of fingers of prosthesis,” in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*. IEEE, 2016, pp. 1133–1138.
- [106] S. G. Meek, S. C. Jacobsen, and P. P. Goulding, “Extended physiologic taction: design and evaluation of a proportional force feedback system,” *J Rehabil Res Dev*, vol. 26, no. 3, pp. 53–62, 1989.
- [107] C. Antfolk *et al.*, “Sensory feedback from a prosthetic hand based on air-mediated pressure from the hand to the forearm skin,” *Journal of rehabilitation medicine*, vol. 44, no. 8, pp. 702–707, 2012.
- [108] C. E. Stepp and Y. Matsuoka, “Relative to direct haptic feedback, remote vibrotactile feedback improves but slows object manipulation,” in *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE*. IEEE, 2010, pp. 2089–2092.
- [109] S. Casini *et al.*, “Design and realization of the cuff-clenching upper-limb force feedback wearable device for distributed mechano-tactile stimulation of normal and tangential skin forces,” in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE, 2015, pp. 1186–1193.
- [110] R. S. Armiger *et al.*, “Enabling closed-loop control of the modular prosthetic limb through haptic feedback,” *Johns Hopkins APL Tech Dig*, vol. 31, no. 4, pp. 345–53, 2013.

- [111] *Robotics research. HDT global*, available at [www.hdtglobal.com/services/robotics/Research/](http://www.hdtglobal.com/services/robotics/Research/).
- [112] S. Wilson and S. Dirven, "Audio sensory substitution for human-in-the-loop force feedback of upper limb prosthetics," in *Mechatronics and Machine Vision in Practice (M2VIP), 2016 23rd International Conference on*. IEEE, 2016, pp. 1–6.
- [113] J. o. Gonzalez, "Auditory display as a prosthetic hand sensory feedback for reaching and grasping tasks," in *Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE*. IEEE, 2012, pp. 1789–1792.
- [114] M. Markovic *et al.*, "Glimpse: Google glass interface for sensory feedback in myoelectric hand prostheses," *Journal of neural engineering*, vol. 14, no. 3, p. 036007, 2017.
- [115] F. Clemente *et al.*, "Humans can integrate augmented reality feedback in their sensorimotor control of a robotic hand," *IEEE Transactions on Human-Machine Systems*, vol. 47, no. 4, pp. 583–589, 2017.
- [116] C. Pylatiuk, A. Kargov, and S. Schulz, "Design and evaluation of a low-cost force feedback system for myoelectric prosthetic hands," *JPO: Journal of Prosthetics and Orthotics*, vol. 18, no. 2, pp. 57–61, 2006.
- [117] M. Markovic *et al.*, "The clinical relevance of advanced artificial feedback in the control of a multi-functional myoelectric prosthesis," *Journal of NeuroEngineering and Rehabilitation*, vol. 15, no. 1, p. 28, Mar 2018. [Online]. Available: <https://doi.org/10.1186/s12984-018-0371-1>
- [118] I. Saunders and S. Vijayakumar, "The role of feed-forward and feedback processes for closed-loop prosthesis control," *Journal of neuroengineering and rehabilitation*, vol. 8, no. 1, p. 60, 2011.
- [119] S. Dosen *et al.*, "Multichannel electrotactile feedback with spatial and mixed coding for closed-loop control of grasping force in hand prostheses," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 25, no. 3, pp. 183–195, 2017.

- [120] M. Franceschi *et al.*, “A system for electrotactile feedback using electronic skin and flexible matrix electrodes: Experimental evaluation,” *IEEE Transactions on Haptics*, vol. 10, no. 2, April 2017.
- [121] C. Hartmann *et al.*, “Towards prosthetic systems providing comprehensive tactile feedback for utility and embodiment,” in *2014 IEEE Biomedical Circuits and Systems Conference (BioCAS) Proceedings*, Oct 2014, pp. 620–623.
- [122] S. Dosen *et al.*, “A novel method to generate amplitude-frequency modulated vibrotactile stimulation,” *IEEE Transactions on Haptics*, vol. 9, no. 1, pp. 3–12, Jan 2016.
- [123] N. M. Malešević *et al.*, “A multi-pad electrode based functional electrical stimulation system for restoration of grasp,” *Journal of neuroengineering and rehabilitation*, vol. 9, no. 1, p. 66, 2012.
- [124] A. Kuhn and T. Keller, “The influence of capacitive properties on nerve activation in transcutaneous electrical stimulation,” in *7th International Symposium on Computer Methods in Biomechanics and Biomedical Engineering*, ser. International Symposium on Computer Methods in Biomechanics and Biomedical Engineering, J. Middleton, Ed., vol. 7, 2007, pp. 151MAS – 153MAS.
- [125] “Valutrode foam (axelgaard manufacturing co., ltd., us).”
- [126] M. Solomonow, J. Lyman, and A. Freedy, “Electrotactile two-point discrimination as a function of frequency, body site, laterality, and stimulation codes,” *Annals of biomedical engineering*, vol. 5, no. 1, pp. 47–60, 1977.
- [127] D. G. Buma *et al.*, “Intermittent stimulation delays adaptation to electrocutaneous sensory feedback,” *IEEE transactions on neural systems and rehabilitation engineering*, vol. 15, pp. 435–441, 2007.
- [128] K. A. Kaczmarek, “Electrotactile adaptation on the abdomen: preliminary results,” *IEEE Transactions on Rehabilitation Engineering*, vol. 8, no. 4, pp. 499–505, Dec 2000.
- [129] A. N. S. D. T. Y. H. Dietl and D. Farina, “Design and evaluation of a novel wearable haptic device for amplitude-frequency modulated vibro-tactile stimulation,” *IEEE Transactions on Haptics*, 2014.

- [130] A. Y. J. Szeto and J. Lyman, “Comparison of codes for sensory feedback using electrocutaneous tracking,” *Annals of Biomedical Engineering*, vol. 5, no. 4, pp. 367–383, Dec 1977.
- [131] A. Szeto and L. Mao, “Dermal effects of electrocutaneous stimulation,” in *Biomedical Engineering I*, S. Saha, Ed., 1982, pp. 121 – 124.
- [132] H. H. Ehrsson *et al.*, “Upper limb amputees can be induced to experience a rubber hand as their own.” *Brain: A Journal of Neurology*, vol. 131, no. 5, pp. 3443–52, 2008-12-11 00:00:00.0.
- [133] P. R. Davidson and D. M. Wolpert, “Widespread access to predictive models in the motor system: a short review,” *Journal of Neural Engineering*, vol. 2, no. 3, 2005. [Online]. Available: <http://stacks.iop.org/1741-2552/2/i=3/a=S11>
- [134] D. Kording, Konrad & Wolpert, “Bayesian integration in sensorimotor learning,” *Nature*, no. 427, pp. 244–247, 2004.
- [135] P. Parker, K. Englehart, and B. Hudgins, “Myoelectric signal processing for control of powered limb prostheses,” *Journal of Electromyography and Kinesiology*, vol. 16, no. 6, pp. 541 – 548, 2006, special Section (pp. 541610): 2006 ISEK Congress.
- [136] N. Jiang *et al.*, “Myoelectric control of artificial limbs is there a need to change focus?[in the spotlight],” *IEEE Signal Processing Magazine*, vol. 29, no. 5, pp. 152–150, 2012.
- [137] E. Biddiss, D. Beaton, and T. Chau, “Consumer design priorities for upper limb prosthetics,” *Disability and Rehabilitation: Assistive Technology*, vol. 2, no. 6, pp. 346–357, 2007.
- [138] P. Bach-y Rita and S. W. Kercel, “Sensory substitution and the human–machine interface,” *Trends in cognitive sciences*, vol. 7, no. 12, pp. 541–546, 2003.
- [139] *Otto Bock Michelangelo Hand*, available at <http://www.living-with-michelangelo.com/gb/home/>.
- [140] *BioTAC SynTouch Company*, available at <http://www.syntouchllc.com/Products/BioTac/>.

- [141] T. Sekitani and T. Someya, “Stretchable, large-area organic electronics,” *Advanced Materials*, vol. 22, no. 20, pp. 2228–2246, 2010.
- [142] T. Someya *et al.*, “Conformable, flexible, large-area networks of pressure and thermal sensors with organic transistor active matrixes,” *Proceedings of the National Academy of Sciences*, vol. 102, no. 35, pp. 12 321–12 325, 2005.
- [143] N. Wettels *et al.*, “Biomimetic tactile sensor array,” *Advanced Robotics*, vol. 22, no. 8, pp. 829–849, 2008.
- [144] K. Kim *et al.*, “Polymer-based flexible tactile sensor up to  $32 \times 32$  arrays integrated with interconnection terminals,” *Sensors and Actuators A: Physical*, vol. 156, no. 2, pp. 284–291, 2009.
- [145] Y.-J. Yang *et al.*, “An integrated flexible temperature and tactile sensing array using pi-copper films,” *Sensors and Actuators A: Physical*, vol. 143, no. 1, pp. 143–153, 2008.
- [146] C. Lucarotti *et al.*, “Synthetic and bio-artificial tactile sensing: A review,” *Sensors*, vol. 13, no. 2, pp. 1435–1466, 2013.
- [147] E. D. Engeberg and S. Meek, “Enhanced visual feedback for slip prevention with a prosthetic hand,” *Prosthetics and orthotics international*, vol. 36, no. 4, pp. 423–429, 2012.
- [148] P. Saccomandi *et al.*, “Microfabricated tactile sensors for biomedical applications: a review,” *Biosensors*, vol. 4, no. 4, pp. 422–448, 2014.
- [149] J. Farserotu *et al.*, “Tactile prosthetics in wisekin,” in *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2015*. IEEE, 2015, pp. 1695–1697.
- [150] J. Farserotu *et al.*, “Smart skin for tactile prosthetics,” in *Medical Information and Communication Technology (ISMICT), 2012 6th International Symposium on*. IEEE, 2012, pp. 1–8.
- [151] M. D’Alonzo, F. Clemente, and C. Cipriani, “Vibrotactile stimulation promotes embodiment of an alien hand in amputees with phantom sensations,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 23, no. 3, pp. 450–457, 2015.

- [152] B. C.-K. Tee, *et al.*, “A skin-inspired organic digital mechanoreceptor,” *Science*, vol. 350, no. 6258, pp. 313–316, 2015.
- [153] H. S. Nalwa, *Ferroelectric polymers: chemistry: physics, and applications*. CRC Press, 1995.
- [154] *SPES Medica, Genoa, Italy*, available at <http://www.spesmedica.com>.
- [155] *Measurement Specialties Inc.*, available at <http://www.meas-spec.com/default.aspx>.
- [156] L. Seminara *et al.*, “Piezoelectric polymer transducer arrays for flexible tactile sensors,” *IEEE Sensors Journal*, vol. 13, no. 10, pp. 4022–4029, 2013.
- [157] L. Pinna and M. Valle, “Charge amplifier design methodology for pvdF-based tactile sensors,” *Journal of Circuits, Systems and Computers*, vol. 22, no. 08, p. 1350066, 2013.
- [158] M. Perovic *et al.*, “Electrical stimulation of the forearm: a method for transmitting sensory signals from the artificial hand to the brain,” *Journal of Automatic Control*, vol. 21, pp. 13–18, 2013.
- [159] B. Geng *et al.*, “Evaluation of sensation evoked by electrocutaneous stimulation on forearm in nondisabled subjects,” *Journal of Rehabilitation Research and Development*, vol. 49, pp. 297–308, 2012.
- [160] T. Tashiro and A. Higashiyama, “The perceptual properties of electrocutaneous stimulation: Sensory quality, subjective intensity, and intensity-duration relation,” *Perception & Psychophysics*, vol. 30, pp. 579–586, Nov 1981.
- [161] L. P. Paredes *et al.*, “The impact of the stimulation frequency on closed-loop control with electrotactile feedback,” *Journal of NeuroEngineering and Rehabilitation*, vol. 12, no. 1, p. 35, Apr 2015. [Online]. Available: <https://doi.org/10.1186/s12984-015-0022-8>
- [162] F. A. A. Kingdom and N. Prins, “Psychophysics: A practical introduction,” *Academic Press*.
- [163] D. Dreyer, M. Hollins, and B. Whitsel, “Factors influencing cutaneous directional sensitivity,” *Sensory processes*, vol. 2, no. 2, p. 7179, June 1978. [Online]. Available: <http://europepmc.org/abstract/MED/715470>



- [164] G. Essick and B. Whitsel, “Factors influencing cutaneous directional sensitivity: A correlative psychophysical and neurophysiological investigation,” *Brain Research Reviews*, vol. 10, no. 3, pp. 213 – 230, 1985. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/0165017385900256>
- [165] E. P. Gardner and B. F. Sklar, “Discrimination of the direction of motion on the human hand: a psychophysical study of stimulation parameters,” *Journal of Neurophysiology*, vol. 71, no. 6, pp. 2414–2429, 1994.
- [166] E. P. Gardner and C. I. Palmer, “Simulation of motion on the skin. i. receptive fields and temporal frequency coding by cutaneous mechanoreceptors of optacon pulses delivered to the hand,” *Journal of Neurophysiology*, vol. 62, no. 6, pp. 1410–1436, 1989.
- [167] E. P. Gardner and C. I. Palmer, “Simulation of motion on the skin. III. Mechanisms used by rapidly adapting cutaneous mechanoreceptors in the primate hand for spatiotemporal resolution and two-point discrimination,” *Journal of Neurophysiology*, vol. 63, no. 4, pp. 841–859, 1990.
- [168] —, “Simulation of motion on the skin. II. Cutaneous mechanoreceptor coding of the width and texture of bar patterns displaced across the OPTACON,” *Journal of Neurophysiology*, vol. 62, no. 6, pp. 1437–1460, 1989.
- [169] L. Shams and A. R. Seitz, “Benefits of multisensory learning,” *Trends in Cognitive Sciences*, vol. 12, no. 11, pp. 411 – 417, 2008. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1364661308002180>
- [170] C. D. Mrch *et al.*, “Exteroceptive aspects of nociception: Insights from graphesthesia and two-point discrimination,” *PAIN*, vol. 151, no. 1, pp. 45 – 52, 2010. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0304395910003192>
- [171] V. Drago *et al.*, “Graphesthesia: A test of graphemic movement representations or tactile imagery?” *Journal of the International Neuropsychological Society*, vol. 16, no. 1, p. 190193, 2010.
- [172] S. J. Loomis, J. M. Lederman, “Tactual perception,” in *Handbook of perception and human performance*, 1986.

- [173] C. E. Kops and E. P. Gardner, “Discrimination of simulated texture patterns on the human hand,” *Journal of Neurophysiology*, vol. 76, no. 2, pp. 1145–1165, 1996.
- [174] *Sensor Hand*, available at <http://www.ottobock.com.au/prosthetics/upper-limb/solution-overview/myoelectric-devices-speedhands/>.
- [175] D. D. Damian *et al.*, “Slip speed feedback for grip force control,” *IEEE Transactions on Biomedical Engineering*, vol. 59, no. 8, pp. 2200–2210, Aug 2012.
- [176] S. J. Lederman and R. L. Klatzky, “Hand movements: A window into haptic object recognition,” *Cognitive Psychology*, vol. 19, no. 3, pp. 342 – 368, 1987. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/0010028587900089>
- [177] D. Silvera Tawil, D. Rye, and M. Velonaki, “Interpretation of the modality of touch on an artificial arm covered with an EIT-based sensitive skin,” *International Journal of Robotics Research (IJRR)*, 2012.
- [178] S. A. Arabshahi and Z. Jiang, “Development of a tactile sensor for braille pattern recognition: sensor design and simulation,” *Smart Materials and Structures*, 2005.
- [179] S. Decherchi *et al.*, “Tactile-data classification of contact materials using computational intelligence,” *IEEE Trans. on Robotics*, 2011.
- [180] S.-H. Kim *et al.*, “Texture classification using a polymer-based mems tactile sensor,” *Journal of Micromechanics and Microengineering*, 2005.
- [181] H. Iwata and S. Sugano, “Human-robot-contact-state identification based on tactile recognition,” *IEEE Trans. on Industrial Electronics*, 2005.
- [182] P. Gastaldo, L. Pinna, L. Seminara, M. Valle, and R. Zunino, “A tensor-based pattern-recognition framework for the interpretation of touch modality in artificial skin systems,” *IEEE Sensors Journal*, 2014.
- [183] A. Ibrahim and M. Valle, “Real-time embedded machine learning for tensorial tactile data processing,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, no. 99, pp. 1–10, 2018.

- [184] B. Scholkopf and A. J. Smola, “Learning with kernels: support vector machines, regularization, optimization, and beyond (adaptive computation and machine learning),” *MIT Press, Cambridge, MA, USA*, vol. 12, p. 3, 2001.
- [185] M. Signoretto, L. De Lathauwer, and J. A. Suykens, “A kernel-based framework to tensorial data analysis,” *Neural networks*, vol. 24, no. 8, pp. 861–874, 2011.
- [186] L. Seminara *et al.*, “Towards integrating intelligence in electronic skin,” *Mechanics*, vol. 34, pp. 84–94, 2016.
- [187] A. Ibrahim *et al.*, “Assessment of fpga implementations of one sided jacobi algorithm for singular value decomposition,” in *VLSI (ISVLSI), 2015 IEEE Computer Society Annual Symposium on*. IEEE, 2015, pp. 56–61.
- [188] S. Kang, S.-Y. Wu, Y.-H. Hsu, C.-C. Fu, and W.-C. Fang, “A vlsi design of singular value decomposition processor for portable continuous-wave diffusion optical tomography systems,” in *Life Science Systems and Applications Workshop (LiSSA), 2011 IEEE/NIH*. IEEE, 2011, pp. 100–103.
- [189] S. Lahabar and P. Narayanan, “Singular value decomposition on gpu using cuda,” in *Parallel & Distributed Processing, 2009. IPDPS 2009. IEEE International Symposium on*. IEEE, 2009, pp. 1–10.
- [190] A. Haidar, J. Kurzak, and P. Luszczek, “An improved parallel singular value algorithm and its implementation for multicore hardware,” in *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*. ACM, 2013, p. 90.
- [191] X. Wang and J. Zambreno, “An fpga implementation of the hestenes-jacobi algorithm for singular value decomposition,” in *Parallel & Distributed Processing Symposium Workshops (IPDPSW), 2014 IEEE International*. IEEE, 2014, pp. 220–227.
- [192] J. Demmel and K. Veselić, “Jacobi’s method is more accurate than qr,” *SIAM Journal on Matrix Analysis and Applications*, vol. 13, no. 4, pp. 1204–1245, 1992.
- [193] V. Camus, J. Schlachter, C. Enz, M. Gautschi, and F. K. Gurkaynak, “Approximate 32-bit floating-point unit design with 53% power-area product reduction,” in *European Solid-State Circuits (ESSCIRC), IEEE Conference*, 2016.

- [194] I. Kuon and J. Rose, “Measuring the gap between fpgas and asics,” *IEEE Transactions on computer-aided design of integrated circuits and systems*, vol. 26, no. 2, pp. 203–215, 2007.
- [195] M. Magno, L. Spadaro, J. Singh, and L. Benini, “Kinetic energy harvesting: Toward autonomous wearable sensing for internet of things,” in *Power Electronics, Electrical Drives, Automation and Motion (SPEEDAM), 2016 International Symposium on*. IEEE, 2016, pp. 248–254.
- [196] L. Seminara, M. Franceschi, L. Pinna, A. Ibrahim, M. Valle, S. Dosen, and D. Farina, “Electronic skin and electrocutaneous stimulation to restore the sense of touch in hand prosthetics,” in *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2017, pp. 1–4.
- [197] T. Liu and S.-L. Lu, “Performance Improvement with Circuit-level Speculation,” in *Microarchitecture (MICRO-33), IEEE/ACM*, 2000.
- [198] J. Schlachter, V. Camus, C. Enz, and K. Palem, “Automatic generation of inexact digital circuits by gate-level pruning,” in *Circuits and Systems (ISCAS), 2015 IEEE International Symposium on*, 2015.
- [199] A. Ibrahim, P. Gastaldo, H. Chible, and M. Valle, “Real-time digital signal processing based on FPGAs for electronic skin implementation,” *Sensors*, 2017.
- [200] V. Camus, J. Schlachter, and C. Enz, “Energy-efficient inexact speculative adder with high performance and accuracy control,” in *Circuits and Systems (ISCAS), IEEE International Symposium*, 2015.
- [201] M. Franceschi *et al.*, “Approximate FPGA Implementation of CORDIC for Tactile Data Processing Using Speculative Adders,” in *2017 New Generation of CAS (NGCAS)*, Sep. 2017, pp. 41–44.
- [202] C. M. Kirsch and H. Payer, “Incorrect systems: it’s not the problem, it’s the solution,” in *Design Automation Conference (DAC), 2012 49th ACM/EDAC/IEEE*. IEEE, 2012, pp. 913–917.
- [203] S. Ghosh, S. Bhunia, and K. Roy, “Crista: A new paradigm for low-power, variation-tolerant, and adaptive circuit synthesis using critical path isolation,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 26, no. 11, pp. 1947–1956, 2007.

- [204] S. S. Basu, P. Garcia del Valle, G. Ansaloni, G. Karakonstantis, and D. Atienza Alonso, “Heterogeneous error-resilient scheme for spectral analysis in ultra-low power wearable electrocardiogram devices,” in *Proceedings of the IEEE Annual Symposium on VLSI 2015 (ISVLSI 2015)*, vol. 1, no. EPFL-CONF-206154. IEEE Press, 2015, pp. 173–180.
- [205] Q. Xu, T. Mytkowicz, and N. S. Kim, “Approximate computing: A survey,” *IEEE Design Test*, vol. 33, no. 1, pp. 8–22, Feb 2016.
- [206] I. Akturk, K. Khatamifard, and U. R. Karpuzcu, “On quantification of accuracy loss in approximate computing,” in *Workshop on Duplicating, Deconstructing and Debunking (WDDD)*, 2015, p. 15.
- [207] J. Ansel *et al.*, “Language and compiler support for auto-tuning variable-accuracy algorithms,” in *Code Generation and Optimization (CGO), 2011 9th Annual IEEE/ACM International Symposium on*. IEEE, 2011, pp. 85–96.
- [208] H. Esmailzadeh *et al.*, “Architecture Support for Disciplined Approximate Programming,” in *Proc. of the 17th International Conference on Architectural Support for Programming Languages and Operating Systems*, 2012, pp. 301–312.
- [209] R. St Amant *et al.*, “General-purpose code acceleration with limited-precision analog computation,” *ACM SIGARCH Computer Architecture News*, vol. 42, no. 3, pp. 505–516, 2014.
- [210] B. Li *et al.*, “Rram-based analog approximate computing,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 12, pp. 1905–1917, 2015.
- [211] S. Venkataramani *et al.*, “Axnn: energy-efficient neuromorphic systems using approximate computing,” in *Proceedings of the 2014 international symposium on Low power electronics and design*. ACM, 2014, pp. 27–32.
- [212] Q. Zhang *et al.*, “Approxann: an approximate computing framework for artificial neural network,” in *Proceedings of the 2015 Design, Automation & Test in Europe Conference & Exhibition*. EDA Consortium, 2015, pp. 701–706.
- [213] S. Sidiroglou-Douskos *et al.*, “Managing Performance vs. Accuracy Trade-offs with Loop Perforation,” in *Proceedings of the 19th ACM SIGSOFT Symposium and the 13th European Conference on Foundations of Software*

- Engineering*, ser. ESEC/FSE '11, 2011, pp. 124–134. [Online]. Available: <http://doi.acm.org/10.1145/2025113.2025133>
- [214] M. Samadi and S. Mahlke, “Cpu-gpu collaboration for output quality monitoring,” in *1st Workshop on Approximate Computing Across the System Stack*, 2014, pp. 1–3.
  - [215] A. B. Kahng and S. Kang, “Accuracy-configurable adder for approximate arithmetic designs,” in *Proceedings of the 49th Annual Design Automation Conference*. ACM, 2012, pp. 820–825.
  - [216] S. Venkataramani *et al.*, “Salsa: systematic logic synthesis of approximate circuits,” in *Design Automation Conference (DAC), 2012 49th ACM/EDAC/IEEE*. IEEE, 2012, pp. 796–801.
  - [217] S. Ganapathy *et al.*, “Mitigating the impact of faults in unreliable memories for error-resilient applications,” in *Proceedings of the 52nd Annual Design Automation Conference*. ACM, 2015, p. 102.
  - [218] S. Mittal, “A survey of architectural techniques for near-threshold computing,” *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, vol. 12, no. 4, p. 46, 2016.
  - [219] A. Sampson *et al.*, “Enerj: Approximate data types for safe and general low-power computation,” in *ACM SIGPLAN Notices*, vol. 46, no. 6. ACM, 2011, pp. 164–174.
  - [220] V. K. Chippa *et al.*, “Scalable effort hardware design,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, no. 9, pp. 2004–2016, 2014.
  - [221] H. Jiang, C. Liu, L. Liu, F. Lombardi, and J. Han, “A review, classification and comparative evaluation of approximate arithmetic circuits,” in *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, 2017.
  - [222] V. Camus, J. Schlachter, and C. Enz, “Energy-efficient digital design through inexact and approximate arithmetic circuits,” in *New Circuits and Systems Conference (NEWCAS), IEEE*, 2015, pp. 1–4.
  - [223] J. Schlachter, V. Camus, and C. Enz, “Near/sub-threshold circuits and approximate computing: The perfect combination for ultra-low-power systems,” in *VLSI (ISVLSI), IEEE Symposium*, 2015.

- [224] X. Jiao, V. Camus, M. Cacciotti, Y. Jiang, C. Enz, and R. Gupta, “Combining structural and timing errors in overclocked inexact speculative adders,” in *Design, Automation & Test in Europe (DATE), IEEE*, 2017.
- [225] N. Zhu, W.-L. Goh, and K.-S. Yeo, “An Enhanced Low-power High-speed Adder For Error-tolerant Application,” in *Integrated Circuits (ISIC), Proc. of the 2009 12th International Symposium on*, Dec 2009, pp. 69–72.
- [226] J. Schlachter, V. Camus, K. V. Palem, and C. Enz, “Design and applications of approximate circuits by gate-level pruning,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2017.
- [227] V. Camus, J. Schlachter, and C. Enz, “A low-power carry cut-back approximate adder with fixed-point implementation and floating-point precision,” 2016.
- [228] K. M. Reddy *et al.*, “Low power, high speed error tolerant multiplier using approximate adders,” in *VLSI Design and Test (VDAT), 2015 19th International Symposium on*. IEEE, 2015, pp. 1–6.
- [229] C. D. Singh and Y. Singh, “Error tolerant adder,” *International Journal of Scientific and Research Publications*, p. 249.
- [230] H. R. Mahdiani *et al.*, “Bio-inspired imprecise computational blocks for efficient vlsi implementation of soft-computing applications,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 57, no. 4, pp. 850–862, 2010.
- [231] A. Ibrahim, M. Valle, L. Noli, and H. Chible, “FPGA implementation of fixed point CORDIC-SVD for e-skin systems,” in *Ph.D. Research in Microelectronics and Electronics (PRIME), 11th Conference*, 2015.
- [232] R. Andraka, “A survey of CORDIC algorithms for FPGA based computers,” in *Field Programmable Gate Arrays (FPGA), ACM/SIGDA Sixth International Symposium*, 1998.
- [233] E. Deprettere, P. Dewilde, and R. Udo, “Pipelined CORDIC architectures for fast VLSI filtering and array processing,” in *Acoustics, Speech, and Signal Processing (ICASSP), IEEE Conference*, 1984.
- [234] M. Franceschi, A. Nannarelli, and M. Valle, “Tunable floating-point for embedded machine learning algorithms implementation,” in *2018 15th International*

- Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD)*, July 2018, pp. 89–92.
- [235] ———, “Tunable floating-point for artificial neural networks,” to appear in 2018 25th IEEE International Conference on Electronics, Circuits and Systems (ICECS), 2018.
  - [236] A. Agrawal, *et al.*, “Approximate Computing: Challenges and Opportunities,” in *2016 IEEE International Conference on Rebooting Computing (ICRC)*, Oct 2016, pp. 1–8.
  - [237] B. Catanzaro, “Computer Arithmetic in Deep Learning,” in *Keynote Talk at the 23rd IEEE Symposium in Computer Arithmetic*, July 2016. [Online]. Available: <http://arith23.gforge.inria.fr/slides/Catanzaro.pdf>
  - [238] *IEEE Standard for Floating-Point Arithmetic*, IEEE Computer Society Std. 754, 2008.
  - [239] E. Nurvitadhi *et al.*, “Can fpgas beat gpus in accelerating next-generation deep neural networks?” in *Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, ser. FPGA ’17, 2017, pp. 5–14.
  - [240] A. Nannarelli, “A multi-format floating-point multiplier for power-efficient operations,” in *Proceedings of the 30th IEEE International System-on-Chip Conference*. IEEE, 2017, pp. 351–356.
  - [241] ———, “Tunable Floating-Point for Energy Efficient Accelerators,” in *25th IEEE Symposium on Computer Arithmetic*, Jun. 2018, pp. 29–36.
  - [242] N. P. Jouppi *et al.*, “In-Datcenter Performance Analysis of a Tensor Processing Unit,” in *ACM/IEEE 44th Annual International Symposium on Computer Architecture (ISCA)*, 2017, pp. 1–12.
  - [243] M. Kim and P. Smaragdis, “Bitwise Neural Networks,” *arXiv preprint arXiv:1601.06071*, 2016.
  - [244] NVIDIA Inc. NVIDIA Tesla V100 GPU Architecture. [Online]. Available: <http://images.nvidia.com/content/volta-architecture/pdf/volta-architecture-whitepaper.pdf>



- [245] V. Popescu *et al.*, “Flexpoint: Predictive Numerics for Deep Learning,” in *25th IEEE Symposium on Computer Arithmetic*, Jun. 2018.
- [246] M. Ercegovac and T. Lang, *Digital Arithmetic*. Morgan Kaufmann Publishers, 2004.
- [247] W. Xiong, *et al.*, “The Microsoft 2016 Conversational Speech Recognition System,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 5255–5259.
- [248] Y. Wu, *et al.*, “Google’s Neural Machine Translation System: Bridging the Gap between Human and Machine Translation,” *arXiv preprint arXiv:1609.08144*, 2016.
- [249] A. Nannarelli. (2018) Tunable Floating-Point Multiplier. DTU Compute, Technical Report. [Online]. Available: [http://people.compute.dtu.dk/alna/projects/TFP/tfp\\_mult.html](http://people.compute.dtu.dk/alna/projects/TFP/tfp_mult.html)
- [250] M. Perovic *et al.*, “Electrical stimulation of the forearm: a method for transmitting sensory signals from the artificial hand to the brain,” *Journal of Automatic Control*, vol. 21, no. 1, pp. 13–18, 2013.
- [251] N. M. Malešević *et al.*, “A multi-pad electrode based functional electrical stimulation system for restoration of grasp,” *Journal of NeuroEngineering and Rehabilitation*, vol. 9, no. 1, p. 66, Sep 2012. [Online]. Available: <https://doi.org/10.1186/1743-0003-9-66>