

Command Detection and Classification in Tongue Drive Assistive Technology

Elnaz Banan Sadeghian, *Student Member, IEEE*, Xueliang Huo, *Student Member, IEEE*, and Maysam Ghovanloo, *Senior Member, IEEE*

Abstract— Tongue Drive System (TDS) is a new assistive technology that enables individuals with severe disabilities such as those with spinal cord injury (SCI) to regain environmental control using their tongue motion. We have developed a new sensor signal processing (SSP) algorithm which uses four 3-axial magneto-resistive sensor outputs to accurately detect and classify between seven different user-control commands in stationary as well as mobile conditions. The new algorithm employs a two-stage classification method with a combination of 9 classifiers to discriminate between 4 commands on the left or right side of the oral cavity (one neutral command shared on both sides). Evaluation of the new SSP algorithm on five able-bodied subjects resulted in true positive rates in the range of 70-99% with corresponding false positive rates in the range of 5-7%, showing a notable improvement in the resulted true-false (TF) differences when compared to the previous algorithm.

I. INTRODUCTION

AMONG currently available assistive technologies (ATs), there are only a few that are widely accepted and used by individuals with severe disabilities in everyday living [1]. The main challenges here are for a device to be highly reliable, both in terms of safety and accurate operation, practical for individuals with various disabilities, and offer sufficient degrees of freedom to the end users. Powered wheelchairs, for example, are the only and the most reliable AT aimed for mobility. However, they require hand motion for joystick control unless they are equipped with alternative controls. Voice recognition systems are another example that cannot be used by SCI patients whose speech function may also be impaired. Sip-n-Puff switches, which operate by blowing or sucking through a straw, on the other hand, are very reliable in a variety of environments, and widely used for wheelchair control. However, they can only provide a limited number of commands (four). Brain Computer Interfaces (BCIs), have the potential to become the ideal AT by offering means to communicate with brain activities. However, the research on BCI has not yet yielded to a reliable AT that can be made available to the public [2]-[4].

Alternatively, human tongue is an attractive option for the control site in an AT. It escapes damage even in severe

spinal cord injuries or most neurological diseases. It is inherently capable of sophisticated motor control tasks and does not fatigue easily. The tongue is easily accessible inside the oral cavity which remains hidden from the sight therefore providing users with privacy [5]. By utilizing the tongue capabilities, we have developed an unobtrusive, minimally invasive AT, called Tongue Drive System (TDS), which can provide a high level of control to the user by offering 7 simultaneously available commands and has the potential to offer even more degrees of freedom [6].

The external TDS (eTDS) consists of a small permanent magnetic tracer, attached to the user's tongue by adhesives or piercing, and an array of magnetic sensors mounted on a wireless headset worn by the user. Magnetic field variations resulted from tongue movements are captured by the magnetic sensor array and wirelessly transmitted to a PC or a Smartphone (iPhone) [7]. A Sensor Signal Processing (SSP) algorithm running on the computer or Smartphone then takes charge of continuous detection and classification of the user intended commands in real-time and provides the output. The extracted commands can then be used for computer access or to control any device that can be controlled by a computer.

Since the central processing core of the TDS is its SSP algorithm, the overall characteristics of the system such as accuracy, response time, and sensitivity heavily depend on this algorithm. Some of the common pitfalls of signal processing algorithms devised for ATs include: the "Midas touch" problem i.e. the issuance of unintended commands [8], [9], sub-optimum computational cost which results in lower speed as well as excessive computational power, and insufficient sensitivity in detecting user control commands.

In previous TDS systems, we were using a single-stage classification algorithm with K-Nearest Neighbor (KNN) classifier applied on PCA components extracted from a pair of magnetic sensors. Even though the resulting performance was acceptable, the system occasionally suffered from misclassification among 7 commands especially when the subjects were using magnetic tongue studs, as opposed to glued-on magnetic tracers. The main objective of this work, however, has been to provide TDS with a more computationally efficient SSP algorithm to detect and classify up to 7 different control states as accurately and promptly as possible and in real-time. We have employed two additional magnetic sensors to achieve a better coverage over the oral cavity resulting in better resolution,

Manuscript received April 15, 2011. This work was supported in part by the National Institute of Biomedical Imaging and Bioengineering grant IRC1EB010915 and the National Science Foundation award CBET-0828882 and IIS-0803184.

Elnaz Banan Sadeghian, Xueliang Huo, and Maysam Ghovanloo are all with the GT-Bionics Lab (www.gt-bionics.org), School of Electrical and Computer Engineering at Georgia Institute of Technology, Atlanta, GA, USA (phone: 404-385-7048; fax: 404-894-4701; email: mgh@gatech.edu).

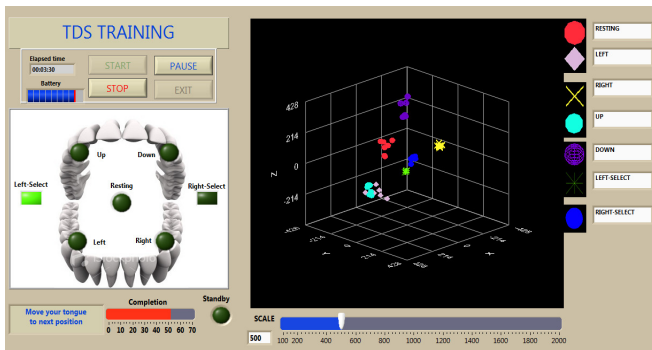


Fig. 1. TDS-GUI for 7 command training session in the LabVIEW environment.

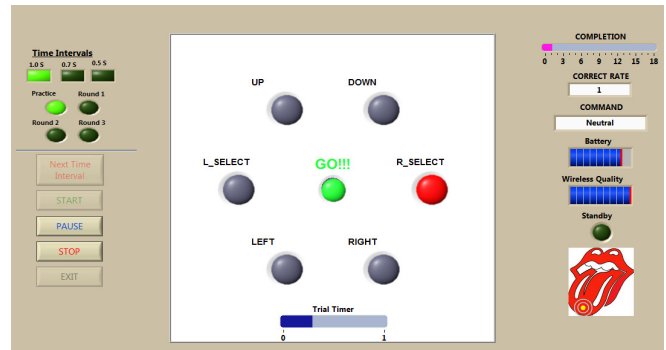


Fig. 2. TDS-GUI for 7 commands testing session with different time intervals.

classification accuracy, and noise cancellation efficacy [10].

II. EXPERIMENTAL PARADIGM

A. Graphical User Interface

The Graphical User Interface (GUI) used in this experiment consists of two parts: the *Training* session and the *Testing* session. During the training session, the subject was asked to issue randomly selected commands when their corresponding lights turn on (Fig. 1). Here, after each light turned on, there was a 2 s interval when the subject was allowed to move his/her tongue to the ordered command position followed by a 1.5 s period when the subject was asked to fix the tongue in that position while the program collects the sensor outputs for that command. In order to provide the operator with a visual feedback of the quality of the data being recorded concerning the inter cluster separability and intra class consistency; we used the differentiated sensor output to display the average of each trial as one data point in a 3-D space, as shown in Fig. 1.

For online experiment, we implemented our new SSP algorithm in C programming language and compiled it into two dynamic-link libraries (DLLs) of Train and Test functions. These functions were then employed in the LabVIEW GUI. After the training data was recorded by repeating each command for 10 times, it was then fed to the Train module. The Train module includes the training phase of the SSP algorithm to train the classifiers, and calculates the parameters needed for online classification of the commands in the testing phase.

For testing session we have used the GUI, originally designed for TDS Information Transfer Rate measurement, to evaluate our new algorithm and measure the resulted improvement in TDS functionality. In the testing session, shown in Fig. 2, one out of 6 commands was randomly selected and its light turned red during when the subject got ready to issue the depicted command. The subject was asked to make the necessary tongue movement once the center light turned green and message “Go” appeared on the screen. A hit was captured when the algorithm detecting the tongue motion moved the center light toward that command and changed its color to green. The subject was instructed to

keep his/her tongue in that position until both lights turn off and then return the tongue back to the Resting position where the algorithm would recognize it as a Neutral command. In order to challenge both the system (algorithm) and the subject, we applied three different time intervals during which the subject had to reach the command position or otherwise a miss hit would be recognized.

The task of recognizing the issued command is due to the testing phase of the SSP algorithm which was implemented as the “Test function” inside another DLL module. The LabVIEW GUI automatically feeds the output parameters of the “Train function” along with the raw data to the input of the Test module and shows the outputted command on the screen (Fig. 2).

B. Dataset

In order to measure the performance of our algorithm we collected the raw sensor outputs along with all the necessary timing information and performed an offline analysis on the data recorded from five able-bodied subjects. Four of these subjects have had tongue piercings and did not have any prior experience with TDS. The tongue studs of these subjects were replaced by magnetic tongue studs, especially manufactured for this purpose, which had a small magnet encased in their upper titanium welded ball. The last subject was however totally familiar with TDS and had the magnetic tracer temporarily attached to her tongue with tissue adhesive. To simulate the dynamic situation, this subject performed the experiment while moving around carrying a notebook computer which was running the software. All the other subjects did the experiment while sitting in front of a desktop computer. The training data consists of 12-dimensional raw sensor outputs i.e. the data recorded from 4 three-axial magnetic sensors sampled at 50 Hz frequency in addition to time and tag information which refers to transition and fixing periods. This data includes 70 trials i.e. 10 repetitions of 7 commands. The testing data was collected during 3 different time intervals of 1s, 0.7s, and 0.5s each consisting of 4 rounds of experiment. Each round includes issuing 24 random commands i.e. each command randomly occurs 4 times. We also repeated the experiment in 5 different sessions.



Fig. 3. eTDS-Headset prototype; the figure shows the alignment of four magneto-resistive sensors which was used in EMI cancellation.

III. DATA PROCESSING

A. Preprocessing

External magnetic interference (EMI) including Earth magnetic field (EMF) can potentially affect the magnetic sensor readings and reduce the signal-to-noise ratio (SNR). In order to cancel EMI effect on four sensor readings, we apply a differential noise cancellation method. In the current TDS prototype we have four 3-axial magnetic sensors mounted on two PCBs which are implemented on two poles extending from both sides of the TDS headset. Noise cancellation is performed separately on each pair of front-side, back-side, two sensors in the right, and two in the left-side of the TDS headset (Fig. 3). Here, we mathematically align two sensors in each pair and only consider their difference for future calculations. Advantageous of this process is that it also reduces the dimensionality of raw data from 12 to 6. For a complete mathematical description on this process please refer to [10].

B. Classification

In signal processing field, it is customary to combine several stages of classification when number of classes increase or different types of patterns are to be recognized [11],[12]. We have employed a two-stage classification algorithm which with near absolute accuracy can distinguish between 7 different control commands of Left, Right, Up, Down, Left-Select, Right-Select, and Resting (Neutral) inside the oral cavity (Fig. 4). In the first phase we mainly focus on performing a hundred percent accurate discrimination between left (left, up, left-select) versus right-side (right, down, right-select) commands. Here, the noise cancellation is performed on left and right-side sensors separately to provide a 6 dimensional vector including 3 components from the left and 3 from the right-side sensors. The reason for this selection is that, alternatively, the difference between front and back-side sensors masks the discriminatory information between left vs. right-side commands since they will both produce two differential vectors with similar magnitudes. Then, we calculate the Euclidean distance of an upcoming point to the left and right command-positions which are averaged from training trials. These distances are normalized to compensate for any asymmetry in user's left vs. right-side commands and then

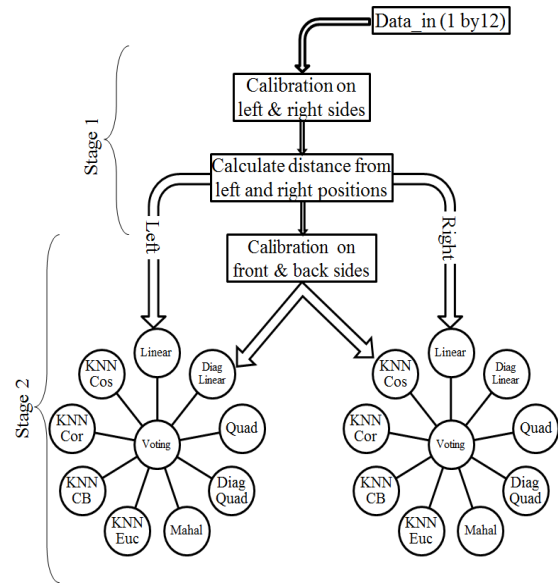


Fig. 4. Flowchart of the 2-stage classification algorithm.

compared to produce a left/right decision. Based on the outcome of the first stage, the second stage of classification is applied on either left or right side of the oral cavity to detect and discriminate between left, up, left-select, and neutral commands on the left side; or right, down, right-select, and neutral commands on the right side. Note that neutral command can be interpreted as belonging to either left or right side and therefore classified to either side in the first stage since it is going to be re-classified in the second stage and against different commands in the left or right-side as well.

Second stage of classification begins with de-noising the raw data using the front and back-side sensors. This provides another 6 dimensional vector including 3 components on the back and 3 on the front side of the oral cavity. Successively, this vector is fed to a group of 9 linear and nonlinear classifiers consisting of Linear, Diagonal linear, Quadratic, Diagonal Quadratic, Mahalanobis minimum distance, and four KNN classifiers. Four different distances used in KNN classifiers include Euclidean, Cosine, Correlation, and City Block among which the last three are defined as follows:

$$d_{\cosine} = 1 - \frac{X_1 X_2'}{\sqrt{(X_1 X_1')(X_2 X_2')}} \quad (1)$$

$$d_{\text{Correlation}} = 1 - \frac{(X_1 - \bar{X}_1)(X_2 - \bar{X}_2)'}{\sqrt{(X_1 - \bar{X}_1)(X_1 - \bar{X}_1)'(X_2 - \bar{X}_2)(X_2 - \bar{X}_2)'}}$$

$$d_{CB} = \sum_{i=1}^n |X_{1i} - X_{2i}|$$

where X_1 is the upcoming test data, X_2 is the training data recorded for every class, and \bar{X}_1 & \bar{X}_2 are their mean values respectively. We calculate the distance of each upcoming sample from the training samples in each class; however, we down-sample the training data by a factor of 10 in order to retain computational efficiency and speed for our online application. At the end, the outputs of all classifiers are combined following a Majority Voting schema to provide a

final result.

IV. RESULTS

A. Offline Experiment

In our offline analysis, we evaluated the algorithm performance throughout the whole time when the test session was running. It included both the transition phase i.e. when the tongue was moving, as well as when the tongue was in a command position. Therefore, 4 different measures of true positive rate (TPR) and false positive rate (FPR) can be provided. These measures were calculated in a sample-by-sample analysis and captured as follows:

$$TPR = \frac{TP}{TP + FN} \quad (2)$$

$$FPR = \frac{FP}{FP + TN}$$

where TP (true positive) and FN (false negative) are a true detection and a misclassified sample during the issuance of a command, respectively; FP (false positive) is a false detected command during the transition phase as well as during a neutral command; TN (true negative) is a sample during the transition or neutral command that has correctly been classified as neutral. Note that in our analysis these definitions differ from the common ones used in binary detection model since we are measuring the accurate detection of 7 commands all at the same time.

To measure the improvement achieved, the results of the new algorithm were compared with the previous algorithm [10] where all 7 commands were classified in one stage using a KNN classifier. Table 1 reveals the TPR (left side columns) and FPR (right side column) percentages for the last four rounds of all subjects. For each subject, only the results regarding the best session are shown.

TABLE I

TP (LEFT SIDE COLUMNS) AND FP (RIGHT SIDE COLUMNS) RATES CORRESPONDING TO THE BEST SESSION DURING THE LAST 4 ROUNDS WITH 0.5S TIME INTERVAL. BOLD NUMBERS REFER TO THE RESULTS OF THE NEW ALGORITHM AND THE OTHERS TO THE PREVIOUS ALGORITHM.

	Round 9		Round 10		Round 11		Round 12	
Subject 1	88	5	87	5	97	4	96	3
	78	10	79	8	81	9	81	9
Subject 2	94	7	94	8	96	5	96	4
	80	15	82	14	80	10	81	10
Subject 3	80	11	85	5	85	6	96	5
	67	11	74	7	74	8	80	9
Subject 4	70	5	76	10	88	4	85	3
	60	10	70	10	70	9	81	9
Subject 5	99	7	99	5	98	7	98	7
	90	10	90	8	95	9	92	9

B. Online Experiment

One major issue we were previously facing was the issuance of some unintended commands during the phase when the tongue was in the transition between different commands. This problem named as "Junk Commands" was better experienced with manually moving the magnet in the 3-D space inside the headset. We could find different positions at which rotating the magnet could result in a

misclassification where a command of the opposite side was being detected. The reason is that orientation of the magnet can potentially change the magnetic flow measured by the magnetic sensors and therefore affect the classifier output. Nevertheless, by relying on the magnitude of the sensor readings during the first stage of classification, we could successfully discriminate left vs. right-side commands with near absolute accuracy. As a result, by implementing the new algorithm, we covered the entire 3-D space inside the headset and recognize 7 command positions with almost no mistake.

V. CONCLUSION

In this work we implemented our new SSP algorithm in TDS software and verified the potential improvement gained in the accuracy and sensitivity of the system in detecting user-intended control commands. Offline experiments with 5 able-bodied subjects showed that with as high as 99% of true detections and as low as 5% of false positives TDS can now offer 7 different regions inside the oral cavity that can be reliably used as user control commands. Based on promising results, this shows a potential for increasing the number of commands to more than 10. We also are in the process of assessing the new system by people with severe disabilities and plan to add the proportional control capability to TDS.

REFERENCES

- [1] A.M. Cook, J.M. Polgar, and S.M. Hussey, *Assistive Technologies: Principles and Practice*, 3rd ed. Mosby, 2008.
- [2] J.D. Mill'an et al, "Combining brain-computer interfaces and assistive technologies: State-of-the-art and challenges," *Frontiers in Neuroscience*, vol. 4, article 161, Sep. 2010.
- [3] P. Brunner et al, "Current trends in hardware and software for brain-computer interfaces (BCIs)", *Journal of Neural Engineering*, vol.8, no. 2, Apr.2011.
- [4] G. Pfurtscheller et al, "Self-paced operation of an SSVEP-based orthosis with and without an imagery-based "brain switch": a feasibility study towards a hybrid BCI", *IEEE Trans on Neural Syst. Rehabilitation Eng.*, vol. 18, no. 4, pp. 409-414, Feb. 2010.
- [5] C. Lau and S. O'Leary, "Comparison of computer interface devices for persons with severe physical disabilities," *Amer. J. Occup. Ther.*, vol. 47, pp. 1022-1030, Nov. 1993.
- [6] X. Huo and M. Ghovanloo, "Using unconstrained tongue motion as an alternative control surface for wheeled mobility," *IEEE Trans. on Biomed. Eng.*, vol. 56, no. 6, pp. 1719-1726, June 2009.
- [7] J. Kim, X. Huo, and M. Ghovanloo, "Wireless control of smartphones with tongue motion using tongue drive assistive technology," *Proc. IEEE 32nd Eng. in Med. and Biol. Conf.*, pp. 5250-5253, Sep. 2010.
- [8] M. M. Moore, "Real-world applications for brain-computer interface technology," *IEEE Trans. Rehabil. Eng.*, vol. 11, no. 2, pp. 162-165, Jun. 2003.
- [9] J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller, and T.M. Vaughan, "Brain-computer interfaces for communication and control," *Clin. Neurophysiol.*, vol. 113, pp. 767-791, 2002.
- [10] X. Huo, J. Wang, and M. Ghovanloo, "A magneto-inductive sensor based wireless tongue-computer interface," *IEEE Trans. Neural Syst. Rehabilitation Eng.*, vol. 16, no. 5, pp. 497-504, Oct. 2008.
- [11] Y. Kutlu, and D. Kuntalp, "A multi-stage automatic arrhythmia recognition and classification system", *Computers in Biology and Medicine*, vol. 41, no. 1, pp. 37-45, Jan. 2011.
- [12] Polat, K., Güneş, "Classification of epileptiform EEG using a hybrid system based on decision tree classifier and fast Fourier transform," *Applied Mathematics and Computation*, vol. 187, no. 2, pp. 1017-1026, 2007.