A Low Power Wearable Stand-Alone Tongue Drive System for People with Severe Disabilities

Ali Jafari, *Student Member, IEEE,* Nathanael Buswell, Maysam Ghovanloo, *Senior Member, IEEE,* and Tinoosh Mohsenin, *Member, IEEE*

***Abstract*—This paper presents a low power stand-alone Tongue Drive System (sTDS) used for individuals with severe disabilities to potentially control their environment such as computer, smartphone and wheelchair using their voluntary tongue movements. A low power local processor is proposed which can perform signal processing to convert raw magnetic sensor signals to user-defined commands, on the sTDS wearable headset, rather than sending all raw data out to a PC or smartphone. The proposed sTDS significantly reduces the transmitter power consumption and subsequently increases the battery life. The proposed local processor reduces the data volume that needs to be wirelessly transmitted by a factor of 3300, from 9.6 kbit/s to 3 bit/s. The proposed processor consists of three main blocks: Serial Peripheral Interface bus for receiving raw data from magnetic sensors, external magnetic interference attenuation to attenuate external magnetic field from the raw magnetic signal, and a machine learning classifier for command detection. A proof of concept prototype sTDS has been implemented with a low power IGLOO-nano FPGA, Bluetooth Low Energy, battery and magnetic sensors on a headset and tested. At clock frequency of 20 MHz, the processor takes 6.6 μs and consumes 27 nJ for detecting a command with a detection accuracy of 96.9%. To further reduce power consumption, an ASIC processor for the sTDS is implemented at the post layout level in 65 nm CMOS technology with 1 V power supply, and it consumes 0.43 mW which is 10$\times$ lower than FPGA power consumption and occupies an area of only 0.016 mm^2.**

***Index Terms*—Tongue Drive System, machine learning classifier, energy efficiency, FPGA, assistive technology, ASIC, paralysis.**

I. Introduction

ASSISTIVE Technologies (ATs) help people with severe disabilities to perform many daily activities with minimal or no assistance and increase their independence [1]–[16]. These systems allow the users to send commands to an external device, such as a motorized chair, smart phone or computer. In order to allow those with severe disabilities to interact with the environment around them, different ATs have been developed that use different sensor modalities such as electroencephalogram (EEG) [17] and electrooculogram (EOG) [18], eye movements [19], head motion [20], and facial muscle activity [21].

This research is based upon work partially supported by the National Science Foundation under Grant No. 00010145.

A. Jafari, T. Mohsenin and N. Buswell are with Computer Science and Electrical Engineering department, at University of Maryland Baltimore County, Baltimore, MD, USA (email: a137@umbc.edu, tinoosh@umbc.edu).

M. Ghovanloo is with the School of Electrical and Computer Engineering, at Georgia Institute of Technology, Atlanta, GA, 30332 USA (e-mail: mgh@gatech.edu)

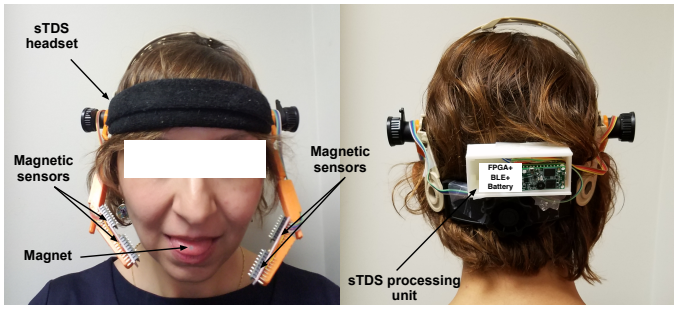

Fig. 1. sTDS prototype placed on a headset which includes a low power FPGA, four magnetic sensors, a Bluetooth low energy transceiver, a battery and a magnetic tracer, which is glued to the user's tongue.

Paralyzed individuals with severe physical disabilities, such as those with spinal cord injury (SCI), traumatic brain injuries (TBI), and some type of stroke, who suffer from tetraplegia, a condition in which all four limbs are paralyzed, heavily rely on ATs to enhance their quality of life and to live more productively and independently [22].

Because of following reasons, the tongue is an ideal source of volitional commands for developing a wearable AT system for people with severe paralysis [23]: 1) Sophisticated motor control capability evident in speech and ingestion, 2) Fast movement with many degrees of freedom and very flexible, 3) It is connected to the brain by a cranial nerve: it escapes even high level spinal cord injuries, 4) Noninvasive access to tongue is possible, 5) it is not afflicted by repetitive motion disorders, 6) it does not fatigue easily [24], 7) It is all inside the mouth and it has privacy advantage, 8) it is not influenced by the position of the rest of the body [25].

Several tongue-computer interfaces have been proposed in recent years [26]–[31]. Moreover, we have demonstrated the effectiveness of the TDS previously [32]–[38]. Typically, these systems capture signals from analog sensors, digitize them after signal conditioning and send all the raw data through a wireless transmitter to a receiver platform for further processing, such as feature extraction and classification. The receiver platform can be a computer or smart phone. In the case of TDS, assuming Analog Front End (AFE) takes at least 50 samples per second [33] for each of the four sensors and each sample has X, Y, and Z data which is 16 bits each, the transmitter needs to send 9.6 kbit/s to the receiver side. Therefore, the drawback of this type of system is that constantly transmitting this amount of raw data using a transmitter such as Bluetooth Low Energy (BLE) results in high power consumption [39].

In this paper, a new stand-alone TDS (sTDS) shown in Fig. 1 is proposed which performs the entire signal processing at the sensor node and only sends the final decision through a BLE to a smart phone or a personal computer. In the proposed system, the BLE doesn't need to be active all the time and it can be in advertisement mode which consumes very low power. Fig. 2 shows a comparison between the power consumption of TDS when sending all raw data out versus preforming the processing locally and sending only 3 bits of decision. As it can be seen from the figure, the power consumption of sTDS is significantly lower than TDS. TDS consumes around 21.6 mW power for receiving and sending 9.6 kbit/s raw sensor data, whereas sTDS consumes 8.8 mW power for receiving, processing and issuing detected commands.

Also, the proposed system is not dependent on a receiver platform to run a software for processing, such as MATLAB and LabVIEW. Therefore, a user does not need to have another device other than a headset which makes the user more independent.

Fig. 1 shows the sTDS prototype which includes a local processor, four magnetic sensors, a BLE transceiver, a battery and a magnetic tracer which is glued to the user's tongue. Two magnetic sensors are placed on each side of the headset and the processor is placed on a box at backside of the headset. Also, the box is used for placing a battery. The box is designed using 3D printing technology and the weight of the box is around 0.14 lb. This paper makes the following major contributions:

- Design a hardware-efficient and low power processor which performs all signal processing locally at the sensor side and sends out only final decision (three bits), rather than sending all raw data.
- Compare three different machine learning classifiers in terms of detection accuracy, number of computations and memory requirement.
- Perform fixed-point analysis to find out the best number of bit representation in each stage of the local processor design.
- Design a low-power, small-size and reconfigurable classification processor using Multinomial Logistic Regression (LR), which consumes less energy and occupies less area in comparison with machine learning algorithms implemented in previous works.
- Develop, implement and test a fully working sTDS on a headset, which integrates four magnetometers, a low-power FPGA for processing, a BLE, a battery and a smartphone/PC.
- Implement an ASIC sTDS processor through full place and route flow in 65-nm CMOS technology and provide analysis in terms of performance, resource utilization and power consumption.
- Design a Maze navigation game and integrate it with sTDS to test the functionality of the system in real-time.

The remainder of this paper is organized as follows. Some related works are provided in section II. Complete system design including all sub-systems and data analysis are presented in Section III. Hardware architectures are shown in section IV. Section V shows FPGA prototyping, ASIC results and comparison with other existing work. Experimental study and performance results are provided in section VI. Finally, section VII presents some concluding remarks.

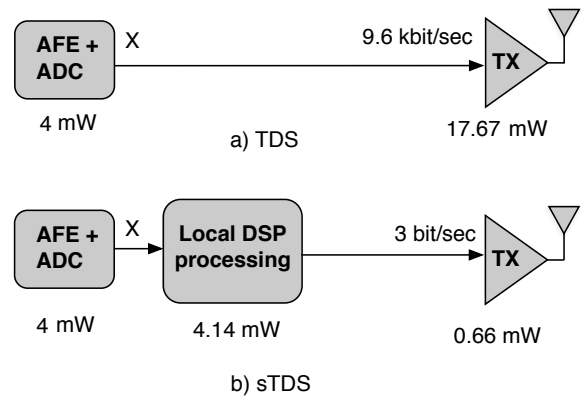

Fig. 2. a) TDS: Sending all raw data out. b) sTDS: performing all processing locally and sending out the command decision. The transmission power is reduced significantly by performing all signal processing locally on the sensor side. Measured power consumption numbers are shown separately for 1) AFE which generates 9.6 kbit/s raw magnetic data, 2) Local DSP processor on a low power FPGA and 3) Bluetooth low energy.

II. Related Work

In recent years, several wearable tongue drive ATs for people with severe disabilities have been developed which are discussed in this section. In [40] authors proposed a TDS for controlling a mouse. In their work, External Magnetic Interference (EMI) effects have been reduced to an acceptable level by adding a reference 3-D compass. Principal component analysis (PCA) and k-Nearest Neighbor (KNN) algorithms are used to associate the magnetic field sensor outputs to 6 different direct mouse commands. In [36], [37] authors evaluated a tongue operated AT as a switch-based pointing device with four directional commands for computer access and achieved an accuracy of 94.7%. Zhang et al. [33] presented a new rehabilitation robot, called Hand Mentor (HM) Pro^{TM}, which reads its pressure and joint angle sensors, combined with control commands from the TDS to enable both isometric and isotonic target-tracking tasks in a coordinated tongue-hand rehabilitation paradigm. In [38], [41] authors introduced a multi-modal version of TDS, which takes different sensor modalities such as tongue motion, speech and head tracking. They used the system for controlling a mouse, typing and sending an email. [38] reported an overall 85.1% $\pm$ 8.8% recognition accuracy. In all these works, raw data needs to be transmitted wirelessly to a receiver platform for further processing, which results in high power consumption. Also, they depend on the receiver platform to run a software for signal processing, such as MATLAB and LabVIEW. In [31], authors presented an optical tongue drive system with an accuracy of 92%. Their proposed prototype is wired, which is not convenient for a paralyzed patient. In their work, each gesture takes 1.5 seconds on average to perform and recognize which is a high latency for real-time applications such as wheelchair driving. Authors in [30] present a tongue machine interface based on using Glossokinetic potentials (GKPs) which are

electric potential responses generated by tongue movement. They use the proposed system for controlling a wheelchair. Their system requires the users to carry a scalp on their head in order to record EEG signals and only detects and uses three commands for wheelchair control. In [25], authors proposed a TDS local processor with an accuracy of 93.3% which performs all signal processing on the sensor side and send out only the detected commands. The results were based on Verilog simulations and actual hardware was not built and tested. KNN was used as a machine learning classifier which consumes high energy due to requiring many computations and a large on-chip memory to store the classifier's training data. **Authors in [28], [29] present a wireless, intraoral and inductive tongue computer interface to type using the keypad and mouse pad area. Their proposed system, does not need a receiver platform such as computer/smartphone to run a software for processing.**

Compared to the previous works, the proposed paper presents a stand-alone TDS which performs the entire signal processing at the sensor node which is a light-weight headset and only sends out the final decision (3 bits) through a BLE. The BLE doesn't need to be active at all, as it sends these 3 bits through advertising packet [42]. Therefore, the power consumption due to the wireless transmission is reduced significantly. Furthermore, the proposed system does not depend on a receiver platform to run a software for processing, such as MATLAB and LabVIEW. **Hence, a user does not need to have another device other than a headset which makes the user more independent.** Also, employing the proposed stand-alone system reduces the cost and system failure rate due to having multiple processing units.

It is worth mentioning that, the intraoral [28], [29] and the headset [38], [43] versions of a tongue-operated AT each have their pros and cons. Therefore, the ultimate choice depends on the preference of the end user, i.e. whether they prefer comfort over aesthetics or vice versa. In fact, their preference might even depend on the environment that they are in. For instance, they might choose to use the headset version at home, but wear the intraoral version outside or in social events. Nonetheless, both versions will immensely benefit from conducting the classification on the sensor side.

III. Algorithm Design and analysis

An overview of the proposed system architecture is discussed here. Fig. 3 shows the high-level block diagram of the proposed system which consists of a Serial Peripheral Interface (SPI), an External Magnetic Interference (EMI) attenuation, and a Machine Learning (ML) classifier kernel. As it is shown in Fig. 3, first the raw data generated by 4 magnetometers are transferred into the EMI kernel in the local processor, using SPI. The EMI kernel attenuates the effect of the external magnetic interference on the raw data and generates some features which are passed into the ML classifier module for classification. The ML classifier detects the user-defined command and finally, BLE sends the generated command out. All the kernels are discussed in details in the following subsections.

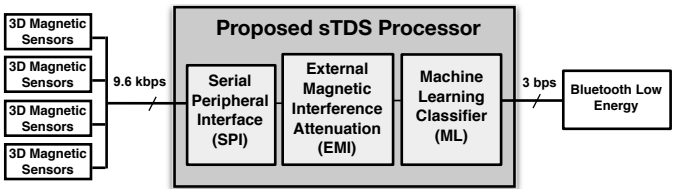

Fig. 3. Block diagram of the proposed sTDS containing SPI interface, External Magnetic Interference cancellation, and Machine learning classifier modules.

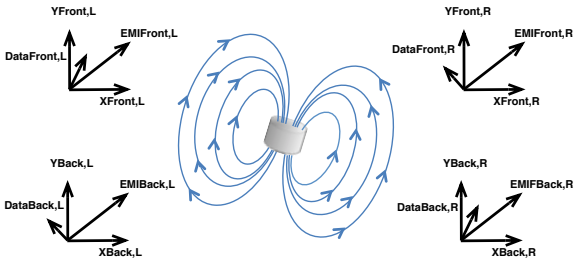

Fig. 4. Relative 3-D position and orientation of the 3-axial sensor modules and the permanent magnetic tracer attached to the user's tongue. External magnetic interference is assumed constant within this environment while the magnetic field generated by the tracer varies between the four sensors.

A. Serial Peripheral Interface (SPI)

The front end of the system is composed of four magnetic sensors (LSM303D), which have onboard 3D compasses that are used to monitor the magnetic field generated by a magnet tracer which is glued to user's tongue [44]. Each sensor is configured to provide a magnetic field full-scale of +/- 8 Gauss. The interface to the AFE uses two SPI buses, one for the left two sensors and one for the right two sensors. Each sensor can be configured to use one of two possible slave addresses, allowing two sensors on the same bus. In order to minimize the device utilization needed for the interface, only one instance of the SPI bus master was instantiated and it is used to read each of the sensors one at a time. Each sensor provides a 16-bit two's complement magnetic reading for each of its three axes. Once all four sensors have been read, the interface then passes the data to the EMI cancellation logic.

B. External Magnetic Interference (EMI) Cancellation

Magnetic sensors are highly sensitive and inevitably affected by external magnetic interference (EMI), such as the Earth's magnetic field (EMF). This results in a poor signal-to-noise ratio (SNR) at the sensor outputs, which degrades the performance of the machine learning algorithm. Therefore, eliminating the EMI is necessary to enhance the TDS performance, and reduce the probability of errors in command interpretation [34]. The methodology behind the EMI cancellation algorithm assumes that the EMI noise is almost equal across each of the four sensors at any moment in time while the magnetic field generated by magnetic tracer will vary from sensor to sensor.

Fig. 4 depicts the relative position and orientation of the 3-axial sensor modules and the magnetic tracer attached to

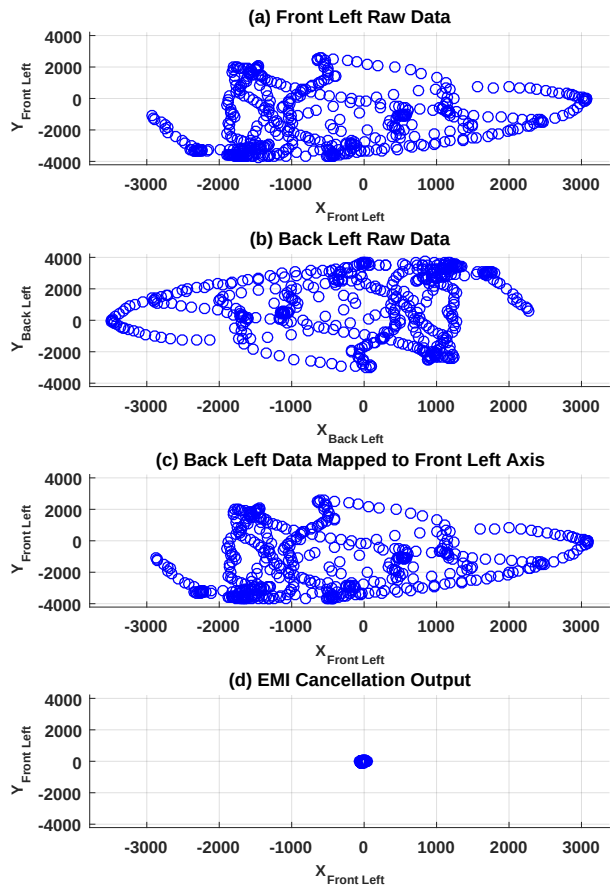

Fig. 5. External Magnetic Interference data before and after applying EMI cancellation. EMI data was collected by removing the magnetic tracer from the system and recording the sensors data. After applying the EMI cancellation algorithm, a cluster at [0,0] is expected.

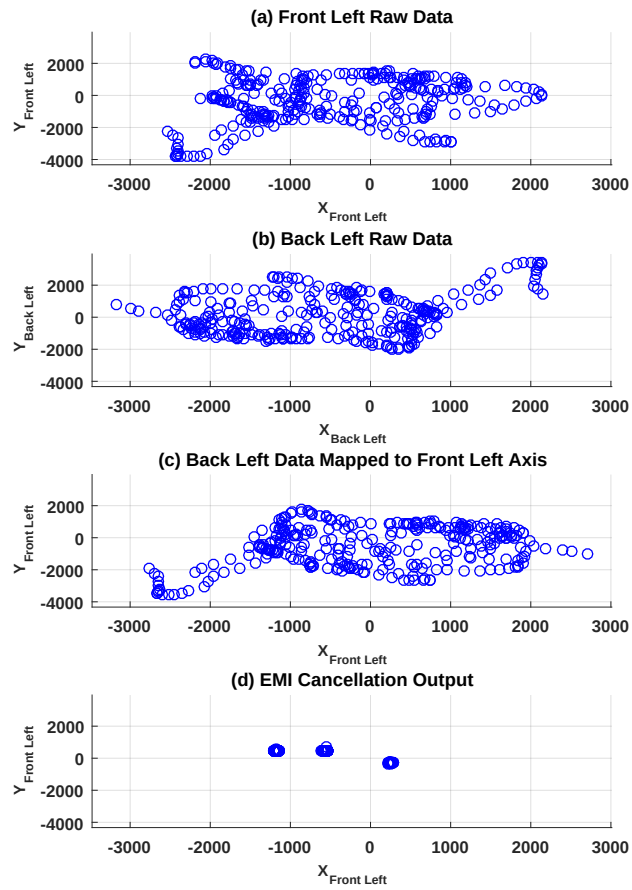

Fig. 6. Magnetic data before and after EMI cancellation. Before applying EMI the data is scattered all over the axis and it is hard to classify, however after performing EMI cancellation, the data are clustered for each command which is very useful for classification purpose.

the user's tongue. The raw data from each sensor includes the user data generated by the tracer and EMI. Since the relative position and orientation of the two modules are fixed, the sensor outputs in response to EMI are linearly related. This enables us to remove the EMI interference by subtracting one sensor measurement from another and then classifying the command using the remaining difference between the two sensor's measurements of the actual user data. Since the sensors provide 3-D measurements, subtracting one measurement from another will provide a 3-D feature for classification. Applying equations (1-5) will eliminate EMI for left-side sensors and they can be repeated for the right-side sensors, producing a total of two 3-D features that can be used for classifying the user's commands. The following abbreviations will be used for the equations (1-7): Raw = R, Data = D, Feature = F, Mapped = M, Front Left = FL, Back Left = BL, Front Right = FR, Back Right = BR, Mapped EMI = MEMI.

$$R_{FL} = D_{FL} + EMI_{FL} \tag{1}$$

$$R_{BL} = D_{BL} + EMI_{BL} \tag{2}$$

$$MR_{BL} = R_{BL} * Mapping_{coefficients} = MD_{BL} + MEMI_{BL} \tag{3}$$

$$EMI_{FL} = MEMI_{BL} = EMI \tag{4}$$

$$F_L = R_{FL} - MD_{BL} = D_{FL} - MD_{BL} \tag{5}$$

Replacing (1-4) into (5) results EMI to be ideally removed from the signal.

Also, in order for (5) to be applied, the data from the back sensor and the front sensor must share a common coordinate system. Since there are process variations that result in the front and back sensors often having different coordinate systems it is necessary to map one sensor data to the other sensor axes. In order for this to be possible, there must be a known relationship between the two sensors axes. In this system the position of the front and back sensors are fixed on the same fixture in order to ensure that the relationship between their two coordinate systems remains linear. Then back data is mapped to the front sensor axes. Since the location of the back sensors axes are fixed with respect to the front sensors axes, a set of linear equations can be found that will map the back sensors axes to the front sensors axes, as shown in equations (6) and (7).

In order to generate the linear equations shown in (6) and (7), first it is needed to record a set of data where there is known to be a linear relationship between the back sensor's measurements and the front sensors's measurements. In order to generate this data set, the magnetic tracer is removed from the system and only the remaining EMI is recorded, which is assumed to be equal across the four sensors. An example of this data is shown in Fig. 5-a and 5-b. This data was input to a linear regression function in order to generate the a, b, c, and d coefficients from (6) and (7), which will map the back senor's data to the front sensor's coordinate system, as illustrated in Fig. 5.c. Fig. 5.d shows the effect of applying the EMI cancellation algorithm to the raw EMI data set, where the EMI noise is cancelled out and the data cluster around the center of the coordinate system is remained.

$$\begin{cases} X_{MBL} = a_{x,L}X_{BL} + b_{x,L}Y_{BL} + c_{x,L}Z_{BL} + d_{x,L} \\ Y_{MBL} = a_{y,L}X_{BL} + b_{y,L}Y_{BL} + c_{y,L}Z_{BL} + d_{y,L} \\ Z_{MBL} = a_{z,L}X_{BL} + b_{z,L}Y_{BL} + c_{z,L}Z_{BL} + d_{z,L} \end{cases} \tag{6}$$

$$\begin{cases} X_{MBR} = a_{x,R}X_{BR} + b_{x,R}Y_{BR} + c_{x,R}Z_{BR} + d_{x,R} \\ Y_{MBR} = a_{y,R}X_{BR} + b_{y,R}Y_{BR} + c_{y,R}Z_{BR} + d_{y,R} \\ Z_{MBR} = a_{z,R}X_{BR} + b_{z,R}Y_{BR} + c_{z,R}Z_{BR} + d_{z,R} \end{cases} \tag{7}$$

Fig. 6 shows the effect of the EMI cancellation algorithm with the magnetic tracer present. The first two plots show the raw data recorded by the left two sensors. The raw data used is generated by recording three distinct commands while moving the test fixture, so as to subject the system to a range of EMI noise values. The Fig. 6.c plot shows the effect of mapping the back senor's data to the front sensors axes. Fig. 6.d shows the output data that will be sent to the ML classifier, which has now been separated into three distinct clusters representing the three original commands.

C. Machine Learning (ML) Classifier

To evaluate the proposed sTDS accuracy, different machine learning classifiers such as K-Nearest Neighbor (KNN), Support Vector Machine (SVM) and Logistic Regression (LR) have been employed and evaluated with respect to their complexity, power consumption and accuracy [45], [46]. These algorithms are briefly discussed in following subsections.

1) KNN: k-Nearest Neighbor (KNN) algorithm classifies test data by a majority vote of the known labels of the KNNs using some distance metric such as Euclidean or Manhattan distance. Training consists of simply storing all the labeled training data. Classification requires computing the distances between a test instance and all training instances, while tracking the k smallest distances.

2) SVM: Support vector machines work by finding the maximum margin hyperplane, i.e., the linear separator that is as far as possible from the closest positive and negative training instances. Kernel functions can be used to project the data into a high dimensional space such that the linear separator is highly non-linear in the input space. Samples are then classified using the function shown in equation (8), where $\vec{s_i}$ is a support vector, $\vec{x}$ is a test vector, $K(\vec{s_i}, \vec{x})$ is the kernel function, α_i and γ_i are the weight and label of the support vector, and b is the bias.

TABLE I
HARDWARE COMPLEXITY ANALYSIS OF THE MACHINE LEARNING ALGORITHMS WHEN TRAINED OFF-LINE

Algorithm	Multiplication	Addition + Subtraction	Memory Requirement (bit)
K-Nearest Neighbor	$F \times N$	$(F-1) \times N + (F \times N)$	$N \times F \times b$
Support Vector Machine	$F \times C \times S$	$(F-1) \times C \times S$	$F \times C \times S \times b$
Logistic Regression	$(F+1) \times (C-1)$	$F \times (C-1)$	$(F+1) \times (C-1) \times b$

N is size of training data, F is number of features, S is number of support vectors, C is number of commands and b is bit resolution. For the proposed system, $N \gg S \gg F$.

$$f(x) = sign\left(\sum_{i=1}^{NUM\_SV} \alpha_i \gamma_i K(\vec{s_i}, \vec{x}) + b\right) \tag{8}$$

3) LR: Logistic Regression is a discriminative probabilistic model that utilizes the logistic function to map real-valued inputs between 0 and 1 which can be interpreted as probabilities. Training is an iterative process that finds maximum likelihood regression coefficients (weight vectors). Samples are typically classified by selecting the label with the greater probability between $P(Y = K|X)$ and $P(Y = 0|X)$ given in equation (9), where $w_{k,i}$ are the weights for each of the N features for the label, k, and x is the input vector:

$$P(Y = k|X) = \frac{1}{1 + exp\left(w_{k,0} + \sum_{i=1}^{N} w_{k,i}x_i\right)} \tag{9}$$

Also, probability ratios are used which significantly reduce computational complexity as shown in equation (10).

$$\begin{aligned} \frac{P(Y = 1 : K-1|X)}{P(Y = K|X)} &\Rightarrow exp\left(w_{0,1:K} + \sum_{i=1}^{N} w_{i,1:k}x_i\right) \\ &\Rightarrow \sum_{i=0}^{N} w_{i,1:k}x_i \end{aligned} \tag{10}$$

D. Dataset and performance measurement

Several different data sets are captured using sTDS for training and testing purpose. To estimate sTDS accuracy, a leave-one-record-out cross-validation approach is used [39], [47]. Let N denote the number of data set records. To estimate the sTDS accuracy, it is trained on $N-1$ records from four different users with N=10 records (Total 40 different records). The sTDS is then tasked with detecting the commands of the withheld test record. This process is repeated N times so that each record is tested. For each round, the detected commands are recorded and accuracy is measured by calculating how many times the sTDS predicted a wrong label for a window of input data generated by the magnetic sensors. Then, average accuracy over all rounds is reported.

Fig. 7 shows a comparison among all ML algorithms. The detection accuracy for KNN using Euclidean distance calculation, SVM with linear kernels and LR is 97.6, 96.7 and 96.9 respectively. Hence, the accuracy for all algorithms are similar and satisfactory for Human-computer Interface applications such as mouse and keyboard control or playing

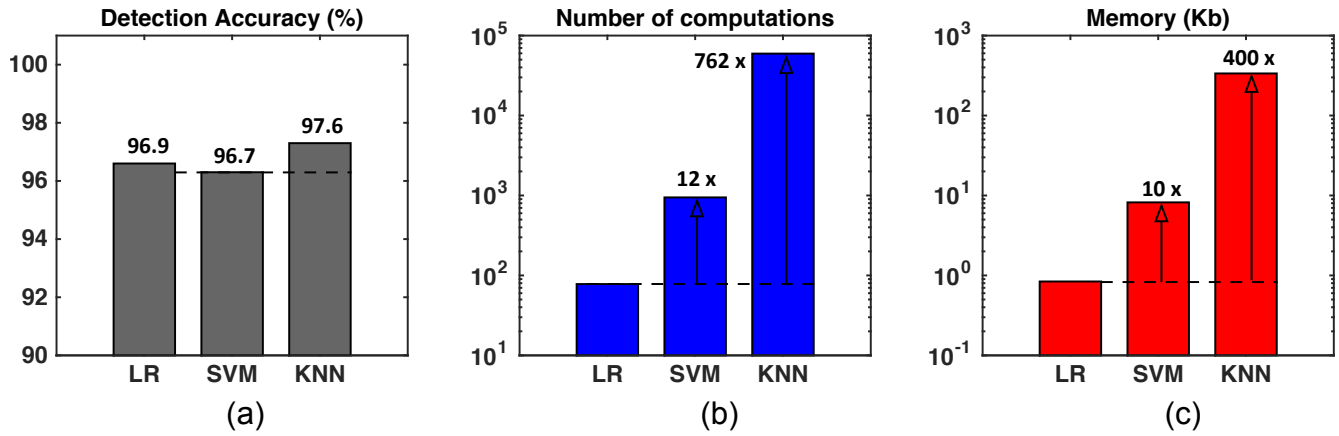

Fig. 7. Machine Learning classifiers comparison in terms of (a) Detection Accuracy, (b) Number of computations and (c) Memory requirements. Although the detection accuracy is almost similar among all algorithms, LR needs 762× and 12× less computation and also requires 400× and 10× less memory (for saving the training model parameters) compared to KNN and SVM algorithms.

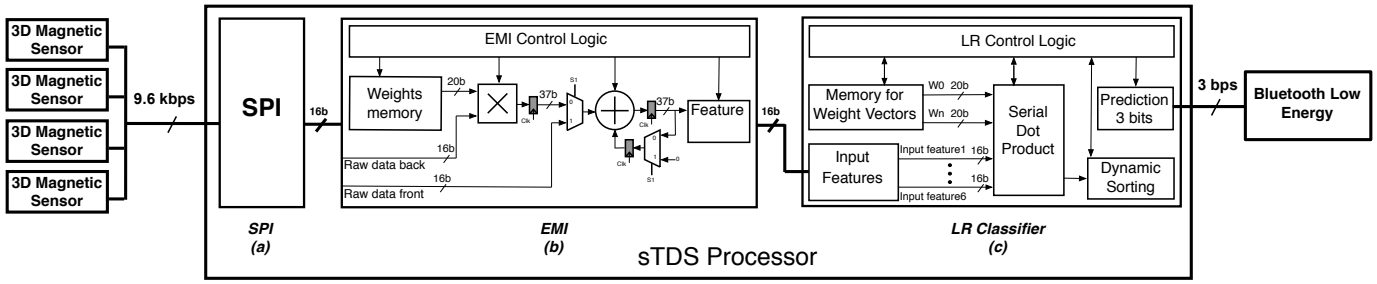

Fig. 8. Hardware architecture used in the sTDS processor. Part a is the SPI module and it is not shown in detail here as it is an off-the-shelf module. Part b shows implementation of EMI kernel which includes memory block, multiplier, adder/subtractor, a few multiplexer and state machine. Also, part c depicts LR implementation which contains serial dot product engine, a memory block, state machine and a dynamic sorting module.

a computer game. The number of computations and memory requirements are calculated based on Table I, which shows the hardware complexity of the ML algorithms when training is performed off-line. Using 3500 training data samples, LR needs 78 computations to detect one command which is 762× and 12× smaller than the number of computation of KNN and SVM. The number of computations directly effects on energy consumption. Also, LR requires 0.8 kb memory for saving training model parameters where as KNN and SVM need 336 kb and 8.2 kb respectively which are approximately 400× and 10× higher than LR. Even if the number of training data samples increases, the number of computations and memory requirement are fixed for the LR; however, they increase for the SVM and KNN algorithms. This analysis results show that LR is the best candidate for the proposed sTDS because not only it could achieve similar accuracy compared to other algorithms, but also it needs smaller number of computations and smaller memory for saving the training model parameters. Hence, in this paper LR is chosen as the ML classifier and implemented on hardware.

IV. Hardware Architecture

Implementing hardware architecture for EMI and ML algorithms faces several challenges such as pre-processing of features, computational model implementation and managing memory transfers. Fig. 8 depicts sTDS hardware architecture with implementation details. Fig. 8-a shows SPI module. As it can be seen from Fig. 8-b, EMI block contains one multiplier, one adder/subtractor, one block ROM for saving EMI calibration data, 2 multiplexers, a few registers, and a state machine block. As it was explained in III-A, the input data coming from SPI is 16-bit two's complement. Also, the EMI calibration data are represented by 20 bits. After performing EMI cancellation, the data is truncated to 16 bits and saved in feature memory. A fixed-point analysis is performed to find out the best number of bit representation in each stage of the hardware architecture design.

Based on the results provided in III-D, LR is chosen to be implemented as the ML classifier. Usually, LR is trained offline [39]. An offline training is performed to obtain weight vector using MATLAB. The weight vectors are converted to fixed-point format and are represented by 20 bits. The floating-point arithmetic is complex and requires more area, therefore use of fixed point arithmetic will avoid complex multipliers. The input to LR is feature data, which is formed by EMI block. The LR architecture consists of four main blocks: a block ROM for saving weight vectors, a serial dot product engine with bias vector addition, a dynamic sorting module to

find maximum values and related indexes, and a state machine for controlling all sub-modules, as shown in Fig. 8-c.

V. Hardware Implementation Results

In this section, different sTDS implementations on both FPGA and ASIC and their results are discussed. Also, a comparison with a previous work is provided.

A. FPGA Implementation Results

The complete proposed sTDS solution which includes SPI, EMI attenuation kernel, and LR machine learning classifier, is implemented on a Microsemi IGLOO-nano FPGA (AGLN250) at clock frequency of 20 MHz and fully tested on the headset which was shown in Fig. 1.

TABLE II
Power consumption breakdown of the proposed sTDS on IGLOO-nano FPGA (AGLN250)

Type	Power consumption (mW)	
	Typical	Best
Network dynamic	3.6	3.25
Gate dynamic	0.33	0.298
I/O dynamic	0.078	0.07
Core static	0.022	0.021
Bank static	0.006	0.006
Memory	0.443	0.4
Total static	0.028	0.026
Total dynamic	4.451	4.017
Total	**4.479**	**4.043**

The results are obtained by Libero tool with operating frequency of 20 MHz and V_{core}= 1.2 V.

Table II shows power consumption breakdown of post-place and route implementation on the FPGA, which is obtained by using Libero XPower tool analysis. As it can be seen from the table, the static power consumption of IGLOO-nano FPGA is around 30 μW which is very small compared to other type of FPGAs. Thus, this type of FPGA is suitable for battery-powered wearable ATs. Based on the results shown in Table II, most of power is consumed for dynamic switching activity in FPGA. Total FPGA power consumption is less than 4.5 mW. Also, it take 6.6 μs for the processor to finish the computations to detect one command.

Fig. 9 shows the device utilization breakdown of the proposed sTDS on the IGLOO-nano FPGA. The pie chart shows that the LR module takes 51%, EMI module 32%, SPI 12% and memory 5% of the whole design.

B. On-board Power Measurement Results and Analysis

Fig. 10 shows experimental setup for power measurement of the FPGA. Power consumption is measured by collecting the current consumed at FPGA core. The current consumption of the FPGA was captured using a Current/Power monitor IC (TI INA219) and sampled by an Arduino Uno as shown in Fig. 10.

To have a fair comparison between the power consumption of the traditional TDS and proposed sTDS, both systems are implemented on hardware. Table III shows a current consumption breakdown of the traditional TDS and proposed

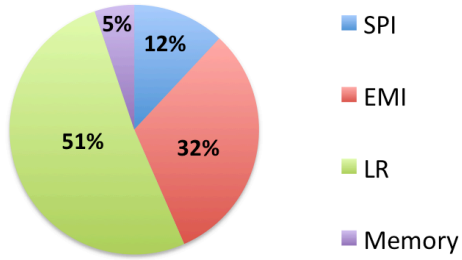

Fig. 9. sTDS Device Utilization breakdown on IGLOO-nano FPGA (AGLN250).

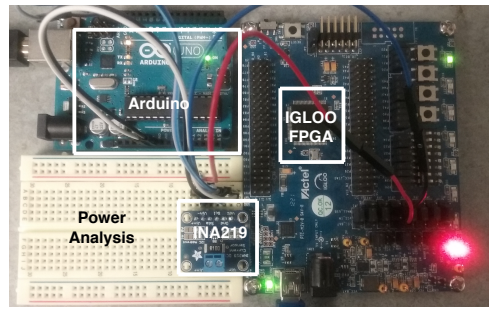

Fig. 10. Experimental setup for power measurement of FPGA core including, INA219 power monitor sensor, Arduino Uno microcontroller and IGlOO-nano FPGA boards. Special care was given to measure exactly the power consumption of the FPGA core on the board.

sTDS. As it can be seen from the table, both systems consume around 1.2 mA current on four magnetic sensors. Current consumption of TDS on FPGA is around 2.6 mA whereas sTDS consumes 3.45 mA. TDS FPGA current (without BLE) is smaller because it does not include EMF and LR. However, current consumption of BLE on TDS is 4.41 mA and for sTDS is 0.2 which is 22 time smaller than TDS. As it was mentioned previously, BLE in sTDS is on advertisement mode all the time. The 3 bit detection commands are sent during the advertisement through the advertisement content. However, in TDS the BLE should be in active mode [48] to send all raw data. Total current consumption of TDS and sTDS including all modules is 8.21 mA and 4.85 mA, respectively. The results show sTDS could achieve approximately 60 % power saving. This leads to a significant energy saving around 23.6 hours (approximately one day) in battery life, for a 400 mAh battery, as it is shown in Fig. 11. Battery life is calculated based on (11). The factor of 0.7 makes allowances for external factors which can affect battery life.

$$Battery\ life = \frac{Battery\ Capacity(mAh)}{Current\ Consumption(mA)} * 0.7 \qquad (11)$$

C. Post-Layout ASIC Simulation Results

To reduce the overall power consumption, a standard-cell register-transfer level (RTL) to Graphic Data System (GDSII) flow using synthesis and automatic place and route is used. The proposed sTDS processor including SPI, EMI attenuation kernel, and Logistic Regression machine learning classifier

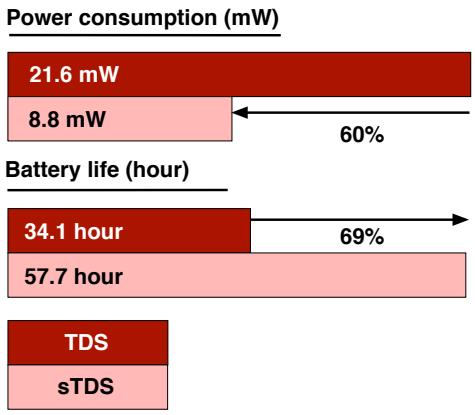

Fig. 11. Measured power consumption and battery life comparison between sending all raw data out and performing all processing locally on IGLOO-nano FPGA, using a 400 mAh battery.

TABLE III
ON-BOARD CURRENT MEASUREMENT RESULTS BREAKDOWN OF DIFFERENT COMPONENTS OF THE TDS AND PROPOSED STDS.

Design	TDS (mA)	sTDS (mA)
4 Magnetic Sensors	1.2	1.2
FPGA	2.6	3.45
BLE	4.41 (active mode)	0.2 (advertisement mode)
Total Current	8.21	4.85
Battery life (hour)	34.1	57.7

is implemented using Verilog to describe the architecture, synthesized with Synopsys Design Compiler, and place and routed using Cadence SOC Encounter. The results are provided in table IV. The processor is able to operate at 800 MHz clock frequency. However, the clock frequency has been reduced to 20 MHz to reduce the power consumption. The ASIC implementation reduces the processor power consumption by a factor of 10. Fig. 12 shows the layout of the sTDS local processor in a 65-nm TSMC standard CMOS technology. It occupies an area of 0.016 mm^2. White rectangles indicate the SPI, EMI and LR modules and memory for saving the calibration coefficients.

D. Comparison with existing work

Table V compares the performance of the proposed work and our previously published work [25]. As it can be seen from the table, we have achieved similar detection accuracy and ASIC implementation area. However, the proposed work can achieve significantly better performance in term of energy consumption which is the most important factor for the applications that are battery-powered, such as sTDS. At 10 MHz frequency, the energy consumption of the proposed sTDS on ASIC and FPGA is 3.9 nJ and 29.7 nJ, respectively and compared to [25] it could achieve 41× and 79× energy saving. Furthermore, sTDS outperforms the earlier implementation in term of memory requirement. The LR classifier designed in the proposed work requires 0.8 kb memory for saving the training data where as KNN which was used in [25] needs 15.2 kb memory which is approximately 19× more memory than LR.

TABLE IV
STDS PROCESSOR ASIC IMPLEMENTATION RESULTS AT OPERATING FREQUENCY OF 20 MHZ.

Post-Layout ASIC Simulation Results	
CMOS fabrication process	65 nm
Power Supply (V)	1
Post-Place & route area utilization (%)	95
Memory for train data (kb)	0.8
Nominal Operating Frequency (MHz)	20
Maximum clock Frequency (MHz)	800
Core area (mm^2)	0.016
Latency to detect one command (μSec)	6.6
Leakage Power Consumption (mW)	0.151
Dynamic Power Consumption (mW)	0.288
Total Power Consumption (mW)	**0.439**
Energy Consumption/command (nJ)	**2.89**

Total power consumption does not consider the sensor and wireless power consumption and includes only the ASIC processor.

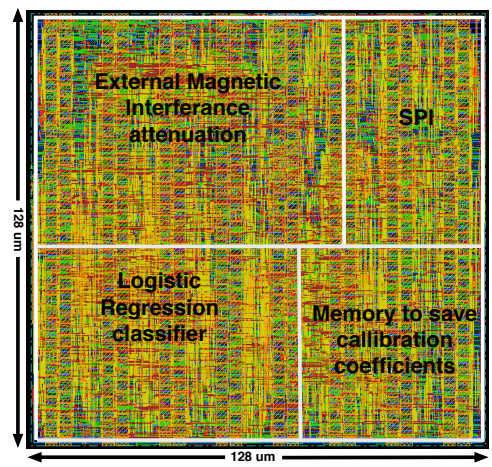

Fig. 12. sTDS processor post-layout ASIC implementation in 65-nm TSMC CMOS technology.

VI. EXPERIMENTAL STUDY AND PERFORMANCE RESULTS

The sTDS experiment is performed in two phases, training and testing. As it was explained in section IV, the training phase is performed offline using MATLAB and the classification model parameters are saved on hardware for the testing phase. EMI calibration data are recorded for calculating mapping coefficients for EMI algorithm, by moving the headset around without moving tongue. Also, training data is recorded and labeled to train the multi-class LR algorithm. A user can choose 6 specific teeth to label the train data. In this work, the teeth shown in Fig. 13 are chosen for labeling the train data, which will be interpreted as start, stop, left, right, up, down commands. Furthermore, if the user doesn't place the tongue on any tooth, sTDS will interpret that as neutral command.

To validate functionality of sTDS in the testing phase, a computer-based Maze navigation game is designed and tested. Fig. 13 presents a GUI which shows the Maze game, elapsed time of the game, command positions on mouth [38] and different buttons which are controlled by sTDS. A user should navigate the Maze using the commands generated by their tongue. The goal is to move from the start to the end point,

TABLE V
COMPARISON OF THE PROPOSED sTDS WITH AN EXISTING WORK, ON BOTH ASIC AND FPGA IMPLEMENTATIONS RESULTS.

Design	[25]	This work	Improvement
CMOS fabrication process	65 nm, 1 V	65 nm, 1 V	–
ML classifier	KNN	LR	–
Detection accuracy (%)	93.3	96.9	1
Computations/ command	12500	132	95×
Latency to detect one command (ms)	x	x	xx×
Memory for train data (kb)	15.2	0.8	19×
Operating Frequency (MHz)	10	10	–
ASIC Area (mm^2)	0.02	0.016	1.25×
Total ASIC energy/ command (nJ)	160	3.9	41×
Total FPGA energy/ command (nJ)	2335	29.7	79×

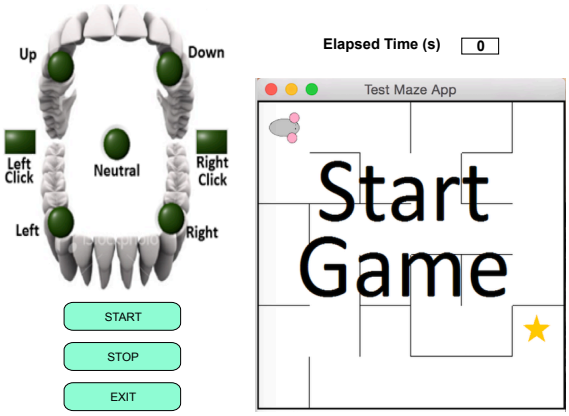

Fig. 13. This figure shows a Maze navigation game with a start and end location. During the experiment a small mouse automatically moves to start location. User starts to move mouse using 4 different tongue commands (left, right, up and down) to navigate from start to end.

which is a star. In order to generate the commands, the user should move his/her tongue to the specific teeth which they used previously in the training phase. The goal of the experiment is to finish the game as fast as possible from start to end using the sTDS. Four users (age: 26-37, 3 male and 1 female, experienced-familiar) played the Maze game five times. The results showed that all users finished each round of the game in less than two minutes.

VII. CONCLUSION

This paper presents a low power sTDS for people with severe disabilities, who have difficulties in accessing their environment and other activities in their daily life. A low power processor is proposed, which can perform all signal processing locally on the wireless headset and sends out only a command to the target device, which can be a PC or smartphone. This approach significantly reduces the transmission power consumption by reducing the transmission rate from 9.6 kbit/s to 3 bit/s. The proposed sTDS is fully implemented on a headset including commercially-available IGLOO-nano FPGA, four magnetic sensors, a BLE and a 400 mAh battery. By employing the sTDS, power consumption has been reduced from 21.6 mW to 8.8 mW (60% less than traditional TDS). Furthermore, an ASIC processor for the sTDS has been implemented in post-layout level using 65 nm CMOS technology with a 1 V power supply, and consumes 2.89 nJ energy/command at clock frequency of 20 MHz, and occupies 0.016 mm^2 area. Power consumption of the ASIC is 10× smaller than the FPGA implementation. In comparison to our prior work in [25], at 10 MHz clock frequency the proposed work could achieve 41× and 79× energy saving on ASIC and FPGA implementation, respectively. Also, the proposed sTDS requires 19× less memory for saving the training data. To validate the functionality of the sTDS, a Maze navigation game is designed and four human subjects were able to finish the game using the sTDS in less than 2 minutes.

ACKNOWLEDGMENT

The authors would like to thank Adam Page and Nazmus Sahadat for their efforts in this work.

REFERENCES

[1] A. O'Brien and R. M. Ruairi, "Survey of assistive technology devices and applications for aging in place," in *Advances in Human-oriented and Personalized Mechanisms, Technologies, and Services, 2009. CENTRIC'09. Second International Conference on*. IEEE, 2009, pp. 7–12.
[2] M. Kowtko, "Using assistive technologies to improve lives of older adults and people with disabilities," in *Systems, Applications and Technology Conference (LISAT), 2012 IEEE Long Island*. IEEE, 2012, pp. 1–6.
[3] C. M. Genari, B. S. Bellini, P. T. Fernandes, C. Gabriela, F. O. Lima, and L. M. Li, "Perception of bci assistive technology by post-ischemic stroke patients," in *Biosignals and Biorobotics Conference (BRC), 2013 ISSNIP*. IEEE, 2013, pp. 1–5.
[4] D. T. Pawluk, R. J. Adams, and R. Kitada, "Designing haptic assistive technology for individuals who are blind or visually impaired," *Haptics, IEEE Transactions on*, vol. 8, no. 3, pp. 258–278, 2015.
[5] D. A. Ross, "Implementing assistive technology on wearable computers," *Intelligent Systems, IEEE*, vol. 16, no. 3, pp. 47–53, 2001.
[6] D. K. Sarji, "Handtalk: Assistive technology for the deaf," *Computer*, vol. 41, no. 7, pp. 84–86, 2008.
[7] M. Sartori, M. Reggiani, E. Pagello, and D. G. Lloyd, "Modeling the human knee for assistive technologies," *Biomedical Engineering, IEEE Transactions on*, vol. 59, no. 9, pp. 2642–2649, 2012.
[8] A. M. Seelye, M. Schmitter-Edgecombe, B. Das, and D. J. Cook, "Application of cognitive rehabilitation theory to the development of smart prompting technologies," *Biomedical Engineering, IEEE Reviews in*, vol. 5, pp. 29–44, 2012.
[9] D. Leonardis, M. Barsotti, C. Loconsole, M. Solazzi, M. Troncossi, C. Mazzotti, V. P. Castelli, C. Procopio, G. Lamola, C. Chisari *et al.*, "An emg-controlled robotic hand exoskeleton for bilateral rehabilitation," *Haptics, IEEE Transactions on*, vol. 8, no. 2, pp. 140–151, 2015.
[10] A. M. Cook, B. Bentz, N. Harbottle, C. Lynch, and B. Miller, "School-based use of a robotic arm system by children with disabilities," *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, vol. 13, no. 4, pp. 452–460, 2005.
[11] M. Matsubara, T. Oba, H. Kadone, H. Terasawa, K. Suzuki, and M. Iguchi, "Wearable auditory biofeedback device for blind and sighted individuals," *IEEE MultiMedia*, no. 1, pp. 68–73, 2015.
[12] J. B. Van Erp, F. Lotte, and M. Tangermann, "Brain-computer interfaces: beyond medical applications," *Computer*, no. 4, pp. 26–34, 2012.
[13] P. Horki, D. S. Klobassa, C. Pokorny, and G. R. Muller-Putz, "Evaluation of healthy eeg responses for spelling through listener-assisted scanning," *Biomedical and Health Informatics, IEEE Journal of*, vol. 19, no. 1, pp. 29–36, 2015.
[14] H. A. Caltenco, E. R. Lontis, S. A. Boudreau, B. Bentsen, J. Struijk, and L. N. Struijk, "Tip of the tongue selectivity and motor learning in the palatal area," *Biomedical Engineering, IEEE Transactions on*, vol. 59, no. 1, pp. 174–182, 2012.
[15] M. L. Mekhalfi, F. Melgani, Y. Bazi, and N. Alajlan, "A compressive sensing approach to describe indoor scenes for blind people," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 25, no. 7, pp. 1246–1257, 2015.

[16] M. R. Williams and R. F. Kirsch, "Evaluation of head orientation and neck muscle emg signals as command inputs to a human–computer interface for individuals with high tetraplegia," *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, vol. 16, no. 5, pp. 485–496, 2008.

[17] D. J. McFarland, D. J. Krusienski, W. A. Sarnacki, and J. R. Wolpaw, "Emulation of computer mouse control with a noninvasive brain–computer interface," *Journal of neural engineering*, vol. 5, no. 2, p. 101, 2008.

[18] J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller, and T. M. Vaughan, "Brain–computer interfaces for communication and control," *Clinical neurophysiology*, vol. 113, no. 6, pp. 767–791, 2002.

[19] R. Barea, L. Boquete, M. Mazo, and E. López, "System for assisted mobility using eye movements based on electrooculography," *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, vol. 10, no. 4, pp. 209–218, 2002.

[20] C. A. M. Pereira, R. Bolliger Neto, A. C. Reynaldo, M. C. d. M. Luzo, and R. P. Oliveira, "Development and evaluation of a head-controlled human-computer interface with mouse-like functions for physically disabled users," *Clinics*, vol. 64, no. 10, pp. 975–981, 2009.

[21] C.-N. Huang, C.-H. Chen, and H.-Y. Chung, "Application of facial electromyography in computer mouse access for people with disabilities," *Disability and Rehabilitation*, vol. 28, no. 4, pp. 231–237, 2006.

[22] D. Carlson and N. Ehrlich, *Assistive Technology and information technology use and need by persons with disabilities in the United States, 2001*. US Department of Education, National Institute on Disability and Rehabilitation Research, 2005.

[23] E. R. Kandel, J. H. Schwartz, T. M. Jessell, S. A. Siegelbaum, and A. J. Hudspeth, *Principles of neural science*. McGraw-hill New York, 2000, vol. 4.

[24] G. Krishnamurthy and M. Ghovanloo, "Tongue drive: A tongue operated magnetic sensor based wireless assistive technology for people with severe disabilities," in *Circuits and Systems, 2006. ISCAS 2006. Proceedings. 2006 IEEE International Symposium on*. IEEE, 2006, pp. 4–pp.

[25] S. Viseh, M. Ghovanloo, and T. Mohsenin, "Toward an ultralow-power onboard processor for tongue drive system," *Circuits and Systems II: Express Briefs, IEEE Transactions on*, vol. 62, no. 2, pp. 174–178, 2015.

[26] C. Lau and S. OLeary, "Comparison of computer interface devices for persons with severe physical disabilities," *American Journal of Occupational Therapy*, vol. 47, no. 11, pp. 1022–1030, 1993.

[27] L. N. A. Struijk, "An inductive tongue computer interface for control of computers and assistive devices," *IEEE Transactions on biomedical Engineering*, vol. 53, no. 12, pp. 2594–2597, 2006.

[28] M. E. Lund, H. A. Caltenco, E. R. Lontis, H. V. Christiensen, B. Bentsen, and L. N. A. Struijk, "A framework for mouse and keyboard emulation in a tongue control system," in *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE*. IEEE, 2009, pp. 815–818.

[29] L. NS Andreasen Struijk, E. R. Lontis, M. Gaihede, H. A. Caltenco, M. E. Lund, H. Schioeler, and B. Bentsen, "Development and functional demonstration of a wireless intraoral inductive tongue computer interface for severely disabled persons," *Disability and Rehabilitation: Assistive Technology*, pp. 1–10, 2016.

[30] Y. Nam, Q. Zhao, A. Cichocki, and S. Choi, "Tongue-rudder: A glossokinetic-potential-based tongue–machine interface," *IEEE Transactions on Biomedical Engineering*, vol. 59, no. 1, pp. 290–299, 2012.

[31] T. S. Saponas, D. Kelly, B. A. Parviz, and D. S. Tan, "Optically sensing tongue gestures for computer input," in *Proceedings of the 22nd annual ACM symposium on User interface software and technology*. ACM, 2009, pp. 177–180.

[32] X. Huo, J. Wang, and M. Ghovanloo, "A magnetic wireless tongue-computer interface," in *Neural Engineering, 2007. CNE'07. 3rd International IEEE/EMBS Conference on*. IEEE, 2007, pp. 322–326.

[33] Z. Zhang, S. Ostadabbas, M. Sahadat, N. Sebkhi, D. Wu, A. Butler, and M. Ghovanloo, "Enhancements of a tongue-operated robotic rehabilitation system," in *Biomedical Circuits and Systems Conference (BioCAS), 2015 IEEE*. IEEE, 2015, pp. 1–4.

[34] X. Huo, J. Wang, and M. Ghovanloo, "A magneto-inductive sensor based wireless tongue-computer interface," *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, vol. 16, no. 5, pp. 497–504, 2008.

[35] X. Huo and M. Ghovanloo, "Evaluation of a wireless wearable tongue–computer interface by individuals with high-level spinal cord injuries," *Journal of Neural Engineering*, vol. 7, no. 2, p. 026008, 2010.

[36] B. Yousefi, X. Huo, and M. Ghovanloo, "Using fitts's law for evaluating tongue drive system as a pointing device for computer access," in *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE*. IEEE, 2010, pp. 4403–4406.

[37] B. Yousefi, X. Huo, J. Kim, E. Veledar, and M. Ghovanloo, "Quantitative and comparative assessment of learning in a tongue-operated computer input device—part ii: Navigation tasks," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 16, no. 4, pp. 633–643, 2012.

[38] X. Huo, H. Park, J. Kim, and M. Ghovanloo, "A dual-mode human computer interface combining speech and tongue motion for people with severe disabilities," *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, vol. 21, no. 6, pp. 979–991, 2013.

[39] A. Jafari, A. Page, C. Sagedy, E. Smith, and T. Mohsenin, "A low power seizure detection processor based on direct use of compressively-sensed data and employing a deterministic random matrix," in *Biomedical Circuits and Systems Conference (BioCAS), 2015 IEEE*. IEEE, 2015, pp. 1–4.

[40] X. Huo, J. Wang, and M. Ghovanloo, "Using magneto-inductive sensors to detect tongue position in a wireless assistive technology for people with severe disabilities," in *Sensors, 2007 IEEE*. IEEE, 2007, pp. 732–735.

[41] M. Sahadat, A. Alreja, P. Srikrishnan, and M. Ghovanloo, "A multimodal human computer interface combining head movement, speech and tongue motion for people with severe disabilities," in *Biomedical Circuits and Systems Conference (BioCAS), 2015 IEEE*. IEEE, 2015, pp. 1–4.

[42] C. Gomez, J. Oller, and J. Paradells, "Overview and evaluation of bluetooth low energy: An emerging low-power wireless technology," *Sensors*, vol. 12, no. 9, pp. 11 734–11 753, 2012.

[43] E. B. Sadeghian, X. Huo, and M. Ghovanloo, "Command detection and classification in tongue drive assistive technology," in *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*. IEEE, 2011, pp. 5465–5468.

[44] S. Mimche, D. Ahn, M. Kiani, H. Elahi, K. Murray, K. Easley, A. Sokoloff, and M. Ghovanloo, "Tongue implant for assistive technologies: Test of migration, tissue reactivity and impact on tongue function," *Archives of Oral Biology*, vol. 71, pp. 1–9, 2016.

[45] A. Kulkarni, Y. Pino, M. French, and T. Mohsenin, "Real-time anomaly detection framework for many-core router through machine-learning techniques," *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, vol. 13, no. 1, p. 10, 2016.

[46] A. Page, C. Sagedy, E. Smith, N. Attaran, T. Oates, and T. Mohsenin, "A flexible multichannel eeg feature extractor and classifier for seizure detection," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 62, no. 2, pp. 109–113, 2015.

[47] A. H. Shoeb and J. V. Guttag, "Application of machine learning to epileptic seizure detection," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 2010, pp. 975–982.

[48] J. Lindh, "Bluetooth low energy beacons," *Texas Instruments, Application Report*, 2016.

Ali Jafari is currently a Ph.D. candidate in Computer Science and Electrical Engineering department, at University of Maryland Baltimore County. His research interests include Low power Analog/Mixed signal ASIC and FPGA design of digital signal processing algorithms such as filters and machine learning algorithms, electronic sensors design, low power hardware-software embedded system design, and developing wearable monitoring systems for low power biomedical applications.

Nathanael Buswell received his B.S. degree in computer engineering from the University of Maryland, Baltimore County, in 2013. He is currently working towards his M.S. degree in computer engineering at University of Maryland, Baltimore County, with a focus on low power FPGA design, digital signal processing, and embedded systems.

Tinoosh Mohsenin is an Assistant Professor in the Department of Computer Science and Electrical Engineering at University of Maryland Baltimore County, where she directs Energy Efficient High Performance Computing (EEHPC) Lab. She received her PhD from University of California, Davis in 2010 and M.S. degree from Rice University in 2004, both in Electrical and Computer Engineering. Prof. Mohsenin's research focus is on the development of highly accurate high performance processors for machine learning, knowledge extraction and data sparsification and recovery that consume as little energy as possible. Prof. Mohsenin has over 60 peer-reviewed journal and conference publications. She currently leads a number of research projects including the design of next generation wearable biomedical processors, hardware accelerators for deep learning and convolutional neural networks, real time brain signal aritifact removal and processing for brain computing interface and assistive devices, which are all funded by National Science Foundation (NSF), Army Research Lab (ARL), Boeing and Xilinx. She has served as associate editor in IEEE Transactions on Circuits and Systems-I (TCAS-I) and currently serves as an associate editor in the IEEE Transactions on Biomedical Circuits and Systems (TBioCAS). She has served as technical program committee member of the International Solid-State Circuits Student Research (ISSCC-SRP), IEEE Biomedical Circuits and Systems (BioCAS), IEEE Circuits and Systems (ISCAS) and International Symposium on Quality Electronic Design (ISQED). She also serves as secretary of IEEE P1890 WG on Error Correction Coding for Non-Volatile Memories.

Maysam Ghovanloo received the B.S. degree in electrical engineering from the University of Tehran, and the M.S. degree in biomedical engineering from the Amirkabir University of Technology, Tehran, Iran in 1997. He also received the M.S. and Ph.D. degrees in electrical engineering from the University of Michigan, Ann Arbor, in 2003 and 2004. Dr. Ghovanloo developed the first modular Patient Care Monitoring System in Iran where he also founded a startup to manufacture physiology and pharmacology research laboratory instruments. From 2004 to 2007 he was an Assistant Professor in the Department of ECE at the North Carolina State University, Raleigh, NC. Since 2007 he has been with the Georgia Institute of Technology, School of Electrical and Computer Engineering, where he is a Professor and the founding director of the GT-Bionics Lab. He has 5 issued patents and authored or coauthored more than 200 peer-reviewed conference and journal publications on implantable microelectronic devices, integrated circuits and micro-systems for IMD applications, and modern assistive technologies. Dr. Ghovanloo was the general chair of the IEEE Biomedical Circuits and Systems (BioCAS 2015) in Atlanta, GA in Oct. 2015. He is an Associate Editor of the IEEE Transactions on Biomedical Engineering and IEEE Transactions on Biomedical Circuits and Systems. He served as an Associate Editor of IEEE Transactions on Circuits and Systems, Part II (2008-2011), as well as a Guest Editor for the IEEE Journal of Solid-State Circuits and IEEE Transactions on Neural Systems and Rehabilitation Engineering. He has also served on the Imagers, MEMS, Medical and Displays subcommittee of the International Solid-State Circuits Conference (ISSCC) from 2009-2014. He has received the National Science Foundation CAREER Award, the Tommy Nobis Barrier Breaker Award for Innovation, and Distinguished Young Scholar Award from the Association of Professors and Scholars of Iranian Heritage.