Embedded Implementation of Signal Pre-processing for Tactile Sensing System

Moustafa Saleh [0000-0002-7028-9917], Yahya Abbass [0000-0002-9225-8900] and Maurizio Valle[0000-0002-7366-6060]

University of Genova, Genoa, Italy
moustafa.saleh@edu.unige.it

Abstract. Smart tactile sensing system has been a subject of research in many application domains such as prosthetics and robotics. Embedding signal pre-processing methods (i.e., filters) along with processing algorithms (i.e., machine learning) into miniaturized electronic units enhance the extraction of high-bandwidth information (e.g., slippage detection). However, it is challenging due to the high computational costs and the real time requirements. This paper proposes a lightweight implementation of pre-processing method for multichannel tactile sensing system. We targeted two filtering methods, Finite Impulse Response (FIR) and Exponential Moving Average Filter (EMAF). The paper presents the analysis of the implementation performance on hardware i.e., number of clock cycles, execution time and touch detection accuracy. Experimental results show that EMAF is more effective than FIR when it comes to the hardware complexity. This means that the computational cost for implementing such pre-processing filter is negligible and thus acceptable for time, and hardware constraint tactile sensing system.

Keywords: tactile sensing system, signal processing, filtering methods, embedded implementation.

1 Introduction and Related Work

Tactile sensing in humans is one of the fundamental sensory modalities (visual, auditory etc.) that plays an important role in conveying information to the brain about objects in contact (e.g., contact surface, roughness, shape, grasp stability and slip detection [1] etc.). Considerable scientific efforts have been devoted to enhance the motor control in robotics to carry-out human-like movements such as manipulation and exploration tasks [2]. Performing such tasks successfully requires a tactile sensing system that can extract high bandwidth tactile information.

Signal processing and feature extraction algorithms are crucial to properly decode and translate tactile sensor signals into useable tactile information. Over the past two decades, the usage of tactile sensing systems in practical applications has been limited due to some limitation in the performance of the multi-channel sensing system (i.e., sensor and signal conditioning circuits). For example, signals from massive amount of tactile sensors may be noisy and containing irrelevant information [3]. Thus, this arises the demand of developing efficient sensing systems at both the hardware level (e.g., highly sensitive sensors and acquisition circuits) and the software level (e.g., signal processing methods). As on the software level, general statistical algorithms and machine learning techniques have been applied to the problem of extracting tactile information. The Fast Fourier Transform (FFT) and Discrete Wavelet Transform (DWT) are leading techniques for preprocessing and simplifying signal output from sensors. The FFT is most used for temporal signals from sensors. DWT is often used when FFT results fail to produce satisfactory features of the touched object (e.g. surface characteristic) [4]. It is also used for feature extraction and its filters can also be applied for the processing of raw signals; DWT can be used for creating high-pass and lowpass filters that can construct the original frequency and the temporal information. However, the deployment of these two techniques for signal filtering result in a high computational load as both methods require the computation of the spectrum and then the spectrum inverse. Therefore, large memory is required, thus the sensing system is characterized with high latency. Statistical and learning algorithms including decision trees (DT), support vector machines (SVM), extreme learning machines (ELM), gradient boosting machines (GBM), maximum likelihood estimations (MLE), k-means clustering (KM), and k nearest neighbors (KNN) are used to categorize and identify many properties of the touched object [5].

Prior performing any of the aforementioned algorithms, signal pre-processing is required, such as removing signal disturbances with appropriate filters. This allows the sensing system to well extract features and interpret the measured information. However, implementing such methods (i.e., filters) along with processing algorithms (i.e., touch modalities classification algorithms) onto the embedded tactile sensing system poses some challenges on system performance in terms of computational cost, response accuracy [6] and time delay. To the extent of our knowledge, such embedded implementation of signal pre-processing and processing methods for a real tactile sensing application is still lacking. Where in most cases data are collected and stored to then be processed offline on PC hosts. Thus, assessing the performance of such processing methods on embedded sensing system hardware is strongly recommended.

This paper proposes a lightweight implementation of pre-processing filtering method for multi-channel tactile sensing system prepared to be used in real application. For this purpose, an end-to-end distributed sensing system was developed on board aiming to collect data corresponding to human-like touches. Two non-recursive digital filters have been deployed on the Interface Electronics (IE) as the acquisition block of the sensing system and applied onto the collected data: FIR and EMAF. These two filters have linear phase characteristics, which unlike other filters (e.g., Infinite Impulse Response and decimation) are more stable and allow retaining the shape of the original signal after filtering. Experimental tests have been carried out to assess the behavior of both filters running on the IE. To select the filter with better performance, analysis metrics have been measured such as Signal-to-Noise Ratio (SNR) of the output tactile signal, detection of the touch events and computational complexity of each technique. Experimental results show that with EMAF few computations are required to filter out tactile signals which unlikely when using FIR. Thus, making EMAF more suitable for a time and hardware constraints tactile sensing system.

This paper is organized as follows: section 2 describes the tactile sensing system components, discusses the proposed pre-processing methods and the implementation methodology onto the IE. Section 3 presents and analyzes the performance of the implemented filters. Finally, section 4 concludes the paper.

2 Methodology

2.1 Tactile Sensing System Architecture

Figure. 1 shows the structure of a typical tactile sensing system. The system includes 1) a sensing array, 2) an IE for data acquisition, and 3) an embedded digital processing unit for tactile data decoding. Sections below just introduce the sensor array structure, and the IE blocks that have been involved in the presented work. Where the proposed filtering methods have been implemented into the IE.

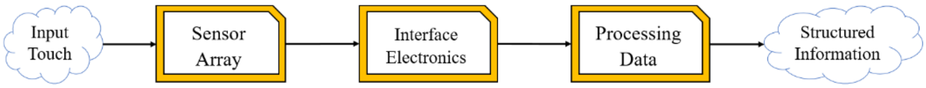

Fig. 1. Tactile sensing system block diagram

Sensor Array Structure.
Sensing arrays reported in [7] require an additional fabrication step to realize skin patches that can be integrated into different applications. The sensor array is based on P(VDF-TrFE) poly(vinylidene fluoride trifluor-oethylene) piezoelectric poly- mer sensors. Fig. 2 shows the structure of the skin patch for the palm. The sensing array used in this work was prepared and integrated following the same approach described in [8]. The arrays were shielded using conductive tapes (Model tesa 60262) and then protected

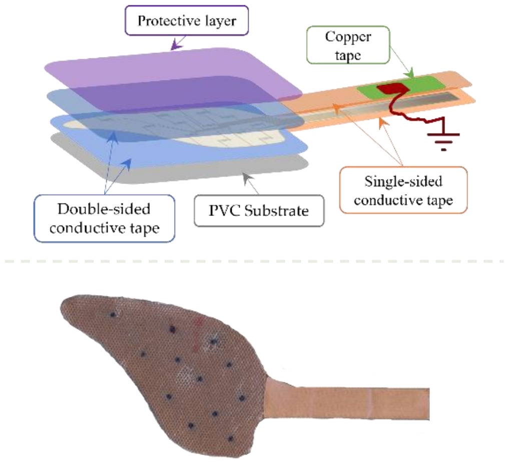

Fig. 2. Top: structure of the skin patch, Bottom: Picture of a real sample of the skin patch.

using a thin flexible protective layer (Art. 5500 Dream, Framisitalia). The structure was mounted on a flexible substrate (i.e., PVC). The shielding layers were connected to the ground reference of the embedded electronics using a self-adhesive copper foil tape and a wire.

Interface Electronics Block.
Fig. 3. depicts the Printed Circuit Board (PCB) of the IE that acts as the acquisition board of the tactile sensing system. It is based on two main off-the-shelf components: BL600 module [9] and DDC232 [10] current-input analog-to-digital converter. The DDC232 converter offers simultaneous sampling for 32 bipolar tactile sensors with configurable sampling rate of up to 6 kHz. The BL600 contains an ultralow-power microcontroller based on an ARM Cortex M0 chip, it is used for retrieving, pre-processing, and transmitting tactile data. The presented design can be powered supply through USB cable, at which is also used for transmitting sensor data throughout the system. The IE has been preliminary validated in previous work [11]. In the present study, the IE was configured to collect and process tactile data from 1 sensor at 2 KSamples/second. It is also endowed with signal pre-processing to detect contact events. Data are transmitted to the PC through a USB cable.

Fig. 3. IE printed circuit board of the tactile sensing system

2.2 Signal Pre-processing Methods Description

In this work we adopted and implemented into the IE two filters as pre-processing methods: FIR and EMAF. The implementation process is introduced and the results of the two methods are discussed in next sections.

Filtering Methods.

Finite Impulse Response.
FIR is a well-known digital signal processing filter. The filter output y(n) is a result of discrete-time convolution process between input signal x(n) and the impulse response

h(k) of the filter (filter coefficients), see equation 1. The design of the FIR filter is composed of two main parameters: cutoff frequency and filter order. The filter order generally is recommended to be high to have more accurate results. Thus, implementing high-order filters to smoothen signals from 32 channels will introduce a huge delay in the extraction of tactile information and thus in the feedback loop. Therefore, a trade-off between the accuracy of the filter and the order should be considered while designing the filter.

$$y(n) = \sum_{k=0}^{N-1} h(k)x(n-k) \tag{1}$$

Exponential Moving Average Filter.

Moving average filter (MA) is one of the popular digital filtering techniques that can smoothen all kind of data and reduce random noise in the data. EMAF is a type of MA filter that operates with low computational burden and can be implemented easily and efficiently. EMAF filter computes a weighted average of time ordered sequence by applying to the previous inputs weights that decrease exponentially [12].

The EMAF on input x[n] is expressed as in equation 2:

$$y[n] = \alpha x[n] + (1 - \alpha)\, y[n - 1] \tag{2}$$

Where x[n] is the current input, y[n] is the current output, and y[n-1] is the previous output; α is the smoothing factor ranges between 0 and 1. As α decreases, high frequencies are attenuated. The design of these filters is introduced in the next sections where experiments have been carried out to define the filters parameters/coefficients.

2.3 Experimental Setup and Filters Implementation

To define the parameters of the filters – for the purpose of implementation – we have arranged an experimental setup (see Fig. 4) to study the response of the sensing system to different touch patterns that might be incorporated while the skin patch interacts with objects. The setup is composed of a skin patch placed on the top of a strain gauge load cell (Tedea Huntleigh, Model 1042) and faced upside. The load cell was used to measure the force applied on its surface where the skin patch is placed. The force stimulus was conditioned by a PXIe-4330 (NI, US) conditioning board while the charge developed by the sensor (response) was conditioned either by the PCB Sensor Signal Conditioner (482C54) or by the IE as illustrated in Fig. 4. A LabVIEW software developed on a National Instruments PXI system was used to collect, visualize, and save the force stimulus and the charge response.

Three touch patterns were selected, Tapping Touch (TT), Press-Hold-Release Touch (PRT), and Continuous Touch (CT). During the tests, the experimenter applied one of the patterns on a single sensor using his/her finger. For example, Fig. 5 and Fig. 6 show the three patterns applied on a single sensor and the corresponding charge response. Fig. 5 shows the response of a single sensor to the TT and PRT patterns. Similarly, Fig. 6 shows the CT pattern and the corresponding sensor response.

The signal peaks are arranged in a sequence reflecting the fact that the touches were applied to the sensors sequentially. The PRT pattern was presented by the sensor by two bursts corresponding to the press and release events, while in-between the bursts there was some wiggling.

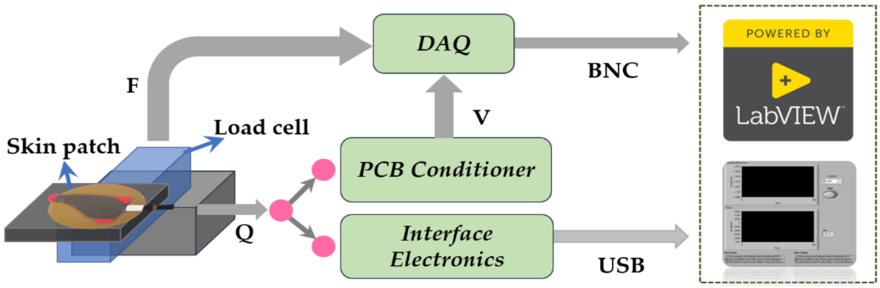

Fig. 4. Experimental Setup Block Diagram

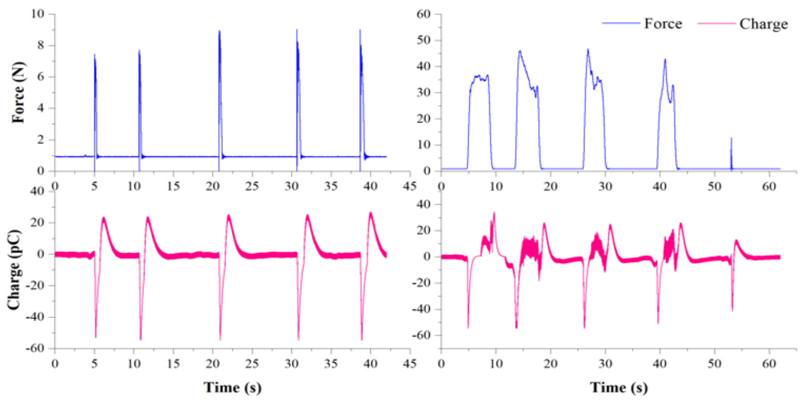

Fig. 5. Response of the single sensor to Tapping Touch (left) and Press-Hold-Release Touch patterns (right)

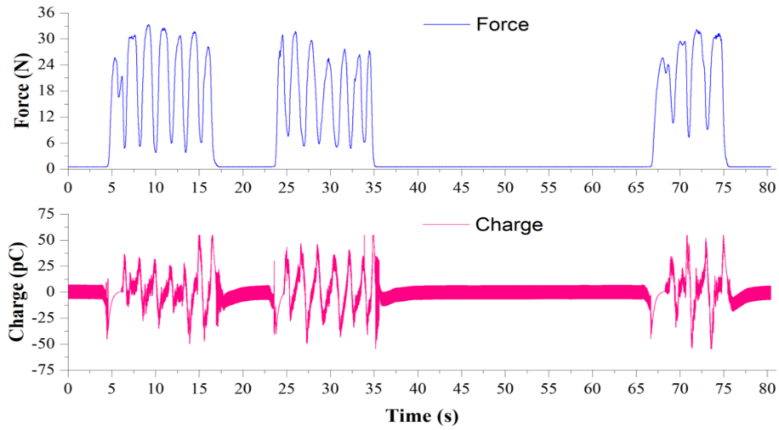

Fig. 6. Response of single sensor to a Continuous Touch pattern

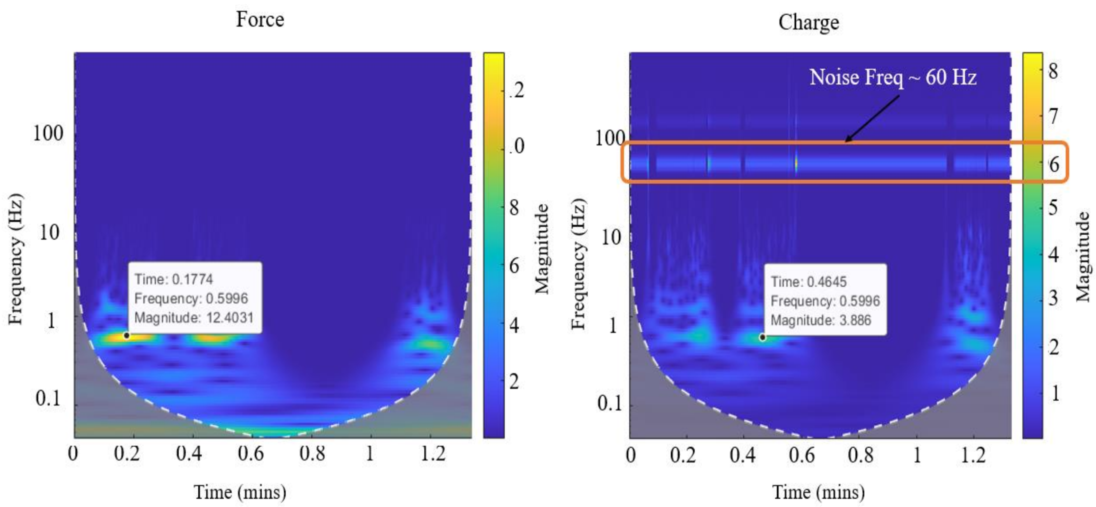

Fig. 7. Continuous wavelets transform (CWT) of the applied stimulus and the sensor response in Fig. 6

Data Collection.

To assess the behavior of the IE with the filtering methods, we used the setup shown in Fig. 4 to collect a tactile dataset that is composed of the response of a single sensor to the three-touch pattern, mentioned before. Fourteen trials have been applied, 5 for TT, 4 for PRT touch and 5 for CT pattern. During each trial, the IE acquires sensor data, applies filtering, and then sends the data to the PC. A LabVIEW GUI collects and stores the tactile data for further analysis.

Filters Coefficients.

In the experiments implemented so far, a high noise level was detected in the response of the sensors. This was more observed when plotting the time-Frequency analysis (Continuous Wavelet Transform) of the CT pattern obtained in Fig. 6 and the corresponding charge response (see Fig. 7). The plot shows frequency content of 60 Hz which is the common noise frequency. Moreover, the Time-Frequency plot indicated that the frequency range of the sensor response is below 30 Hz. Such frequency range could be extracted by applying a simple low-pass digital filter with a cutoff frequency of 30 Hz.

The filter coefficients have been computed using MATLAB and saved into the IE local memory. The FIR design was based on Hamming window to create a low pass filter type with a cut-off frequency of 30 Hz. The filter order was set to be 58, the minimum value by which the desired frequencies were filtered out with low hardware computations as possible. As for the EMAF, the α coefficient has been computed by measuring the angular cut-off frequency w_c (equation 3) at the half power point of the filter frequency response. Therefore, the corresponding α value for setting 30 Hz as cut-off frequency, was founded to be 0.0909.

$$w_c = \arccos\frac{\alpha^2+2\alpha-2}{2\alpha-2} \tag{3}$$

3 Performance Analysis

This section provides an experimental evaluation of the sensing system performance. The metrics of interest for our performance analysis are touch detection accuracy, number of reconstructed samples over all trials and filtering execution time.

3.1 Touch Accuracy and Reconstructed Samples.

Table 1 reports a comparison between FIR and EMAF in terms of accuracy, number of reconstructed samples and signal-to-noise. The accuracy was computed as the error in detecting the applied touches i.e., difference between the number of actual touches and the one detected by the sensing system. The actual touches have been defined based on the force stimulus detected by the load cell as in Fig. 2 and Fig. 3. Each of the three touch patterns has its corresponding number of applied actual touches i.e., 3 touches for PRT, 10 touches for each of TT and CT. In each trail, a Detection Threshold (DT) was set by enabling the IE to record signals from the sensor for at least 3 secs with no mechanical interaction. Increasing the no touch duration provides more accurate DT. During these three seconds the DT was set by finding the highest (δmax) and the lowest amplitude (δmin) to detect press and release events, respectively (Fig. 8). Both press and release events are considered as detected touches. Table 1 shows that an accuracy of 93.89% was achieved after applying filtering with either FIR or EMAF technique. Furthermore, it shows the impact of filtering on the threshold levels which allows reconstructing a greater number of samples below and above the new thresholds. Results show 12x and 14x increase in the number of reconstructed samples with EMAF and FIR, respectively.

Table 1. Average Values of Touch Detection Accuracy, Number of Reconstructed Samples and Signal-to-Noise Ratio (SNR) for all 14 trials collected from the IE

Processing Method	No Filter	EMAF	FIR
Touch Detection Accuracy (%)	49.61	93.89	93.89
Number of Reconstructed Samples	709	8322	10130
SNR (dB)	-5.007	-2.239	-1.944

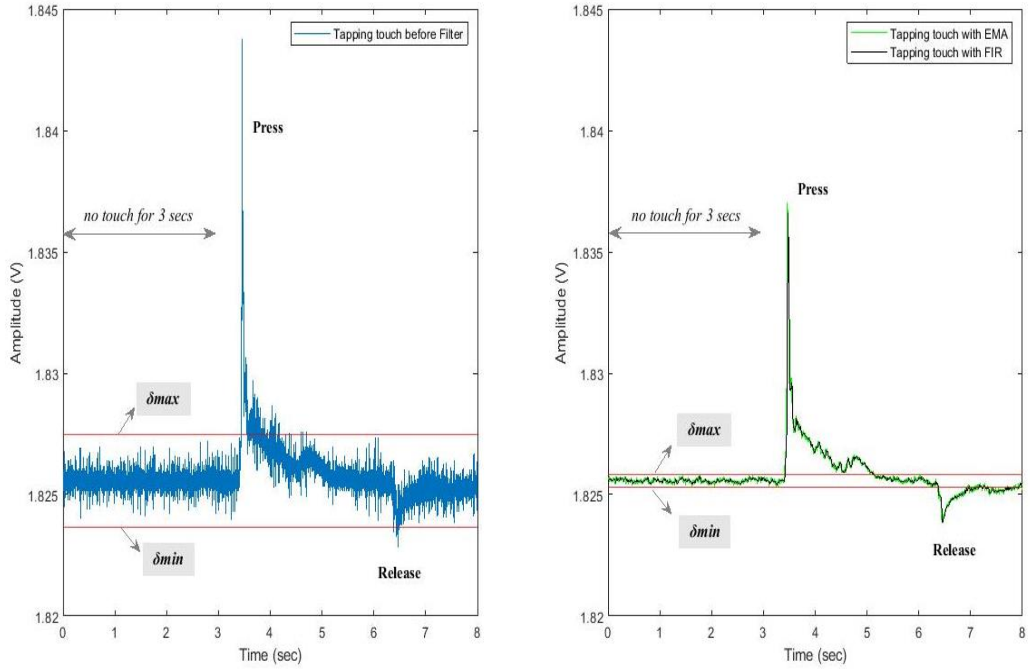

Fig. 8. Max and Min thresholds are set for all collected signals in all trials to detect press and release events; signal before filtering with thresholds (left); signal after filtering (right) one with FIR (black) and one with EMAF (green)

3.2 Filter Execution Time.

Referring to Table 1, FIR filter proved better power and sample reconstruction performance among the EMAF. This would lead, as a conclusion, to selecting FIR filter for implementation on the IE. However, the complexity of FIR is more considerable compared to EMAF, and this is obviously presented in Table 2 by measuring the execution time (ET) and number of clock cycles (CC) preformed for executing both EMAF and FIR filters on the IE. The µVision Debugger supported by ARM Keil framework has been used for debugging, loading, and executing the application program onto the IE. When program execution stops, µVision shows the current state of the CPU registers. Two registers have been monitored to observe the approximated values of ET and CC. As shown in Table 2 the EMAF is 190x faster than the FIR, requiring just 47 clock cycles to be executed. These values correspond to single input sensor; however, the system carries at least 32 input sensors that are sampled simultaneously every 500us. So, with EMAF just 96 us (32 x 3 us) takes for filtering all the 32 sensors, whereas the FIR execution time is 570 us exceeding the sampling time (500 us) in the IE. Thus, resulting in a loss in some tactile data which consequently reduces the sampling frequency of the system (2 KSample/second).

Table 2. Filtering Execution Time and Clock Cycles performed to filter one input signal running onto the IE

Processing Method	EMAF	FIR
Filtering Execution Time (us)	3	570
Clock Cycles	47	9120

Therefore, using EMAF will lead to a faster system in acquiring, processing, and transmitting data. In addition to the significant improvements in the quantity of touch samples, which is useful for identifying several levels of touch force.

4 Conclusion

This work presents the implementation of a lightweight pre-processing filtering methods for tactile sensing system. Two filtering methods were considered, FIR and EMAF filters. Both filters were implemented into the IE acquisition board of a complete tactile sensing system. Experimental results showed the effect of filtering methods on the system performance and provides a light-weight pre-processing method based on the EMAF filter compared to the FIR. The sensing system, with EMAF, achieves high touch detection accuracy compared to system running without filtering. In addition to the significant improvement in the number of reconstructed tactile samples, paving the way toward identifying high resolution touch force levels. The future work involves the implementation of more advanced signal processing methods (i.e., machine learning) with the aim of extracting features locally on hardware.

References

1. R. S. Johansson and J. R. Flanagan, "Coding and use of tactile signals from the fingertips in object manipulation tasks," *Nat. Rev. Neurosci.*, vol. 10, no. 5, pp. 345–359, May 2009, doi: 10.1038/nrn2621.
2. I. Saunders and S. Vijayakumar, "The role of feed-forward and feedback processes for closed-loop prosthesis control," *J. Neuroeng. Rehabil.*, vol. 8, no. 1, p. 60, 2011, doi: 10.1186/1743-0003-8-60.
3. O. Kroemer, C. H. Lampert, and J. Peters, "Learning Dynamic Tactile Sensing With Robust Vision-Based Training," vol. 27, no. 3, pp. 545–557, 2011, [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5752870&casa_token=T6-SrVedqKgAAAAA:SyWga8VVMbeHb2jlg9btAisVjb6yYbK0_M29RGGB0Zu5gVSMDTqPzvYauEpFCczgdzamv6Wb&tag=1.
4. S. A. A. Karim, M. H. Kamarudin, B. A. Karim, M. K. Hasan, and J. Sulaiman, "Wavelet Transform and Fast Fourier Transform for signal compression: A comparative study," *Int. Conf. Electron. Devices, Syst. Appl.*, pp. 280–285, 2011, doi: 10.1109/ICEDSA.2011.5959031.
5. A. Masteller, S. Sankar, H. B. Kim, K. Ding, X. Liu, and A. H. All, "Recent Developments in Prosthesis Sensors, Texture Recognition, and Sensory Stimulation for Upper Limb

Prostheses," *Ann. Biomed. Eng.*, vol. 49, no. 1, pp. 57–74, 2021, doi: 10.1007/s10439-020-02678-8.

6. C. Gianoglio, E. Ragusa, P. Gastaldo, and M. Valle, "A Novel Learning Strategy for the Trade-Off between Accuracy and Computational Cost: A Touch Modalities Classification Case Study," *IEEE Sens. J.*, vol. 22, no. 1, pp. 659–670, 2022, doi: 10.1109/JSEN.2021.3129323.
7. H. Fares, Y. Abbass, M. Valle, and L. Seminara, "Validation of Screen-Printed Electronic Skin Based on Piezoelectric Polymer Sensors," *Sensors*, vol. 20, no. 4, p. 1160, Feb. 2020, doi: 10.3390/s20041160.
8. Y. Abbass, M. Saleh, S. Dosen, and M. Valle, "Embedded Electrotactile Feedback System for Hand Prostheses Using Matrix Electrode and Electronic Skin," *IEEE Trans. Biomed. Circuits Syst.*, vol. 15, no. 5, pp. 912–925, 2021, doi: 10.1109/TBCAS.2021.3107723.
9. S. Mode, B. Low, E. Ble, and P. Bl-sa, "BL600 - Datasheet."
10. ["DDC232, 32-Channel, Current-Input Analog-to-Digital Converter," 2004. Accessed: May 02, 2019. [Online]. Available: www.ti.com.
11. M. Saleh, Y. Abbass, A. Ibrahim, and M. Valle, "Experimental Assessment of the Interface Electronic System for PVDF-Based Piezoelectric Tactile Sensors," *Sensors*, vol. 19, no. 20, p. 4437, Oct. 2019, doi: 10.3390/s19204437.
12. S. W. Smith, "Digital Signal Processing, Chapter 15," *Calif. Tech. Publ.*, pp. 277–284, 1999.