Hand Gestures Recognition for Human-Machine Interfaces: A Low-Power Bio-Inspired Armband

Andrea Mongardi, *Member, IEEE*, Fabio Rossi, *Member, IEEE*,
Andrea Prestia, *Graduate Student Member, IEEE*, Paolo Motto Ros, *Member, IEEE*,
Massimo Ruo Roch, *Member, IEEE*, Maurizio Martina, *Senior Member, IEEE*,
and Danilo Demarchi, *Senior Member, IEEE*

***Abstract*—Hand gesture recognition has recently increased its popularity as Human-Machine Interface (HMI) in the biomedical field. Indeed, it can be performed involving many different non-invasive techniques, e.g., surface ElectroMyoGraphy (sEMG) or PhotoPlethysmoGraphy (PPG). In the last few years, the interest demonstrated by both academia and industry brought to a continuous spawning of commercial and custom wearable devices, which tried to address different challenges in many application fields, from tele-rehabilitation to sign language recognition. In this work, we propose a novel 7-channel sEMG armband, which can be employed as HMI for both serious gaming control and rehabilitation support. In particular, we designed the prototype focusing on the capability of our device to compute the Average Threshold Crossing (ATC) parameter, which is evaluated by counting how many times the sEMG signal crosses a threshold during a fixed time duration (i.e., 130 ms), directly on the wearable device. Exploiting the event-driven characteristic of the ATC, our armband is able to accomplish the on-board prediction of common hand gestures requiring less power w.r.t. state of the art devices. At the end of an acquisition campaign that involved the participation of 26 people, we obtained an average classifier accuracy of 91.9% when aiming to recognize in real time 8 active hand gestures plus the idle state. Furthermore, with 2.92 mA of current absorption during active functioning and 1.34 ms prediction latency, this prototype confirmed our expectations and can be an appealing solution for long-term (up to 60 h) medical and consumer applications.**

***Index Terms*—Artificial Neural Networks, Edge Computing, Event-based, Embedded Machine Learning, Real Time Gesture Recognition, Surface Electromyography, Wearable Devices.**

I. Introduction

DURING the last decade, hand gesture recognition has been increasingly investigated by both academia and industry, following the versatility and easiness of its implementation. Indeed, thanks to technology progresses in the electronics fields, the recognition of hand movements can be now performed directly with body sensors [1], without the needing of a complex equipment, making it one of the best candidates for Human-Machine Interface (HMI) systems.

The main market response to this phenomenon is surely the realization of the Myo armband by Thalmic Labs, which is composed by 8 channels for the acquisition of surface ElectroMyoGraphic (sEMG) signals. Thanks to its capability to adapt to many different scenarios [2], this armband received a huge positive feedback from the community. Indeed, even if its production was halted some years ago, the literature is still packed with several contributions which involve the Myo armband as the main acquisition device [3], [4], [5], [6], exploiting its small, compact design which fits into many different practical applications. On the other hand, there are still many studies where the circumstances demand the design of a custom armband, e.g., to address specific constraints like low power consumption [7] or low noise contribution on the signal [8], or to obtain solutions with augmented functionalities and an increased number of recognized gestures [9], [10], [11], [12], [13].

Therefore, the large availability of commercial devices and the knowledge acquired by multiple research groups around the world made hand gesture recognition be involved in many HMI use-cases. Among the most active, we can find computer interaction [14], outdoor music handling [15], biomedical applications like prostheses [16], [17], [18] and robotic arm control [6], [19], [20], or even sign language communication [21], [22], which requires a lot of sensitivity in recognizing each single finger movements. Furthermore, by exploiting the other actor of the HMI (i.e., the machine), hand gesture recognition can be involved also in the (tele-)rehabilitation field, both with passive and active approaches. Indeed, standard rehabilitative sessions can motivate the patients making them interact with the so-called serious games [23], [24], [25] or by giving them more simple feedback during movements execution [26], [27], [28]. On the other hand, for people which are not able to autonomously perform the rehabilitation routine, the therapist can, for example, apply the functional electrical stimulation on the injured muscle(s), both involving data compression [29] and maintaining the association of the gestures one by one [30], [31].

Regarding the features involved in the recognition, many studies take the raw sEMG signal as input, sampling wide time

Manuscript received 8 March 2022; revised 4 May 2022, 3 July 2022, and 31 August 2022; accepted 16 September 2022. Date of publication 3 October 2022; date of current version 14 February 2023. This paper was recommended by Associate Editor M. Atef. *(Corresponding author: Andrea Mongardi.)*

The authors are with the Department of Electronics and Telecommunications, Politecnico di Torino, 10129 Turin, Italy (e-mail: andrea.mongardi@polito.it; fabio.rossi@polito.it; andrea.prestia@polito.it; paolo.mottoros@polito.it; massimo.ruoroch@polito.it; maurizio.martina@polito.it; danilo.demarchi@polito.it).

This work involved human subjects or animals in its research. Approval of all ethical and experimental procedures and protocols was granted by Comitato Bioetico di Ateneo of the University of Turin under Application No. 445136.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TBCAS.2022.3211424.

Digital Object Identifier 10.1109/TBCAS.2022.3211424

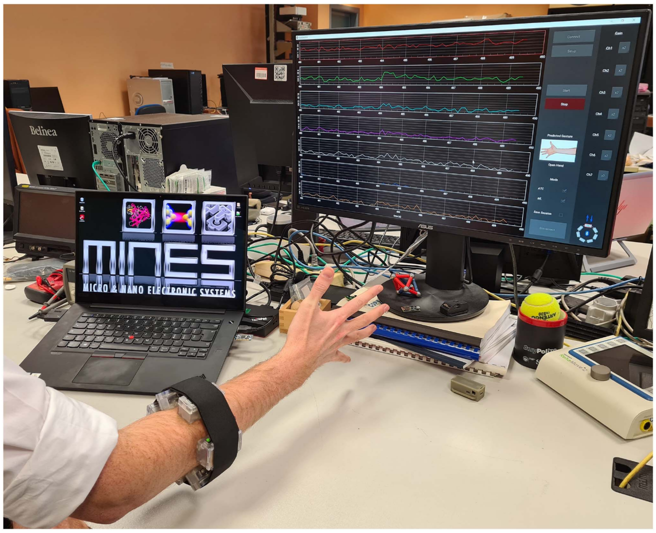

Fig. 1. Example of the armband application: the device streams the ATC values and the related recognized gesture, which are represented on the graphical user interface.

windows (e.g., longer than 250 ms) on the available channels and, in most cases, feeding a deep learning algorithm like the Convolutional Neural Network (CNN), which can be configured with multiple internal structures [16], [32], [33] emulating the functions of the human brain. Furthermore, few studies tried to exploit the real brain behavior as classifier for hand gesture recognition, both designing liquid spiking network structures [34] and trying to directly classify the brain electrical signals [27]. However, considering that we are in the era of Internet of Things (IoT) and embedded devices, some studies [28], [35], [36], [37] focus on the implementation and on the optimization of less complex Machine Learning (ML) algorithms (e.g., Artificial Neural Networks (ANNs), Support Vector Machines (SVM) and Decision Trees (DTs)) to fulfill the requirements of most common microcontrollers, which typically have less than 1 MB of memory space and cannot perform computations at high speed.

In this work, we extended the application domain of our last live demonstration [38], which involved two standalone channels only, by leveraging our previous knowledge of hand gesture recognition [39], [40], [41]. In particular, we designed a custom prototype of a modular armband, in Fig. 1 captured during real time operations, in order to improve the spatial resolution of our acquisition system, thus being able to recognize more gestures than the previously achieved three [38], while maintaining the power consumption as low as possible.

Contrary to our previous applications [39], [40], [41], which were all based on an old hardware version of our acquisition device [42], the armband we designed in this work is equipped with seven identical custom Printed Circuit Boards (PCBs), which are an upgraded version of the board described in [43] and already integrated in our last live demonstration [38]. These acquisition devices feature a digital part and an Analog Front-End (AFE) circuit for sEMG acquisition each, both designed with off-the-shelf components only. The AFE acquisition chain mainly consists of a differential high-pass filter, an instrumentation amplifier, a programmable gain stage, and a low-pass filter, to obtain a standard-conditioned sEMG signal. After that, a hardware voltage comparator is configured to extract the Threshold Crossing (TC) signal. This bio-inspired signal is driven to the on-board microcontroller which simply counts the incoming TC events during predefined (i.e., 130 ms) time windows, thus obtaining the Average Threshold Crossing (ATC) parameter [44], which has been demonstrated to be positively correlated with the exerted muscular force [45].

With the ATC parameter as our unique feature, proceeding with the analysis started in the live demonstration [38], we chose to maintain the ANN as the recognition algorithm because it adequately fits the hardware resources (i.e., 1 MB of Flash memory, 384 kB of RAM, and a dedicated floating-point unit) provided by the on-board Ambiq Apollo3 Blue Micro Controller Unit (MCU) [46]. Furthermore, thanks to the minimal computational overhead introduced by the ANN forward propagation, the energy required by each gesture prediction was reduced as much as possible.

To summarize, taking advantage of our previous research on the bio-inspired field [38], [40], [43], in this work we introduced the following novel contributions:

- a modular armband structure, inclusive of specific 3D-printed cases for each board and an elastic band to make it adjustable to different forearm sizes;
- a custom communication protocol, developed on top of the I^2C serial bus, to manage inter-boards communications;
- two additional operating modes (w.r.t. the simpler ATC streaming previously available), allowing the user to request the prediction class alone or concurrently to the ATC values transmission;
- a dedicated external software package, composed of an Application Programming Interface (API) and a Graphical User Interface (GUI), to decode the commands of the user and transmit them to the armband, as well as letting her/him supervise its status;
- an embedded ANN architecture, constrained to 2 hidden layers of 50 nodes each in order to fit the memory size of the MCU;
- acquisition protocols for both the training and testing campaign, aimed at obtaining data with equally represented gestures while ensuring the necessary rest for the subjects.

II. Methodology

This section guides the reader to a detailed description of all the design, application and validation phases of the proposed armband. The first subsection II-A, II-B, II-C, II-D, and II-E report the key-aspects for the conceptual development of the armband, from its logical architecture to the definition of its internal and external data communication. The 3D-printed realization of the wearable case is shown in subsection II-F, while subsection II-G introduces the control software, also equipped with a graphical user interface, to easily use the armband from an external computer. The remaining subsections explain the

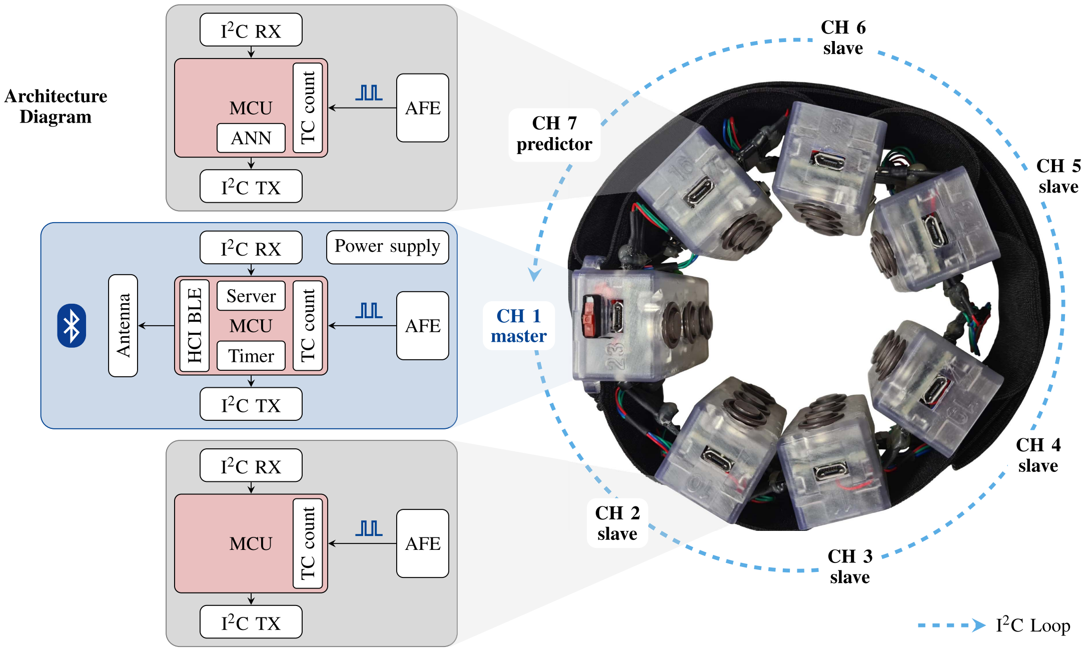

Fig. 2. The architecture diagram (left) shows the technical differences between the modules: the master (blue) has the major workload, while the slaves (gray) only need to count the TC events, with the exception of CH 7, i.e., the predictor, also responsible of ANN predictions. In the physical prototype (right), with the elastic band keeping the cases together, power and I^{2}C wires electrically connect the boards.

procedures we followed to collect the hand gesture dataset for the ANN training (II-H) and the validation setup for the online testing phase (II-J).

A. Armband Concept

The armband has been designed as a circular ring, to fit around the forearm, composed by seven units: one working as the master board, which provides the wireless interface with the user and controls all the operations, and the remaining six acting as its slaves, which respond to the master's commands. Data communication among the armband's boards is achieved by arranging the units in an I^{2}C closed-loop daisy-chain configuration, whose API implements a custom protocol.

As already anticipated in Section I, all the modules are based on our custom sEMG acquisition channel (detailed in [43]). Being equipped with an adaptable AFE for the TC signal, a low-power MCU for embedded ATC computations, and a Bluetooth Low Energy (BLE) [47] transceiver for wireless connectivity, the acquisition channel represents our wearable edge-computing node. Evolving from [43], this second version of our prototype features more selectable AFE gains w.r.t. its predecessor, as the unique difference relevant to this project. In particular, while the first version included a programmable gain spanning from 500 V/V to 2500 V/V, with 500 V/V steps, this new prototype provides to the user an extended gain range from 250 V/V to 4000 V/V, with lower steps of 250 V/V, thus obtaining higher flexibility and resolution.

In [43] and [38] each board worked as a standalone module, streaming the ATC data independently. In this work, we maintained the same exact PCB design, fabrication and assembly to ensure the interchangeability of each board (e.g., in case of hardware faults) and to allow us to distribute the computational effort among the boards, if needed. Thus, we only reconfigured the firmware of the devices to operate their functional role in the armband setup. In particular, the master board is responsible for the wireless (i.e., BLE) communications with the external host and for the wired (i.e., I^{2}C) transmission of commands to the slave boards, which only have to wait for the master triggers and to count TC events when requested. On the other hand, regarding gesture recognition, we preferred to move the ML computation to a separate board from the master to not overload it, considering both the computational effort during the classification task and the memory requirements needed to store the ANN parameters. Thus, we introduced the slave (sub-)role of the predictor.

Hence, merging the above roles, we defined the following organized and functional structure for the armband, as represented in Fig. 2:

- Channel 1: *master* board – it represents the operative center of the device, which translates the user actions into executable tasks; as the first channel, it is employed to arrange the armband in the right location of the forearm;
- Channels 2 - 6: *slave* board – it silently waits a command from the master to perform the related function;
- Channel 7: *predictor* board – as a slave, it responds only to a master interaction; additionally, it is in charge of gesture recognition.

A single rechargeable battery has been supposed to power up all the boards. Therefore, a total of seven bridging connectors, each one composed by four wires (i.e., power (V_{DD}), GND, I^{2}C data, and I^{2}C clock), have been provided for the interconnection between the boards.

In the last prototyping phase, we designed a 3D-printed case for each board, similar to the one of [43], to ensure the robustness and the modularity of the armband. Then, to physically define the armband itself, we bound the boards together with an elastic band, which also tightens the armband in contact with the skin and allows it to suit the anatomical dimensions of different limbs.

B. Firmware Implementation

The MCUs of each board were configured according to their role, through custom firmware modules, with the main difference lying between the master version and the slave ones, as represented in the architecture diagram in Fig. 2.

The firmware needed by the master board was based on the one developed in [43] and was extended by adding the I^{2}C message module, which was made necessary to handle communications with the slave boards along the armband loop. At the same time, we reconfigured the BLE server structure to handle information and commands to and from multiple boards, and to accept bigger payloads. The timer in charge of the ATC window duration runs unmodified; however, the end of the timer cycle was made transmittable to the other boards to synchronize the acquisition window among them. Differently, the counting of TC events was moved from the GPIO interrupts routines to a dedicated timer unit operating in counter mode, thus achieving a fully-hardware event detection without the need to wake up the Central Processing Unit (CPU) each time a TC edge reaches the MCU. The concurrent execution of these different tasks made the use of the FreeRTOS operating system [48], as the kernel of the master board, still necessary. Although this real-time operating system would increase the firmware complexity of this single device, its involvement allows the BLE module to send and receive data while the CPU is performing other computations.

The slave boards, instead, do not need an embedded OS because of their reduced functionalities. Indeed, their BLE transceivers have not been enabled, and the only modules which have been activated are the counter module for the ATC evaluation and the I^{2}C interface to handle master communications. The reduced number of active functionalities allows the CPU to stay in a deep sleep state for most of the time (with the I^{2}C triggered-actions as the only exceptions during run-time), thus saving the battery life. On the other hand, the predictor firmware is enhanced with dedicated custom functions to reproduce the ANN structure and to perform online predictions, thus consuming more energy during the active state. However, we implemented these software routines relying on the CMSIS package [49], which perfectly exploits the hardware capabilities of the ARM 32-bit Cortex-M4F processor. Indeed, being aware of the low-level mechanisms of the MCU, the CMSIS DSP sub-package is able to optimize the hardware execution of the floating-point matrix multiplications needed for ANN forward propagation, thus consuming a smaller quantity of energy and taking less time to execute w.r.t. unvectorized floating-point operations.

Regarding code memory requirements, the firmware we developed does not occupy the entire 1 MB of Flash memory available [46]. Indeed, even the master code, which has the most complicated structure having to handle so many different modules, can fit into the minimum settable Flash size (i.e., 500 kB). The data memory (i.e., RAM) was configured at its minimum size available too, i.e., 96 kB, to power down the unnecessary memory banks, thus minimizing leakage currents. In fact, the data space required by master and slave is around 50 kB and 20 kB, respectively. Only the RAM size of the predictor would be configured separately, after the hyper-parameters training, because the needed space strictly depends on the number of hidden nodes selected for the ANN.

TABLE I
BLE SERVER FOR THE ARMBAND APPLICATION

Characteristic	Handle	Size (bytes)	Permission[1]
Service Declaration	0x00A0	-	-
Command	0x00A2	1-3	W
Status	0x00A4	1	R/N
Status$_{\text{CCCD}}$[2]	0x00A5	2	R/W
ID	0x00A7	7	R
Gain	0x00A9	7	R/W
ATC	0x00AB	7-8	N
ATC$_{\text{CCCD}}$	0x00AC	2	R/W
Gesture	0x00AE	1-2	N
Gesture$_{\text{CCCD}}$	0x00AF	2	R/W

[1] R: read; W: write; N: notify.

[2] Client Characteristic Configuration Descriptor

C. BLE Server

As briefly introduced in Section II-A, data exchange and user controls are based on the BLE server built on the master board, which acts as the peripheral node in a client/server communication with a central device (e.g., a laptop, tablet, or smartphone). Starting from the private service introduced in [43], considering the seven channels configuration of the armband, which involves the stream of multiple data for each acquisition period, we extended the structure of the BLE server (see Table I), redefining the available characteristics as follows:

- *Command:* when written, it triggers the execution of a specific task (e.g., ATC threshold calibration);
- *Status:* it represents the current operating state of the armband; its value can be read or directly notified to the user every time it is updated on the board;
- *ID:* when a read request is issued by the host, the identification number of each board is returned;
- *Gain:* it can be read, to obtain the gain multiplier (i.e., from 1 to 16) of each board, or written, in order to set a specific gain for one or more AFE(s);
- *ATC:* the seven values contained in this characteristic represent the latest computed ATC data, which are updated every ATC window (i.e., 130 ms) if the related notifications are enabled;
- *Gesture:* when requested, the value of the latest predicted class (i.e., hand gesture) is notified.

As it can be seen from Table I, both *ATC* and *Gesture* characteristics present a variable size. If only one of their notifications is requested, the dimension of the characteristics matches the

TABLE II
I^2C API FOR ARMBAND LOOP FUNCTIONALITIES

Action	Description	Timeout	I^2C packet			
			`#Byte`	`Target`	`Command`	B_1–B_n
req_ID	it requests the IDs of all the boards	5 s	`0x01-0x07`	ID/`0xFF`	`0x01`	[Boards IDs]
req_gain	it logs the current AFE gain of each board	5 s	`0x01-0x07`	ID/`0xFF`	`0x02`	[AFE gains]
set_gain	it sets the AFE gain to the multiplier value (i.e., ×1,...,×16) written on B1	5 s	`0x02-0x08`	ID/`0xFF`	`0x03`	[value, result(s)[1]]
auto_thr	it launches the automatic threshold searching	30 s	`0x01-0x07`	ID/`0xFF`	`0x04`	[result(s)[1]]
start_ATC	it starts the timer counting for the ATC window on master device and enables the TC edge detection for all the slaves boards	5 s	`0x01`	ID/`0xFF`	`0x05`	[ ]
ATC_win	it records the ATC values when the ATC window expires	5 s	`0x01-0x08`	ID/`0xFF`	`0x06`	[ATC value(s)]
stop_ATC	it stops the ATC timer on master device and disables the TC events counter on slave boards	5 s	`0x01`	ID/`0xFF`	`0x07`	[ ]
predict	it is the only command which is always sent from the predictor board to the master after the completion of the prediction operation	-	`0x02`	ID/`0xFF`	`0x08`	[pred. class]

[1] boolean value

data representation of the values they are carrying (i.e., 7 B for 7 channels ATC values, and 1 B to store the gesture class). Alternatively, if both of them stream concurrently, we resize the above dimensions by adding one more byte, standing as the packet number, in order to correctly pair the two BLE packets in the receiving software routine.

D. Armband Communication Protocol

Fulfilling its role of master node of the armband, the main board also handles the I^2C loop protocol. We defined a custom structure for the communication packet which shows the following data fields (see Fig. 3):

- `I²C ADDR`: it represents the address of the receiving board, generally set to `0x41` because of the simple peer-to-peer communication among each couple of boards;
- `#Byte`: it identifies the total number of bytes included into the payload of the I^2C packet;
- `Target`: this value points the message to the right receiver; it could be a board ID (if the command has to be performed only by one board) or `0xFF`, acting as broadcast indicator (if the operation has to be executed by each board on the armband);
- `Command`: it gives the information about which action needs to be executed;
- B_1–B_n: they composed the variable length payload.

On the master side, every time an armband action is requested, the corresponding I^2C packet is built and sent through the loop. Then, the master waits for the packet to come back and, once received, checks the correctness of its payload. On the slaves side, any time an I^2C message is correctly received (check on the `I²C ADDR` field), the `#Byte` and `Target` blocks are processed to verify the payload entirety and the addressee board. If the target does not correspond to the ID of the receiving board, the I^2C packet is put again in the loop unmodified; otherwise, if the target coincides, the requested task is executed, its result (when expected) is appended to the packet's payload, and the I^2C message is sent to the next board.

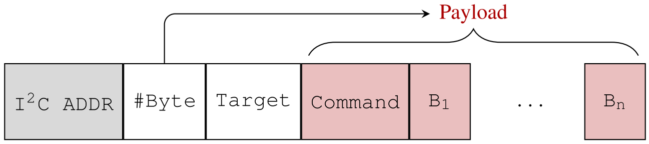

Fig. 3. I^2C packet structure.

Since the I^2C loop functionality strictly depends on the proper functioning of each board, a timeout (whose value depends on the expected processing time of each task) has been implemented on the master side as a control mechanism. If no message is received by the master within the timeout, a *hardware* error is arisen.

Additional operation failures are supervised by checking the content of the payload: it can be achieved by assessing the task result (e.g., *auto_thr*) or by verifying the appropriate data range of values (e.g., *req_gain*). Consequently, errors are handled by updating the server *Status* characteristic and sending the notification to the user.

An inclusive list of all the implemented commands, their description, timeout definition, and I^2C packet structure are reported in Table II.

E. Functional Operations

Apart from the configuration commands, two main armband functionalities are defined, i.e., ATC mode and ML mode, designed for data acquisition and class prediction, respectively. Their implementation was performed by interfacing the properties of the BLE server, the defined I^2C commands, and the firmware routines corresponding to the different board roles.

In ATC mode, after enabling the ATC notifications, the master board starts the ATC timer and sends the *start_ATC* command to activate the TC signal edge-detection for all the slave boards. Then, every time the capture and compare unit of the timer raises an interrupt for the end of an ATC window, the master loops the

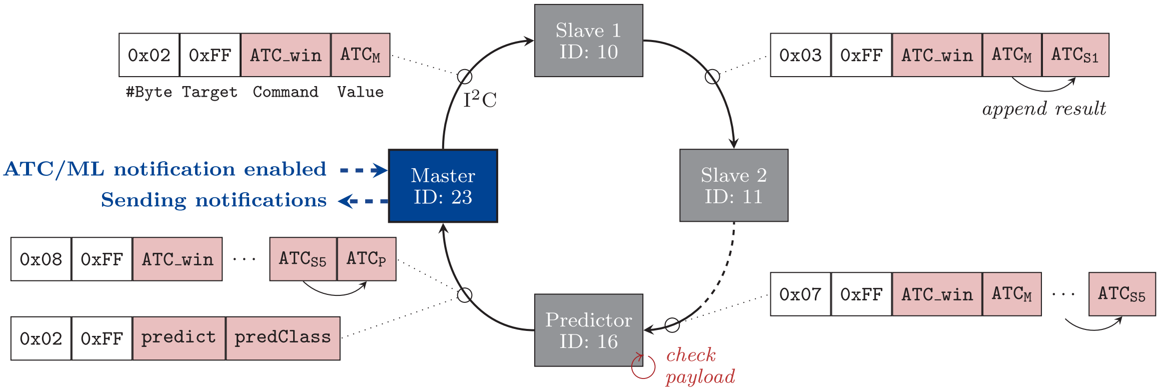

Fig. 4. I^{2}C protocol communication example during ATC and ML modes working concurrently: the master board sends the *ATC_win* command (attaching its TC count as first payload's value) to request for the ATC values of all the boards; the predictor, after checking the payload's dimension, firstly attaches its TC count to return the ATC I^{2}C packet to the master, then predicts the gesture and closes the loop sending the recognition result.

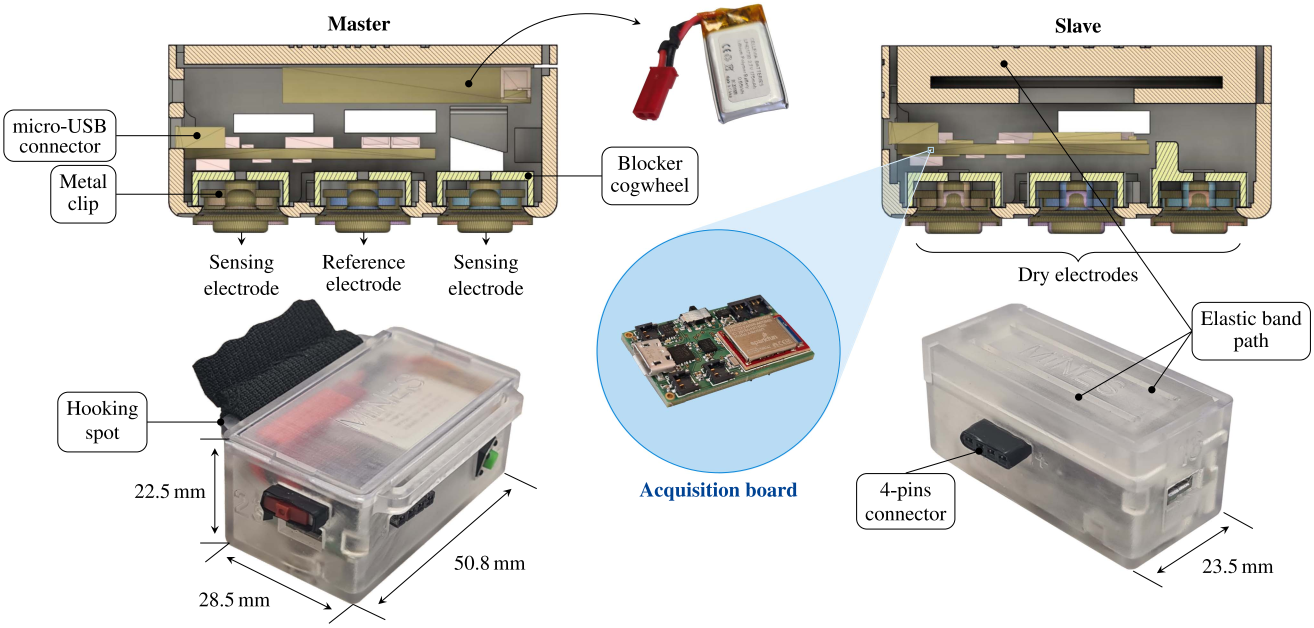

Fig. 5. Detailed views of the 3D-printed cases, and their internal organization, for both the master and slave units.

ATC_win command to collect the TC count from all the other boards and, after appending its value last, it updates the server's characteristic and sends the notification. The *ATC_win* cycle continues until the notifications are disabled, stopping the timer and commanding the *stop_ATC*, which consequently inactivates the counter of the TC edges on all the boards. Considering the above configuration, only the master board runs the timer for the ATC window while all the slaves wait for the *ATC_win* command to end their observation window and to attach their TC count.

Differently, during the ML mode (gesture notifications enabled), the master slightly changes the above behavior during the ATC collection by attaching its own value as first byte of the payload while looping the *ATC_win* command. In this way, the I^{2}C packet arrives to the predictor board with the payload containing the ATC values of all the other modules which, integrated with its one, represent the entire input to perform the gesture prediction. Once the ML operations are completed, the predictor puts the result into the prediction packet and sends it to the master.

Therefore, as depicted in Fig. 4 for the ML case, the predictor board passively understands which operation it needs to perform by checking the length of the I^{2}C packet's payload (when the command is *ATC_win*), i.e, 5 B in ATC mode and 6 B in ML mode.

Moreover, both modalities can work together, with the master sending both the ATC and the gesture notifications to the BLE host, as the unique functional difference. Indeed, the predictor board would identify the message as if in ML mode and would correctly perform the prediction, and the master would be the only board to be aware of the requested concurrency.

F. 3D Modeling

Besides the electronic aspects discussed until here, the physical realization of the armband still needs to be presented in order to complete the wearable prototype. As reported in Fig. 5, we designed two distinct cases for the master and slave boards by using a commercial 3D printer, which employs a photopolymeric, bio-compatible and lightweight resin to extrude the model.

Both the printed cases feature the openings to dispose the dry electrodes, arranged in lines following the sensing-reference-sensing configuration, on their bottom surface, just barely below the acquisition unit to minimize electrodes-to-circuit interference. The electrodes are kept in a secure position thanks to internal metal clips, which grab the tips of the electrodes, and by some blocker cogwheels, that limit their vertical displacements. At the right and left sides, we opened two rectangular slots for the insertion of the four pins connectors to attach the bridging (among the boards) cables carrying the I^2C and power supply lines. The micro-USB connector remains accessible from the front wall, thus giving to the user the possibility to easily debug and flash the modules. As last common trait, both cases provide a push-button slot on one side, to manually achieve the MCU reset when needed.

Additionally, the internal space of the master package was arranged to contain also a small rechargeable battery (i.e., a 175 mAh LiPo battery [50]), and two switches, which allow the user to modify the power configuration of the device. In particular, one switch is placed on the positive pole of the battery, to completely turn off the armband when needed, and the other one is inserted onto the V_{DD} branch of the slave boards, to cut them off during battery charging.

At last, the boxes are closed by means of sliding covers, which also provide the hooking spots for the elastic band to shrink the armband around the forearm. Indeed, the strip is sewed to one of the two wings of the master cover, while, at the other extremity, it is pulled until the armband is properly positioned. Differently, the slave case features an eyelet U-path inside its cover, which, thanks to the internal squared corners, avoids the band to slip away once the armband is worn.

In conclusion, we obtained the armband prototype shown in Fig. 2, with the dimension of the modules of 50.8 mm length, 22.5 mm height, and 28.5 mm or 23.5 mm width depending on the master or slave case, respectively.

G. PC Interface

Once the physical structure of the armband and its communication protocols were defined, we implemented the software for controlling the armband from a computer using the Python programming language. The code is divided into three layers with a bottom-up structure: the lowest layer corresponds to the BLE module, which establishes the connection with the armband and handles the data communication with it, accessing the server functionalities and dispatching the input/output packets; at the mid-layer, the armband module translates all the user commands into executable actions, driving the correspondent BLE antenna operations, and processes the received packet's content to extract the meaningful data for further usage; at the top layer, the application layer, the GUI features multiple widgets (e.g., buttons, spinners), to ease the interface with the user, and plots the requested acquisition data, i.e., the ATC values and the recognized gesture.

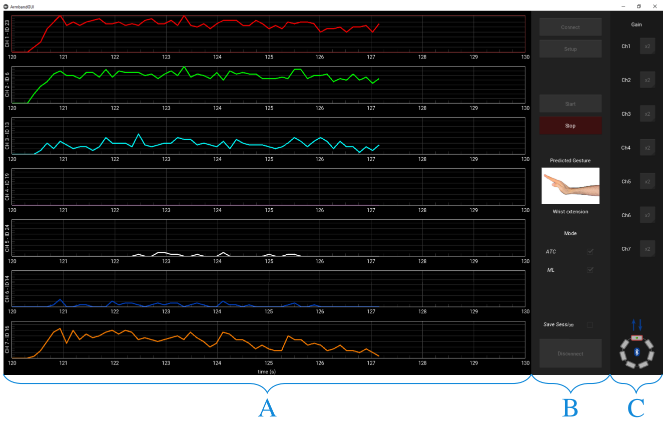

Fig. 6. Screenshot of the proposed GUI, composed of a graph for each armband channel (A), a control panel (B), and the gain selectors (C).

All the direct top-down functions are triggered by method callbacks of the objects, while the internal bottom-up data communication are based on queues. Indeed, each layer features one or multiple queues, depending on type of data to be transmitted (e.g., ATC or ML values, board errors), in order to obtain a stratified and organized data flow control.

Considering a laptop as the standard machine running the software (although the Python multi-platform nature extends its compatibility to multiple devices), the BLE connectivity is provided using an external USB-dongle (i.e., nRF52840 [51]) to guarantee the performance of the application by standardizing its wireless protocol stack.

A screenshot of the proposed GUI is reported in Fig. 6. The operational flow for running the application, once the device is worn and turned on, can be summarized as follows:

1) Connection: the USB-dongle is initialized as the central device, and the scan operation (with a *white-list* to filter undesired devices) is launched to search for the armband's advertising packet. If found, the connection is established, the armband's status notifications are enabled, and the IDs request (i.e., *req_ID*) is performed to check the correct hardware connectivity and to log the boards information. Lastly, seven ATC graphs are instantiated, one for each board belonging to the armband.
2) Setup: it consists of the ATC thresholds calibration process, which performs this task one board at a time along the armband loop (15 s ca. total); the user needs to maintain the forearm relaxed during this step.
3) Operation mode selection: if the ATC mode is chosen, the armband will stream the ATC values; vice versa, the ML mode only requests for the predicted gesture class. As previously introduced, the two modes can work concurrently.
4) Gain selection: the multiplier factor of each board's AFE can be adjusted proportionally to the captured signal's amplitude; default value is ×2 (i.e., 500 V/V).

5) Data acquisition: the data stream is started by enabling the notifications for the proper server characteristic (i.e., ATC and/or ML depending on the selected working mode(s)). Every time a packet arrives, if it contains ATC values, they are plotted on the graphs; otherwise, a picture showing the recognized gesture is displayed.
6) Data saving: the acquired data can be saved on files if the *Save Session* checkbox is active. The files are organized into sessions folders; within each one, every document reports the ATC values and/or the recognized class along with their timestamps.
7) Disconnection: the existing BLE link is terminated and the GUI can be closed.

H. Acquisition Protocol and Dataset Creation

Once the hardware and software components described in previous sections were finalized and fully tested, a data acquisition campaign was launched. Considering the hardware extraction of our event-driven feature, we focused on *in-vivo* tests to assess the actual performance of our system in real scenarios. In particular, 20 able-bodied people, 7 females and 13 males, aged between 23 and 29 years old, were involved. The experimental campaign accurately followed the ethical procedures specified in the Application n. 445136 approved by the Comitato Bioetico di Ateneo of the University of Turin [52]. Each participant was informed about the physical safety of the armband itself and about how their data would be anonymized and encrypted for the purposes allowed by the experimentation.

For each attending person, the armband is positioned on the right forearm, at one third of the line between the elbow and the wrist, with CH 1 (i.e., the master board) above the *Extensor Digitorum*. This position was previously determined, both according to literature reference positions [53], [54], [55] and by consulting with some fellow physicians about how deep and superficial muscles interact in this area. In particular, the forearm muscles have a very small body section along all their limb segment, resulting in a lot of cross actions between adjacent ones. However, while in the section near the wrist the muscles are mainly tendinous, resulting in weak sEMG signals almost comparable with the environmental noise [56], in the third proximal to the elbow the superficial ones have their principal body, from which the muscular activation can be sensed with an appreciable amplitude [54].

The final standardized position, starting from CH 1, involves the following muscles:

1) *Extensor digitorum*
2) *Extensor carpi ulnaris*
3) *Flexor digitorum profundus*
4) *Flexor carpi ulnaris*
5) *Flexor carpi radialis*
6) *Brachioradialis*
7) *Extensor carpi radialis*

After the armband is tighten up with the elastic band, the GUI is run in ATC mode to check if the acquisition channels are behaving correctly and if the obtained TC values are consistent with the position of each channel around the limb. In case one or more channels do not represent a physiological response from the underlying muscles (e.g., timings and amplitudes of muscular activation [57], [58]), the armband can be rotated slightly to fix an incorrect positioning. Furthermore, the gain of the misbehaving channels can be modified in order to enhance the sensitivity of the acquired information, most of the time going to balance particularly under- or over-developed forearms. This procedure is repeated until the acquired ATC profiles, as plotted in the GUI, are responsive to gestures change and their combination as a whole is coherent with the executed trial gestures.

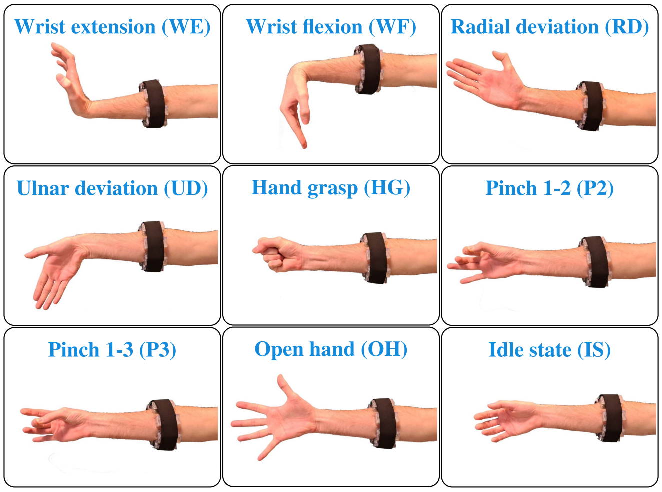

Fig. 7. Hand gestures analyzed in this study, ordered (from left to right and from top to bottom) as performed in the acquisition campaign.

Once the initial calibration is completed, the subject is asked to not lean the forearm on any surface and to keep it in the neutral position (i.e., with the fingers vertically overlapping). Then, each volunteer is instructed with the gestures s/he has to perform during the session, as depicted in Fig. 7, in this particular order:

1) **Wrist extension (WE)**: the hand has to be moved right, with its back going toward the distal section of the forearm;
2) **Wrist flexion (WF)**: the hand moves left, with the fingers moderately closing toward the inner part of the forearm;
3) **Radial deviation (RD)**: the hand is moved up, keeping the fingers adjacent to each other and the thumb as much relaxed as possible;
4) **Ulnar deviation (UD)**: the fingers point down toward the floor, maintaining themselves straight and aligned with the forearm;
5) **Hand grasp (HG)**: the fingers are bent towards the hand palm, resulting in a fist;
6) **Pinch 1-2 (P2)**: the thumb (i.e., finger 1) is internally rotated and the index finger (i.e., finger 2) is flexed to make their distal phalanges touch;
7) **Pinch 1-3 (P3)**: same intent of gesture 6, but involving the middle finger (i.e., finger 3) instead of the index finger;
8) **Open hand (OH)**: the fingers are stretched and opened out, trying to make the hand as tense as possible.

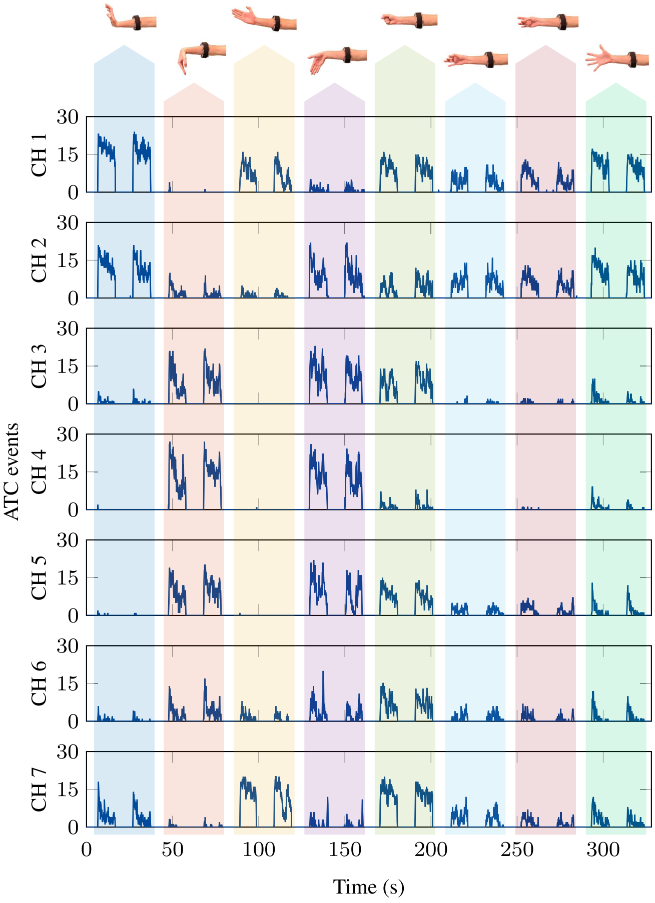

Fig. 8. ATC profiles during dataset acquisition.

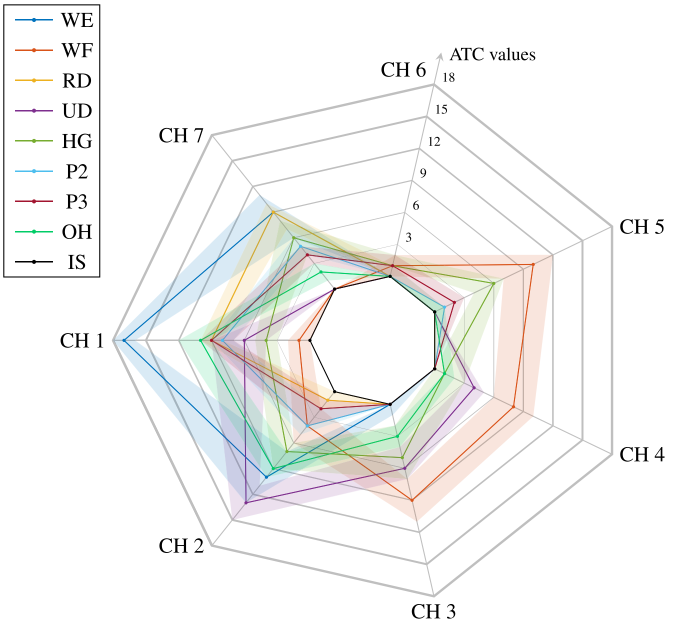

Fig. 9. Acquired dataset with idle norm set to 5. Each color represents a different gesture and the light-colored areas take into account from the 40th to the 60th percentile.

This sequence of gestures has to be performed respecting a specific timing in order to keep each gesture clearly distinct from the previous and following ones, making the post-processing phase more efficient. The subject is helped in following the time schedule by a custom application, which simply displays on the laptop monitor a countdown and a graphical indication of the gesture to be performed. In particular, after the supervisor starts the ATC mode to begin the signals acquisition, the following routine commence:

1) 15 s of rest are observed, in order to avoid glitches in the first part of the acquisition;
2) the gesture displayed by the application is performed and maintained for 10 s;
3) the hand is relaxed for 10 s to avoid muscular fatigue;
4) the previously executed gesture is repeated, again maintaining it for 10 s;
5) a 15 s rest period is observed, also being instructed about the following gesture;
6) step 2 to 5 are repeated until the last gesture is performed and the graphical help displays the end signal;
7) a 60 s rest period allows the subject to fully relax and stretch her/his hand;
8) step 2 to 7 are repeated for three times, in order to acquire a consistent amount of data.

In Fig. 8, the signals acquired during a typical acquisition session are plotted. Each subplot represents the ATC values obtained from one of the acquisition boards, ordered as in the physical position they occupy in the armband. Each gesture has a particular combination of the seven force indicators, with antagonist movements galvanizing the electrodes of opposite boards, as expected. As an example, WE presents muscular activation on CHs 1, 2, and 7, which are placed on the lateral section of the forearm, while WF activity is mainly sensed by CHs 3, 4, 5, and 6, which are positioned on the forearm's medial section.

At the end of the 20 acquisitions, performed in different days and with non-controlled ambient conditions, the anonymized data were processed to obtain a labeled dataset suitable for machine learning training. In particular, the signals from each subject have been segmented according to the timings detailed above, obtaining many 40 s slices, each one consisting of the two consequent repetitions of a single gesture, the rest period between them, and 5 s of rest for both the begin and end of each slice. Then, before labeling each slice, a threshold was established on the norm of each set of ATC values, as reported in (1), where f_x is the TC value of the x^{th} channel.

$$N = \sqrt{f_1^2 + f_2^2 + f_3^2 + f_4^2 + f_5^2 + f_6^2 + f_7^2} \tag{1}$$

The values whose norm was greater than or equal to the threshold were labeled according to the class performed at that particular timing, while the values whose norm was lower were identified as **Idle state (IS)** (i.e., the ninth class), as summarized in (2), where Th is the chosen threshold value.

$$Class = \begin{cases} \text{Actual class}, & N \geq Th \\ \text{Idle}, & N < Th \end{cases} \tag{2}$$

After this process we obtained a completely labeled dataset, which includes also the information on the rest periods of the acquisition sessions. In order to perform an exhaustive analysis on the effects of the chosen threshold on the final implementation of the ANN for the testing phase, we decided to extract one labeled dataset from the ATC raw data for each value of Th included in the range 1–30. Thus, after a proper selection of the idle threshold, the classifier for the final application would have the possibility to discern among a voluntary movement and little contractions due to body posture or to interactions with external objects. In Fig. 9, a representation of the dataset, obtained by setting the norm threshold to 5, is displayed by exploiting the parallel coordinates paradigm [59] wrapped in a heptagonal shape, thus recalling the spatial distribution of the activations of the forearm muscles.

I. *Offline Training*

The machine learning training phase, based on the multiple obtained datasets, was performed on the MATLAB environment. The ANN was primarily selected for its simple internal structure and the small size of its final MCU implementation w.r.t. other supervised algorithms (e.g., SVM, K-Nearest Neighbours (KNN) or Naive Bayes). Indeed, the complexity of an ANN is not dependent on the dimension of the dataset, but only on the information we need to extract from the inputs to obtain the outputs. Furthermore, the ANN has the intrinsic capability to extract latent information from the input features [60], thus allowing us to provide only the hardware-extracted ATC parameter, without any significant loss of accuracy in the prediction.

The ANN optimization routine was structured by taking advantage of the Deep Learning Toolbox$^{\text{TM}}$, which allows the user to precisely customize the overall network structure and the training parameters. In particular, we chose to perform a 5-fold cross-validation exploiting the capability of the Adam method [61] as gradient optimizer, with a variable learning rate starting from 0.03 and automatically decreasing every 10 epochs in order to improve training accuracy. Furthermore, a learning patience of 5 was selected, to stop the procedure when the validation error does not decrease over 5 consecutive samples, thus increasing the timing efficiency of the whole routine.

Once the main parameters of the learner were defined, we started an extensive analysis of the best implementation available varying the ANN physical structure and involving all the 30 processed dataset, while only fixing the Rectified Linear Unit (ReLU) as activation function. In particular, we made the optimizer try all the possible permutations going from 2 to 4 hidden layers and from 10 nodes for each layer up to 200, with steps of 10 nodes. Furthermore, each of the 60 different configurations was given as input all the 30 datasets with increasing idle norms, thus resulting in an overall analysis covering 1800 possibilities.

The aim of this hyper-parameter training phase was to identify the best solutions to maximize the prediction accuracy, while maintaining an acceptable user comfort. In fact, the increasing of the idle norm could bring to a final implementation which would require much more effort from the subjects to activate the desired gesture, or could even implicate the disappearance of one of the less powerful gestures from the dataset, thus making it impossible to be predicted during the online phase.

Furthermore, while the theoretical fluctuation in the ANN performance by varying the ANN structure or the idle norm threshold one at a time is known, the influence of one variable on the other is not trivial to be determined. Therefore, being the number of combinations small enough to be covered on a desktop computer in a reasonable time (e.g., 2–3 days), we decided to exhaustively perform this extensive analysis in order to avoid the risk of the final solution to be stuck in a local minimum of the loss function.

J. *Testing Phase*

After the proper definition of the ANN models which can be suitable for a real time usage, we proceeded to the final tests of our prototype. In particular, we performed another acquisition campaign, with the aim to evaluate the accuracy and the latency of the classifier, and the overall power consumption of the armband.

1) Online Testing Protocol: The testing campaign involved 6 healthy volunteers (none of whom were included in the training phase), 1 female and 5 males aged between 24 and 28 years old, who were informed about the purpose and conduct of our research.

Armband positioning was performed as reported in Section II-H, supervising the quality of acquired muscle activity and adjusting the armband placement accordingly. Since the armband performances are dependent on the ability of the user to perform the desired movement correctly, each subject was instructed prior to the testing phase in the proper movements execution, in accordance to what is described in Section II-H. Once the subject was ready, both ATC and gesture notifications were requested from the GUI in order to activate concurrent functional operations (as described in Section II-E). By doing so, the data received on the laptop, thanks to the packet number placed before both ATC and gesture payloads, were collected into ordered pairs; these doublets, after the end of each session, were used to evaluate if the predictions were correct.

The fulfillment of the testing phase required the subject to comply with the indications provided by an application displaying on a monitor the movements to be performed. The routine was the same as in Section II-H, except for step 8, where the number of repetitions was decreased to two instead of three. During this phase, the predicted output class was not displayed to the subject in order to avoid any conscious adjustments of the hand movements.

The entire process was repeated three times, updating the firmware of the predictor board (see Section II-B) with the ANN parameters obtained from the training procedures with an idle norm equal to 5, 10, and 15, respectively (see Section III-A). In particular, the network weights were saved in one dedicated array for each layer interface (e.g., with 2 hidden layers, we would create 3 arrays), and the prediction routine was resized and trimmed to the exact number of mathematical operations needed for the online propagation (i.e., 1 matrix multiplication and 1 ReLU activation function for each layer interface).

2) Setup for Timing Evaluation: The prediction latency was evaluated directly on the MCU of the predictor board, by involving one of the timer peripherals. In particular, during the setup phase of the armband, the `TIMER2` of the predictor is configured, taking in input a 6 MHz clock, in order to obtain the most precision possible. Then, during active operations, if the selected mode is ML, the timer is started just before the execution of the prediction and its value is read immediately after the predicted class is returned. The obtained timer value is converted in ms and transmitted via UART communication to a computer, which saves the delays and performs the average at the end of the test. In order to obtain a statistically significant database, this time analysis was performed continuously for about 5 min, thus acquiring more than 2000 samples.

3) Setup for Power Consumption Analysis: The last analysis we performed was the measure of the current absorbed by our developed armband. In order to maintain the size of the device as small as possible, we did not place a current sense resistor on the PCB during the design phase, so an external setup was to be prepared for the measurement. We chose to perform the analysis with the INA240 current sense amplifier by Texas Instruments [62], which, thanks to its enhanced switching noise rejection, better reacts to MCU state change w.r.t. a current probe or a multi-meter, which is even worse due to the RMS characteristic of its measurements. A 10 Ω sense resistor was placed at the amplifiers inputs and the INA240A1 version (i.e., 20 V/V gain) was selected. Last, the output of the amplifier was captured with a standard probe of the Rigol MSO5104 DSO [63], which allowed us to digitally save the current measurements for further processing.

In particular, we acquired 2 s of data for each one of the armband active states we considered more significant:

- Advertising: few seconds after being switched on, the master board starts advertising its presence to potential listening devices. The slave nodes are at this time inactive, with the MCU already in its deep sleep state;
- Connected: once a BLE client initializes a link with the armband, the master board enters the stable connection status. Both client and server nodes wait for the other to send some data and periodically check if the other peer is still active. The armband slave boards are still in deep sleep;
- ATC mode: when the BLE client requests for ATC notifications, the master module has to inform the slave boards of the requested task and all the seven boards start counting the TC events. Every 130 ms the master board sends a notification command to the lower levels of the BLE stack, which will effectively send the packet over the air at the next available connection interval;
- ML mode: as for the ATC mode, all the boards are active and counting. Furthermore, the predictor performs the ANN forward propagation every times it receives valid ATC values in the I^{2}C packet;
- ATC and ML mode combined: the boards effort is the same of the two previous modes, with the master sending both notifications, almost concurrently, to the BLE client.

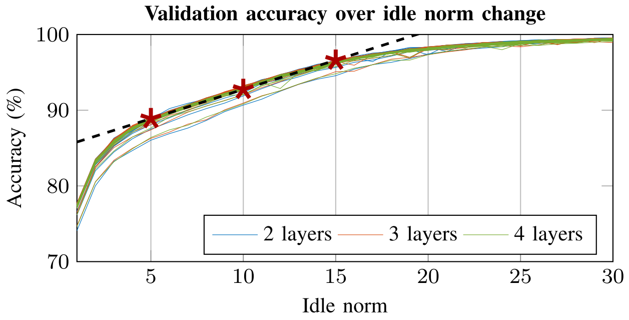

Fig. 10. ANN validation accuracy of the 1800 configurations trained. Each line represents a single network structure, which is trained with input dataset ranging from idle norm 1 up to 30.

III. Results and Discussion

In this section, the results of the test phase are reported and discussed, starting with the offline parameters definition for the ANN structure, through the accuracies obtained from the online acquisition campaign, and ending with the hardware performances of the armband, i.e., latency of the prediction and current absorption of the device.

A. Network Architecture Analysis

The hyper-parameters tuning, performed exploiting the Adam optimizer on MATLAB, brought to very homogeneous results across all the possible combinations of idle norm and hidden layers structures. In Fig. 10 all the 1800 accuracies obtained as output of the performed 5-fold cross-validation are plotted as function of the idle norm. It is possible to observe how the results of the different configurations overlap on the same trend, starting from about 77% with an idle norm of 1 up to more than 99% with a norm value of 30. The few exceptions to this trend are the configurations with 10 and 20 nodes, which are the ones slightly below the main curve, having the lowest validation accuracy of 74%.

Going into details of this uniform behavior, it is also possible to observe how the derivatives of the curves vary with the increasing of the norm. In particular:

- in the region between 1 and 5 the growing is almost logarithmic, with a fast increase from 1 to 2 and then an asymptotic slowdown;
- from 5 to 15 the accuracies follow an almost perfect linear trend (i.e., with an RMSE of 0.0056), as highlighted by the black line interpolation in Fig. 10, with an angular coefficient of 0.0077;
- from 15 to the end of our analysis the derivatives of the accuracies begin a further decline, with their values exiting the linear trend and going asymptotically to 100%.

All the above considered, we chose to take into account, for the final implementation, the configurations involving as inputs the dataset with idle norm of 5, 10, and 15. In fact, these three values are all in the linear region, having passed the first fast growing phase and being before the final slackening. Thus, they are selected as the more suitable options to analyze how the

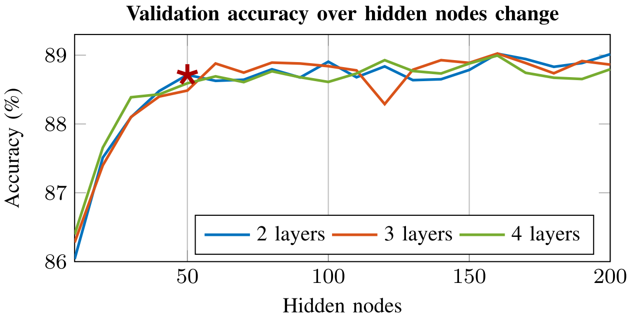

Fig. 11. ANN validation accuracy for a defined idle norm (i.e., 5). The performance behavior is highlighted at varying the network configuration.

differences in the norm affect the users effort during the online testing.

Then, the physical internal structure of the ANN was still to be selected. In Fig. 11, the validation accuracies of all the trained networks, with in input the dataset built on an idle norm of 5, are shown. The results obtained increasing the number of hidden layers are practically the same. In fact, all the three solutions present a high increase in accuracy through the first 5 configurations (i.e., moving from 10 to 50 hidden nodes) and then continue flat till the end of the analysis sweep. Furthermore, this trend of the validation accuracy for layers with more than 50 nodes is a clear representation of how those configurations contain redundant information, which would bring to a final implementation with similar accuracy but unnecessary resources usage.

Therefore, the configuration composed by 2 hidden layers with 50 nodes each has been chosen, being the best trade-off among complexity and performance, with an accuracy of 88.71%. The weights obtained from the training in this configuration were saved and successively used for the online testing phase on the MCU. Furthermore, thanks to the reduced size of the selected configuration (i.e., about 15 kB, considering each weight mapped on a 32-bit floating-point number), it was possible to maintain the RAM active size of the predictor down to 96 kB, thus avoiding power squandering.

B. Online Testing Performance

The data collected from the testing campaign (see Section II-J1) are summarized as reported in Fig. 12, and divided according to the three different tested idle norms. In particular, confusion matrices (Fig. 12, left), accuracy, precision, recall, and f1-score metrics [64] (Fig. 12, right) are disclosed.

To this end, the classes *predicted* by the armband were compared with the *true* classes obtained by segmenting the acquired signals. The segmentation of the movements and the subsequent *true* labeling are based on what was indicated by the support application used during the testing phase. The *true* IS class was labeled as described in Section II-H, by setting the same idle norm applied to the training dataset of the tested ANN. The reported confusion matrices were computed by summing those obtained from the six involved volunteers.

TABLE III
EVALUATION METRICS FOR MERGED PINCHES CLASS

Idle norm	Evaluation metrics (%)			
	Accuracy	Precision	Recall	F1-score
5	82.85	61.62	51.80	56.28
10	85.80	56.84	37.61	45.27
15	88.36	52.23	26.01	34.72

In order to balance the testing dataset, only 40 s of IS were considered since this is the time spent for each movement during the testing phase. Clearly, the dataset does not contain the same number of total elements for all the classes because of the physiological reaction times of the subjects, who do not respond instantaneously to the changes of the displayed gestures. Moreover, since the labeling process comes after the idle norm operation, the total number of labeled movements could differ for low and high idle norm because the associated muscles activities could be too weak to be labeled differently from idle (e.g., for the two pinches).

Looking at Fig. 12, and considering the evaluation metrics overall, the movement with the best outcomes is WF. It resulted in accuracies of 97.26%, 97.84%, and 98.83% for the three idle norms analyzed, respectively.

On the other hand, P2 and P3 show the worst results. Although the accuracies of the two pinches are higher than that of OH, the obtained values of precision, recall, and consequently f1-score are lower, especially in the case of idle norm equal to 15. Considering the similarity of the two gestures, which often leads to interchange the predicted class between them (as evidenced by the confusion matrices), we also report in Table III the evaluation metrics obtained by merging these two gestures into the same class. Despite a worsening of accuracy for the merged pinches class (the number of true negatives decreased more than the increase of true positives), which, however, is still above 80%, precision and recall improved considerably. Indeed, in correspondence of idle norm equal to 15, we obtained a recall value of 26.01% w.r.t. the lower 9.1% resulted by predicting P2 as distinct gesture.

As mentioned before, OH is among the gestures that, together with the two pinches, provides not totally satisfactory outcomes. However, while for the two pinches the difficulty lies in their poor amount of predictions, the reasons for the low performance of OH are caused by the quite high presence of misclassifications of other gestures with it (as can be seen from the confusion matrices). As an explanation for this behavior, it was observed that the volunteers who took part in the testing phase tended to extend their fingers during the execution of other gestures, obtaining values of muscular activation such as to involve the misclassification of those movements with OH.

Although using the idle norm equal to 5 results in the lowest values among the reported evaluation metrics, the overall outcomes of this test scenarios (including volunteers' feedback and considering the data distribution) makes us consider it the most efficient in an application context (e.g., to control an HMI). Indeed, using higher idle norms has the consequence of losing many movements samples, such as reported for the case of the

Idle norm = 5

True class \ Predicted class	WE	WF	RD	UD	HG	P2	P3	OH	IS
WE	1637	7	49	13	32	49	83	35	1
WF	4	1515	13	109	61	6	71	75	3
RD	20	22	768	173	90	188	325	157	4
UD	87	37	42	1416	8	17	43	291	2
HG	10	48	116	20	1421	22	135	272	1
P2	437	4	181	17	22	257	561	320	2
P3	125	1	101	50	23	247	794	446	1
OH	432	1	11	32	138	71	148	1069	1
IS	0	0	1	0	0	0	0	0	1848

Evaluation metrics (%)

	Accuracy	Precision	Recall	F1-score
WE	91.78	59.48	85.89	70.29
WF	97.26	92.66	81.58	86.77
RD	91.13	59.91	43.96	50.71
UD	94.41	77.38	72.88	75.06
HG	94.07	79.16	69.49	74.01
P2	87.27	29.99	14.27	19.34
P3	85.98	36.76	44.41	40.22
OH	85.57	40.11	56.17	46.80
IS	99.90	99.19	99.95	99.57
Avg.	**91.93**	**63.85**	**63.18**	**62.53**

Idle norm = 10

True class \ Predicted class	WE	WF	RD	UD	HG	P2	P3	OH	IS
WE	1652	3	60	6	1	45	50	43	0
WF	0	1482	2	30	7	1	2	34	27
RD	3	83	743	60	157	47	274	77	19
UD	14	84	39	1465	0	6	9	181	7
HG	2	15	14	31	1650	7	13	135	6
P2	241	3	20	30	25	303	77	333	13
P3	39	11	109	30	147	271	188	367	24
OH	242	6	6	48	79	158	25	1040	16
IS	0	0	0	0	0	0	0	0	1848

	Accuracy	Precision	Recall	F1-score
WE	94.76	75.33	88.82	81.52
WF	97.84	87.85	93.5	90.59
RD	93.21	74.82	50.79	60.50
UD	95.97	86.18	81.16	83.59
HG	95.53	79.86	88.09	83.78
P2	91.06	36.16	29.0	32.18
P3	89.86	29.47	15.85	20.61
OH	87.75	47.06	64.20	54.31
IS	99.22	94.29	100.0	97.06
Avg.	**93.91**	**67.89**	**67.93**	**67.13**

Idle norm = 15

True class \ Predicted class	WE	WF	RD	UD	HG	P2	P3	OH	IS
WE	1640	1	17	3	2	27	32	82	23
WF	0	1413	0	20	15	0	4	32	28
RD	0	7	533	3	148	68	182	18	47
UD	30	6	4	1339	2	0	0	179	16
HG	2	19	3	23	1383	5	2	177	37
P2	141	0	48	69	48	61	72	205	26
P3	49	7	84	11	81	127	115	258	40
OH	151	3	10	24	43	12	11	985	14
IS	0	0	0	0	0	0	0	0	1848

	Accuracy	Precision	Recall	F1-score
WE	95.38	81.47	89.76	85.42
WF	98.83	97.05	93.45	95.22
RD	94.73	76.25	52.98	62.52
UD	96.78	89.75	84.96	87.29
HG	94.99	80.31	83.77	82.0
P2	93.0	20.33	9.10	12.58
P3	92.08	27.51	14.90	19.33
OH	89.94	50.88	78.61	61.77
IS	98.09	88.89	100.0	94.12
Avg.	**94.87**	**68.05**	**67.5**	**66.69**

Fig. 12. Confusion matrices and evaluation metrics concerning the entire testing dataset. Obtained results are divided according the three idle norms used, i.e., 5, 10, and 15, respectively.

two pinches. Moreover, using lower values for idle norm allows the armband user to achieve the classification of the gesture being performed with less muscular effort. Furthermore, considering the IS row of the confusion matrix related to idle norm equal to 5, the number of idles misclassified as an active movement is already practically equal to zero. Therefore, there is no stringent need to use high idle norms, even to cope with the noise of acquired muscle activity.

C. *Online Timing Analysis*

The timing latencies measured using the timer on the predictor board, involving the idle norm equal to 5, are analyzed in details. The obtained values are practically constant, with an average of 1.342 ms. The stable outcome of this measure is mainly due to two factors: the intrinsic nature of the ANN structure and the uninterrupted sequential operations of the CPU performing the requested matrix multiplications. In particular, regarding the first point, the ANN forward propagation is more deterministic w.r.t. other classifiers' implementation, since it always requests the same amount of multiplications/activations, independently from the input values (on the other hand, e.g., a RF prediction can provide the output class at any available leaf, depending on the magnitude of the input features, thus causing a more significant deviation in latency). Furthermore, considering the second aspect, the removal of the interrupt on the TC signal brought the CPU back to a completely deterministic behavior, allowing the execution of any piece of code without the possibility of unexpected interruptions.

The application latency of a possible HMI which would involve our armband as human control sensor would be of 131.3 ms, considering in that case the ATC window contribution would be added to the predictor latency alone. Even considering this overall delay, our armband will still react a lot faster than the 300 ms threshold, commonly considered for the real-time control of robots and prostheses [65].

D. Online Power Performance

The last evaluated performance metric is the current absorption of the armband, considered in its completeness. The overall results, obtained with the setup detailed in Section II-J3, are shown in Table IV. The recorded current absorption values have a quite low variance, with the lowest value during advertising (i.e., 2.712 mA) and the highest during the active mode involving both notifications sending (i.e., 3.029 mA), as supposed. Furthermore, the reported values are coherent w.r.t. the ones obtained in [43], when measured on a single board. However, the current drained in this application is lower to the expected value of 3.36 mA (i.e., 7 × 0.48 mA), thanks to the differentiation among the firmware of the master and slave boards, which allows the latters to enter the deep sleep state and save power.

In order to confirm the correct behavior of the setup and of the armband itself, we analyzed in details the digitized waveforms. In Fig. 13, a representative window of 300 ms, captured during ML mode, is reported. As it is possible to observe, the variations of the current requested by the armband are coherent with the specifications of the BLE protocol and with the expected behavior of the armband itself. In particular, few regular events alter the baseline:

- The double bump waveforms starting at about 1.51 s, 1.64 s and 1.77 s are to be attributed to the master board handling the end of each ATC window (i.e., every 130 ms) and to the consequent I^2C activity. Indeed, during that arc of time, the six slave boards have to wake up and communicate with the following one, and the predictor has also to perform the ML computations. The width of the waveform during this type of activity is related to the low speed of the MCU clock (i.e., 24 MHz) and of the I^2C module, which exchanges data at 400 kHz. However, the amplitude is limited to 2 mA, thus having a modest impact on the average consumption;
- The highest peaks, which reach 11 mA, are related to the sending of the BLE notifications, which requires the RF module to be powered on, even if for a very small amount of time. These peaks are always at steps of 50 ms due to the selected connection interval, and, thanks to a slave latency of 4, they appear only after a master action occurs within the previous connection interval (e.g., no transmission is visible at 1.75 ms);
- Peaks similar to the previous ones but slightly lower in amplitude, reaching almost 10 mA, are to be attributed to a BLE check for further packets, acknowledging to have received the previous notification packet. Most probably they have the same amplitude of the notifications ones, but their very short duration makes them too fast to be entirely captured by the DSO sampling. Also these peaks follow the timing of the BLE connection, being aligned to the 50 ms ticks too.

All above considered, the calculated mean current absorption of 2.92 mA is practically represented by the baseline of the graph, since the three main activities of the armband have a current request either small in amplitude or short in duration, thus not affecting the average value very much. In conclusion, the armband can continuously operate in ML mode for about 60 h, without needing a battery recharge, thus allowing its inclusion in outdoor activities, even if spread on more than one day.

TABLE IV
CURRENT ABSORPTION IN DIFFERENT OPERATING CONDITIONS

	Advertising	Connected	Active mode		
			ATC	ML	Both
C.A.[1] **(mA)**	2.712	2.750	2.986	2.920	3.029
O.T.[2,3] **(h)**	64.5	63.6	58.6	59.9	57.8

[1] Current Absorption, [2] Operating Time

[3] Considering our 175 mAh Lithium Polymer Battery

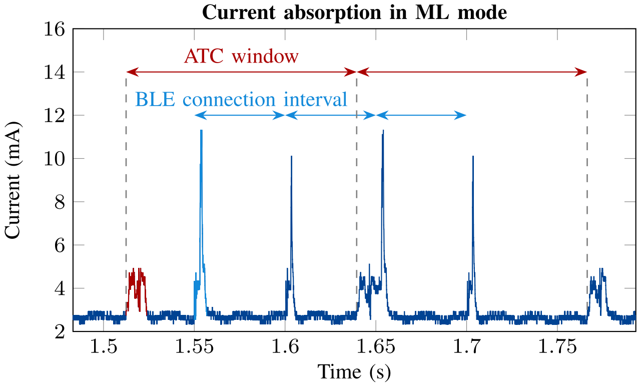

Fig. 13. Profile of the current absorbed by the armband, whose average measured value is 2.92 mA. Both the consumption concerning ATC data management (red, at 1.51 s) and the peaks related to the BLE communication (light blue, at 1.55 s) are distinguishable.

IV. Comparison With SoA Works

Having analyzed in details the performance and the overall behavior of our armband prototype in Section III, here we conclude the discussion by inserting our work into the perspective of other literature studies. Since the state of the art regarding sEMG-based hand gesture recognition is very broad, as we already discussed in the introduction of this manuscript, we selected the works we believe the more relevant for the sake of our comparison. In Table V, we reported two of the predecessors [38], [40] of this study and seven works of other research groups, all published in the last four years. The variety of applications and, sometimes, the ambiguous phrasing of some results make the performances indicators (i.e., column 9–14) not totally populated. However, despite the lack of data for some indexes, we did our best in analyzing each metric in detail, critically discussing its relevance w.r.t. the target application.

All the reported works developed their wearable device, but only [7], [16], [29], [40] managed to fit their learner into an embedded prototype. The armband developed in [9], for example, is reported to be designed to support the embedding of ML algorithms, but all the predictions are still evaluated on a computer. Similarly, [3], [23] use the Myo armband as acquisition device, sending data via wireless communication to an

TABLE V
COMPARISON WITH STATE OF THE ART sEMG-BASED HAND GESTURE CLASSIFIERS

Work	Year	*f*	#Ch.	#Mov.	Armband	Embedded Classifier	Classifier Type	Accuracy (%)	Latency (ms)	V_{DD} (V)	C.A.[1] (mA)	B.C.[2] (mAh)	O.T.[3] (h)
[7]	2018	DWT[4]	4	5		✓	SVM	94.0	$256^{\diamond}$ $33^{\star}$	n.a.	n.a.	200	129
[9]	2019	raw sEMG	10	11	✓		ConvNet	89.5	$250^{\diamond}$	n.a.	n.a.	100	6
[40]	2020	ATC	3	6		✓	ANN	96.3	$130^{\diamond}$ $8.5^{\star}$	n.a.	n.a.	n.a.	n.a.
[29]	2020	multiple	8	4	✓	✓	LDA[5]	>90	$150^{\diamond}$	3.7	n.a.	380	n.a.
[3]	2020	multiple	8	6	✓		ANN	94.7	$200^{\diamond}$ $138^{\star}$	n.a.	n.a.	n.a.	n.a.
[23]	2020	raw sEMG	8	7	✓		Conv-GRU[6]	82.2	$39^{\star}$	n.a.	n.a.	n.a.	n.a.
[16]	2020	custom	32	8		✓	CNN	98.2	$5^{\star}$	3.3	15	n.a.	n.a.
[33]	2021	raw sEMG	128	30			CNN	92.4	$100^{\diamond}$	n.a.	n.a.	n.a.	n.a.
[38]	2021	ATC	2	3			ANN	97.9	$130^{\diamond}$	3.7	0.96	220	230
This[7]	2022	ATC	7	9	✓	✓	ANN	91.9	$130^{\diamond}$ $1.34^{\star}$	3.7	2.92	175	60

[1] Current Absorption, [2] Battery Capacity, [3] Operating Time, [4] Discrete Wavelet Transform, [5] Linear Discriminant Analysis $^{\diamond}$Acquisition window
[6] Gated Recurrent Unit, [7] Considering the most comfortable use-case with idle norm equal to 5 *Prediction

elaboration unit, and [33] still involves electrodes matrices wired to a data collection unit, which sends the processed information to a computer for the further prediction. Also in our introductory work [38] we did not perform ML computations on-board, since we were still employing standalone modules.

Regarding the implemented algorithms, the totality of the computer-based works involve the NN structure in their predictions. Many research groups exploit the capability of the CNN paradigm, even with some enhanced features added, while only [3] maintains the simpler ANN structure. On the other hand, all the embedded solutions fell back on more compact classifiers, like smaller ANNs [40], SVM [7] or even the Linear Discriminant Analysis (LDA) [29]. The only work that was able to perform a CNN computation on a microcomputer inside a prosthesis is [16], which, applying 32 High-Density (HD) sEMG electrodes on the skin covered by the prosthesis body, was able to recognize 8 gestures with 98.2% accuracy. Apart from this study, the numbers of channels used to predict the gestures are pretty stable across the literature, maintaining a 1 vs 1 channels/gestures ratio with a value around 8 (probably due to the big influence the Myo armband has on the field at this moment). The other few exceptions found are represented by [9], in which a 10-channel custom armband is developed, recognizing 11 different hand movements with an overall accuracy of 89.5%, and [33], whose HD electrodes matrices allows the researchers to predict (on a computer) 30 movements with an average accuracy of 92.4%. The accuracies of the remaining studies are all around the 90%, with variations due to more channels involved or to less complexity of the output space.

On the other hand, the prediction latency presents wider variations, with a few values missing, going from the lowest delay of our work (i.e., 1.34 ms) and increasing up to 138 ms for [3]. Nevertheless, these values have to be interpreted according to the indication of how they are sampled, in order to obtain a meaningful relation. Indeed, [3], [7] reported both the prediction time and the length of the acquisition window needed by the algorithm, but [9], [29], [33] only wrote down their acquisition window and no indication on the prediction latency can be found. Furthermore, in [16], [23] it is possible to identify the prediction latency but there is no information about how much time is requested for the acquisition of the features. In our case, both considering the pure prediction latency and adding the ATC window needed for the TC computation (i.e., a total of 131.3 ms), we are still well competitive w.r.t. the other works, with a delay of about the half of the others'.

Last, regarding power consumption and consequent operating time, we have very poor data from the other works to be able to perform a significant analysis. Apart from our previous work, we found data only from three different studies: the CNN implementations of [9], [16] brought their prototype battery life down to 6 h or even less, while [7] has the longer operating time (among embedded solutions), with a battery life of about 129 h. However, since the duration of their acquisition window is twice as long as ours, the reported hours of operation are comparable with the battery duration we obtained, i.e., 60 h.

All the above considered, we fit pretty well into present state of the art literature, not with only top scores, but with a good balance of all the considered metrics. So, to the best of our knowledge, our prototype can fulfill its purpose in the field of wearable armband for HMI applications, giving to the users a wider choice when selecting the device needed for rehabilitation as well as robotic control.

V. CONCLUSION

In this work, we introduced the armband we designed for HMI applications, aiming to provide a competitive solution for hand gesture recognition w.r.t. the current state of the art. The presented armband is based on the bio-inspired technique known as ATC, which allows our device to be low-power and to reduce the complexity of the input space for classification tasks. In particular, this paper describes the modular physical structure of the armband, the functioning of its hardware/software components, including the communication protocol adopted, and the properties of its embedded machine learning algorithm, i.e., an ANN.

We tested the armband with multiple configurations of ANN, and the one we considered most efficient led to a testing accuracy equal to 91.9%. The latencies due to the prediction resulted to be 1.34 ms, which is two orders of magnitude lower than the maximum implicit delay of ATC (i.e., the 130 ms window), proving the usability of our approach for real-time applications. Moreover, thanks to the low current absorption, which is equal to 2.92 mA, the armband has an operating time of up to 60 h.

Future perspectives for the designed armband involve the need and interest to test it with further classification algorithms, such as SVM, DT or KNN, in order to analyze which is the most suitable solution. Moreover, we are planning to develop a serious game to validate the use of our device in an application scenario besides evaluating its classification performances.

Acknowledgment

The authors would like to thank their past master students Federica Pasquali, Andrea Zimara, Matteo Tolomei, and Vincenzo Barresi (in reverse chronological order of the thesis conclusion), for their effort towards the realization of this project, and all the volunteers who participated in this study.

References

[1] S. Jiang, P. Kang, X. Song, B. P. Lo, and P. B. Shull, "Emerging wearable interfaces and algorithms for hand gesture recognition: A survey," *IEEE Rev. Biomed. Eng.*, vol. 15, pp. 85–102, 2022.

[2] P. Visconti, F. Gaetani, G. Zappatore, and P. Primiceri, "Technical features and functionalities of Myo armband: An overview on related literature and advanced applications of myoelectric armbands mainly focused on arm prostheses," *Int. J. Smart Sens. Intell. Syst.*, vol. 11, no. 1, pp. 1–25, 2018.

[3] Z. Xu, L. Shen, J. Qian, and Z. Zhang, "Advanced hand gesture prediction robust to electrode shift with an arbitrary angle," *Sensors*, vol. 20, no. 4, 2020, Art. no. 1113. [Online]. Available: https://www.mdpi.com/1424-8220/20/4/1113

[4] B. Kundu and D. Subarram Naidu, "Classification and feature extraction of different hand movements from EMG signal using machine leaning based algorithms," in *Proc. Int. Conf. Elect., Commun., Comput. Eng.*, 2021, pp. 1–5.

[5] X. Hu, H. Zeng, A. Song, and D. Chen, "Robust continuous hand motion recognition using wearable array myoelectric sensor," *IEEE Sensors J.*, vol. 21, no. 18, pp. 20596–20605, Sep. 2021.

[6] C. Tepe and M. Erdim, "Classification of EMG finger data acquired with Myo armband," in *Proc. Int. Congr. Hum.- Comput. Interact., Optim. Robot. Appl.*, 2020, pp. 1–4.

[7] V. Kartsch, S. Benatti, M. Mancini, M. Magno, and L. Benini, "Smart wearable wristband for based gesture recognition powered by solar energy harvester," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2018, pp. 1–5.

[8] M. Ergeneci, K. Gokcesu, E. Ertan, and P. Kosmas, "An embedded, eight channel, noise canceling, wireless, wearable sEMG data acquisition system with adaptive muscle contraction detection," *IEEE Trans. Biomed. Circuits Syst.*, vol. 12, no. 1, pp. 68–79, Feb. 2018.

[9] U. Côté-Allard, G. Gagnon-Turcotte, F. Laviolette, and B. Gosselin, "A low-cost, wireless, 3-D-printed custom armband for sEMG hand gesture recognition," *Sensors*, vol. 19, no. 12, Art. no. 2811, 2019. [Online]. Available: https://www.mdpi.com/1424-8220/19/12/2811

[10] A. T. Ossaba, J. J. J. Tigreros, and J. C. T. Orjuela, "Open source multichannel EMG armband design," in *Proc. IX Int. Congr. Mechatronics Eng. Automat.*, 2020, pp. 1–6.

[11] S. Jiang, Q. Gao, H. Liu, and P. B. Shull, "A novel, co-located EMG-FMG-sensing wearable armband for hand gesture recognition," *Sensors Actuators A: Phys.*, vol. 301, 2020, Art. no. 111738. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0924424719308891

[12] O. Ozturk, A. Golparvar, and M. K. Yapici, "Smart armband with graphene textile electrodes for EMG-based muscle fatigue monitoring," in *Proc. IEEE Sensors*, 2021, pp. 1–4.

[13] Z. Lu, X. Chen, Q. Li, X. Zhang, and P. Zhou, "A hand gesture recognition framework and wearable gesture-based interaction prototype for mobile devices," *IEEE Trans. Human-Mach. Syst.*, vol. 44, no. 2, pp. 293–299, Apr. 2014.

[14] T. Phienthrakul, "Armband gesture recognition on electromyography signal for virtual control," in *Proc. 10th Int. Conf. Knowl. Smart Technol.*, 2018, pp. 149–153.

[15] M. Woźniak, P. Pomykalski, D. Sielski, K. Grudzień, N. Paluch, and Z. Chaniecki, "Exploring EMG gesture recognition-interactive armband for audio playback control," in *Proc. Federated Conf. Comput. Sci. Inf. Syst.*, 2018, pp. 919–923.

[16] S. Tam, M. Boukadoum, A. Campeau-Lecours, and B. Gosselin, "A fully embedded adaptive real-time hand gesture classifier leveraging HD-sEMG and deep learning," *IEEE Trans. Biomed. Circuits Syst.*, vol. 14, no. 2, pp. 232–243, Apr. 2020.

[17] M. Arozi, M. Ariyanto, A. Kristianto, Munadi, and J. D. Setiawan, "EMG signal processing of Myo armband sensor for prosthetic hand input using RMS and ANFIS," in *Proc. 7th Int. Conf. Inf. Technol., Computer, Elect. Eng.*, 2020, pp. 36–40.

[18] O. W. Samuel et al., "Intelligent EMG pattern recognition control method for upper-limb multifunctional prostheses: Advances, current challenges, and future prospects," *IEEE Access*, vol. 7, pp. 10150–10165, 2019.

[19] S. Kim et al., "Development of an armband EMG module and a pattern recognition algorithm for the 5-finger myoelectric hand prosthesis," *Int. J. Precis. Eng. Manuf.*, vol. 20, pp. 1997–2006, 2019. [Online]. Available: https://doi.org/10.1007/s12541-019-00195-w

[20] S. Singhvi and H. Ren, "Comparative study of motion recognition with temporal modelling of electromyography for thumb and index finger movements aiming for wearable robotic finger exercises," in *Proc. 3rd Int. Conf. Adv. Robot. Mechatronics*, 2018, pp. 509–514.

[21] A. B. H. Amor, O. E. Ghoul, and M. Jemni, "Sign language handshape recognition using Myo armband," in *Proc. 7th Int. Conf. ICT Accessibility*, 2019, pp. 1–5.

[22] S. Kim, J. Kim, S. Ahn, and Y. Kim, "Finger language recognition based on ensemble artificial neural network learning using armband EMG sensors," *Technol. Health Care: Official J. Eur. Soc. Eng. Med.*, vol. 26, pp. 249–258, 2018.

[23] N. Nasri, S. Orts-Escolano, and M. Cazorla, "An sEMG-controlled 3D game for rehabilitation therapies: Real-time time hand gesture recognition using deep learning techniques," *Sensors*, vol. 20, no. 22, 2020, Art. no. 6451. [Online]. Available: https://www.mdpi.com/1424-8220/20/22/6451

[24] L. van Dijk, C. K. van der Sluis, H. W. van Dijk, and R. M. Bongers, "Learning an EMG controlled game: Task-specific adaptations and transfer," *PLoS ONE*, vol. 11, 2016, Art. no. e0160817.

[25] P. N. Müller et al., "Flex your muscles: EMG-based serious game controls," in *Serious Games*. Berlin, Germany: Springer, 2020, pp. 230–242.

[26] S. S. Esfahlani, B. Muresan, A. Sanaei, and G. Wilson, "Validity of the kinect and Myo armband in a serious game for assessing upper limb movement," *Entertainment Comput.*, vol. 27, pp. 150–156, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1875952117300952

[27] F. Rossi, F. Savi, A. Prestia, A. Mongardi, D. Demarchi, and G. Buccino, "Combining action observation treatment with a brain-computer interface system: Perspectives on neurorehabilitation," *Sensors*, vol. 21, no. 24, 2021, Art. no. 8504. [Online]. Available: https://www.mdpi.com/1424-8220/21/24/8504

[28] A. Baraka, H. Shaban, M. Abou El-Nasr, and O. Attallah, "Wearable accelerometer and sEMG-based upper limb BSN for tele-rehabilitation," *Appl. Sci.*, vol. 9, no. 14, 2019, Art. no. 2795. [Online]. Available: https://www.mdpi.com/2076-3417/9/14/2795

[29] Z. Bi et al., "Wearable EMG bridge–A multiple-gesture reconstruction system using electrical stimulation controlled by the volitional surface electromyogram of a healthy forearm," *IEEE Access*, vol. 8, pp. 137330–137341, 2020.

[30] A. Prestia et al., "Motion analysis for experimental evaluation of an event-driven FES system," *IEEE Trans. Biomed. Circuits Syst.*, vol. 16, no. 1, pp. 3–14, Feb. 2022.

[31] F. Rossi, P. Motto Ros, R. M. Rosales, and D. Demarchi, "Embedded biomimetic system for functional electrical stimulation controlled by event-driven sEMG," *Sensors*, vol. 20, no. 5, 2020, Art. no. 1535. [Online]. Available: https://www.mdpi.com/1424-8220/20/5/1535

[32] J. O. Pinzón-Arenas, R. Jiménez-Moreno, and J. E. Herrera-Benavides, "Convolutional neural network for hand gesture recognition using 8 different EMG signals," in *Proc. 22th Symp. Image, Signal Process. Artif. Vis.*, 2019, pp. 1–5.

[33] X. Chen, Y. Li, R. Hu, X. Zhang, and X. Chen, "Hand gesture recognition based on surface electromyography using convolutional neural network with transfer learning method," *IEEE J. Biomed. Health Inform.*, vol. 25, no. 4, pp. 1292–1304, Apr. 2021.

[34] E. Donati et al., "Processing EMG signals using reservoir computing on an event-based neuromorphic system," in *Proc. IEEE Biomed. Circuits Syst. Conf.*, 2018, pp. 1–4.

[35] M. Tavakoli, C. Benussi, P. A. Lopes, L. B. Osorio, and A. T. de Almeida, "Robust hand gesture recognition with a double channel surface EMG wearable armband and SVM classifier," *Biomed. Signal Process. Control*, vol. 46, pp. 121–130, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1746809418301794

[36] A. Phinyomark and E. Scheme, "A feature extraction issue for myoelectric control based on wearable EMG sensors," in *Proc. IEEE Sensors Appl. Symp.*, 2018, pp. 1–6.

[37] Y. Hu, Y. Wong, Q. Dai, M. Kankanhalli, W. Geng, and X. Li, "sEMG-based gesture recognition with embedded virtual hand poses and adversarial learning," *IEEE Access*, vol. 7, pp. 104108–104120, 2019.

[38] M. Becchio et al., "Live demonstration: Event-driven hand gesture recognition for wearable human-machine interface," in *Proc. IEEE Biomed. Circuits Syst. Conf.*, 2021, p. 1.

[39] S. Sapienza et al., "On-line event-driven hand gesture recognition based on surface electromyographic signals," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2018, pp. 1–5.

[40] A. Mongardi, P. Motto Ros, F. Rossi, M. Ruo Roch, M. Martina, and D. Demarchi, "A low-power embedded system for real-time sEMG based event-driven gesture recognition," in *Proc. 26th IEEE Int. Conf. Electron., Circuits, Syst.*, 2019, pp. 65–68.

[41] A. Mongardi et al., "Live demonstration: Low power embedded system for event-driven hand gesture recognition," in *Proc. IEEE Biomed. Circuits Syst. Conf.*, 2019, p. 1.

[42] D. A. Fernandez Guzman, S. Sapienza, B. Sereni, and P. Motto Ros, "Very low power event-based surface EMG acquisition system with off-the-shelf components," in *Proc. IEEE Biomed. Circuits Syst. Conf.*, 2017, pp. 1–4.

[43] F. Rossi, A. Mongardi, P. Motto Ros, M. Ruo Roch, M. Martina, and D. Demarchi, "Tutorial: A versatile bio-inspired system for processing and transmission of muscular information," *IEEE Sensors J.*, vol. 21, no. 20, pp. 22285–22303, Oct. 2021.

[44] M. Crepaldi et al., "A quasi-digital radio system for muscle force transmission based on event-driven IR-UWB," in *Proc. IEEE Biomed. Circuits Syst. Conf.*, 2012, pp. 116–119.

[45] S. Sapienza, M. Crepaldi, P. Motto Ros, A. Bonanno, and D. Demarchi, "On integration and validation of a very low complexity ATC UWB system for muscle force transmission," *IEEE Trans. Biomed. Circuits Syst.*, vol. 10, no. 2, pp. 497–506, Apr. 2016.

[46] Ambiq Micro. Apollo3 blue, 2022. Accessed: Sep. 22, 2022. [Online]. Available: https://ambiq.com/apollo3-blue/

[47] Bluetooth SIG Inc. Bluetooth specification version 4.2, 2014. Accessed: Sep. 22, 2022. [Online]. Available: https://www.bluetooth.com/specifications/specs/core-specification-4-2/

[48] Amazon Web Services, Inc. FreeRTOS: Real-time operating system for microcontrollers. Accessed: Sep. 22, 2022. [Online]. Available: https://www.freertos.org/

[49] Arm. CMSIS library. Accessed: Sep. 22, 2022. [Online]. Available: https://developer.arm.com/tools-and-software/embedded/cmsis

[50] Cellevia Batteries. Polymer Lithium-Ion battery, 2016. Accessed: Sep. 22, 2022. [Online]. Available: https://www.tme.eu/Document/040b757b80bd482e7b6e1bb3f31fe8a6/cel0062.pdf

[51] Nordic Semicondutor. nRF52840 dongle. Accessed: Sep. 22, 2022. [Online]. Available: https://www.nordicsemi.com/Products/Development-hardware/nrf52840-dongle

[52] Università degli Studi di Torino. Comitato di bioetica dell'ateneo, 2008. Accessed: Sep. 22, 2022. [Online]. Available: https://www.unito.it/ricerca/strutture-e-organi-la-ricerca/comitato-di-bioetica-dellateneo

[53] J. Tomczyński, T. Mańkowski, and P. Kaczmarek, "Influence of sEMG electrode matrix configuration on hand gesture recognition performance," in *Proc. Signal Process.: Algorithms, Architectures, Arrangements, Appl.*, 2017, pp. 42–47.

[54] H. Hermens et al., "SENIAM project," 1996. Accessed: Sep. 22, 2022. [Online]. Available: http://www.seniam.org/

[55] M. Barbero, R. Merletti, and A. Rainoldi, *Atlas of Muscle Innervation Zones*. Milano, Italy: Springer, 2012.

[56] C. J. D. Luca, "Surface electromyography: Detection and recording," 2002. [Online]. Available: https://www.delsys.com/downloads/TUTORIAL/semg-detection-and-recording.pdf

[57] D. Farina, R. Merletti, and D. F. Stegeman, *Biophysics of the Generation of EMG Signals*. Hoboken, NJ, USA: Wiley, 2004, ch. 4, pp. 81–105. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/0471678384.ch4

[58] R. Merletti and S. Muceli, "Tutorial. surface EMG detection in space and time: Best practices," *J. Electromyogr. Kinesiol.*, vol. 49, 2019, Art. no. 102363. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1050641119302536

[59] A. Inselberg, "The plane with parallel coordinates," *Vis. Comput.*, vol. 1, pp. 69–91, 1985.

[60] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, 2015. [Online]. Available: https://doi.org/10.1038/nature14539

[61] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017. Accessed: Sep. 22, 2022. [Online]. Available: https://arxiv.org/abs/1412.6980v9

[62] Texas Instruments Inc. INA240, 2016. Accessed: Sep. 22, 2022. [Online]. Available: https://www.ti.com/product/INA240?keyMatch=INA240&tisearch=search-everything&usecase=GPN

[63] Rigol Technologies EU GMBH. Digital Oscilloscope MSO5104, 2020. Accessed: Sep. 22, 2022. [Online]. Available: https://rigolshop.eu/product-oscilloscope-mso5000-mso5104.html

[64] D. M. W. Powers, "Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation," 2020. [Online]. Available: https://arxiv.org/abs/2010.16061

[65] B. Gao, C. Wei, H. Ma, S. Yang, X. Ma, and S. Zhang, "Real-time evaluation of the signal processing of sEMG used in limb exoskeleton rehabilitation system," *Appl. Bionics Biomech.*, vol. 2018, Oct. 2018, Art. no. 1391032.

Andrea Mongardi (Member, IEEE) received the bachelor's degree in electronic engineering and the master's degree in electronic engineering, embedded systems specialization, from Politecnico di Torino, Turin, Italy, in 2017 and 2019, respectively. Since November 2020 he has been working toward the Ph.D. degree with the Politecnico di Torino, Department of Electronics and Telecommunications, focusing on the design of low power embedded systems for sEMG acquisition. He is a Member of the Micro and Nano Electronic Systems (MiNES) Laboratory.

Fabio Rossi (Member, IEEE) received the M.Sc. degree in biomedical engineering and the Ph.D. degree in electronic engineering from the Politecnico di Torino, Turin, Italy, in 2017 and 2022, respectively. Since 2016, he has been a Research Member of the Micro and Nano Electronic System (MiNES) Group, Department of Electronics and Telecommunications (DET) of the Politecnico di Torino. His research interests include the design of low-power sEMG acquisition and processing networks for healthcare monitoring systems, rehabilitative applications, and human-machine interfaces.

Andrea Prestia (Graduate Student Member, IEEE) received the bachelor's and master's degrees in biomedical engineering from the Politecnico di Torino, Turin, Italy, in 2018 and 2020, respectively. Since November 2021, he has been working toward the Ph.D. degree with the Politecnico di Torino, Department of Electronics and Telecommunications. He is a Member of the Micro and Nano Electronic System (MiNES) Research Group. His research interests include the design and development of electronic systems for biomedical applications, mainly focusing on the rehabilitation field.

Paolo Motto Ros (Member, IEEE) received the M.Sc. and Ph.D. degrees in electronic engineering from the Politecnico di Torino, Turin, Italy, in 2005 and 2009, respectively. He is currently an Assistant Professor with the Politecnico di Torino, Department of Electronics and Telecommunications. He was Postdoc Researcher at Politecnico di Torino during 2009–2012, Postdoc Researcher at the Istituto Italiano di Tecnologia during 2012–2019 (Senior since 2014), and Senior Postdoc Researcher at Politecnico di Torino during 2019–2022, where he was also Adjunct Professor since 2017. He is an author and co-author of more than 80 international scientific publications. Dr. Motto Ros was a Member of the organizing committee of IEEE ICECS 2019 and the FoodCAS Satellite Event at IEEE ISCAS 2021, Review Committee Member of IEEE BioCAS 2021 and 2022, special session organizer at IEEE MeMeA 2021, Program Committee Member of IEEE LASCAS 2022, Guest Editor of MDPI *Sensors* and Guest Associate Editor of *Frontiers in Neurorobotics*. He is currently Guest Associate Editor of *Frontiers in Bioengineering and Biotechnology* and Associate Editor of IEEE TRANSACTIONS ON AGRIFOOD ELECTRONICS.

Massimo Ruo Roch (Member, IEEE) received the M.S. and Ph.D. degrees in electrical engineering from the Politecnico di Torino, Turin, Italy, in 1989 and 1993, respectively. He is currently an Assistant Professor from the VLSI-Lab Group, Politecnico di Torino. He is co-author of more than 60 papers. His research interests include VLSI design and implementation of architectures for digital signal processing, communications, embedded systems design, smart systems with distributed intelligence, logic in memory architectures, and cad tools for nanotechnology based architectures.

Maurizio Martina (Senior Member, IEEE) received the M.S. and Ph.D. degrees in electrical engineering from the Politecnico di Torino, Turin, Italy, in 2000 and 2004, respectively. He is currently a Full Professor with the VLSI-Lab Group, Politecnico di Torino. He edited one book and authored or coauthored three book chapters on VLSI architectures and digital circuits for video coding, wireless communications, and error correcting codes. He has more than 100 scientific publications and is coauthor of two patents. His research interests include VLSI design and implementation of architectures for digital signal processing, video coding, communications, artificial intelligence, machine learning, and event-based processing. Prof. Martina is currently an Associate Editor for IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS–I: REGULAR PAPERS. He had been part of the Organizing and Technical Committee of several international conferences, including BioCAS 2017, ICECS 2019, and AICAS 2020. He is also the Counselor of the IEEE Student Branch at the Politecnico di Torino and a Professional Member of IEEE HKN.

Danilo Demarchi (Senior Member, IEEE) received the Engineering and Ph.D. degrees in electronics engineering from the Politecnico di Torino, Turin, Italy, in 1991 and 1995, respectively. He is currently a Full Professor with the Politecnico di Torino, Department of Electronics and Telecommunications. He was a Visiting Professor with Tel Aviv University, Tel Aviv, Israel, during 2018–2021, and at EPFL Lausanne in 2019. He was a Visiting Scientist with the Massachusetts Institute of Technology and Harvard Medical School in 2018. He is author and co-author of five patents and more than 300 international scientific publications. Prof. Demarchi is the Leader of the MiNES (Micro&Nano Electronic Systems) Laboratory with the Politecnico di Torino, a Member of IEEE Sensors Council, a Member of the IEEE BioCAS Technical Committee, an Associate Editor for IEEE SENSORS JOURNAL and IEEE OPEN JOURNAL OF ENGINEERING IN MEDICINE AND BIOLOGY, and of the *Springer Journal BioNanoScience*, the General Chair of IEEE BioCAS 2017 and founder of the IEEE FoodCAS Workshop (Circuits and Systems for Better Quality Food).