Published in final edited form as:

Conf Proc IEEE Eng Med Biol Soc. 2010 ; 2010: 5250–5253. doi:10.1109/IEMBS.2010.5626294.

Wireless Control of Smartphones with Tongue Motion Using Tongue Drive Assistive Technology

Jeonghee Kim, Xueliang Huo[Student Member, IEEE], and Maysam Ghovanloo[Member, IEEE]

GT-Bionics Lab, School of Electrical and Computer Engineering at Georgia Institute of Technology, Atlanta, GA, USA

Maysam Ghovanloo: mgh@gatech.edu

Abstract

Tongue Drive System (TDS) is a noninvasive, wireless and wearable assistive technology that helps people with severe disabilities control their environments using their tongue motion. TDS translates specific tongue gestures to commands by detecting a small permanent magnetic tracer on the users' tongue. We have linked the TDS to a smartphone (iPhone/iPod Touch) with a customized wireless module, added to the iPhone. We also migrated and ran the TDS sensor signal processing algorithm and graphical user interface on the iPhone in real time. The TDS-iPhone interface was evaluated by four able-bodied subjects for dialing 10-digit phone numbers using the standard telephone keypad and three methods of prompting the numbers: visual, auditory, and cognitive. Preliminary results showed that the interface worked quite reliably at a rate of 15.4 digits per minute, on average, with negligible errors.

I. Introduction

Persons with severe disabilities as a result of causes ranging from spinal cord injuries (SCI) to amyotrophic lateral sclerosis (ALS) need assistive technologies (AT) on a daily basis for gaining greater independence. Among various ATs, those providing alternative control for computer access and wheelchairs are the most important for today's lifestyle since they can improve users' quality of life by enabling communication and independent mobility [1], [2].

There is a wide variety of ATs, such as sip and puff, eye trackers, head pointers, electromyography (EMG) switches, and voice controllers for individuals with severe disabilities, which are commercially available [3]. However, each of these devices has limitations that render it unsuitable or unsafe for both computer access and wheelchair control in various indoor and outdoor environments. This is why individuals eventually end up depending on and carrying two or more ATs, each for certain tasks or environments. This will impose learning and financial burden on the individual to begin with, as well as dependence on a caregiver for switching between ATs and adjusting them on an ongoing basis. Sometimes individuals forgo certain tasks or avoid certain environments simply because their AT may not be assistive enough.

There are certain aspects of the ATs that result in their uselessness in certain environments or for certain tasks. Some of these factors depend on the type of inputs that the AT requires

Correspondence to: Maysam Ghovanloo, mgh@gatech.edu.

32nd Annual International Conference of the IEEE EMBS
Buenos Aires, Argentina, August 31 - September 4, 2010

to operate. For instance, eye trackers depend on eye motion and always interfere with their users' normal visual tasks. They are also affected by ambient light and cannot be used in the dark or direct sunlight. Or voice controllers need a quiet environment because they are highly susceptible to external noise and interference. Other factors that directly affect the AT usability are size, weight, functionality, ease of use, maintenance, cost, and even the appearance of the AT.

In order to achieve a single AT that can be used for various tasks in different environments, we have been developing the Tongue Drive System (TDS) over the last few years, which block diagram is shown in Fig. 1 [4], [5]. TDS is a wireless and wearable assistive technology that can help people with severe disabilities control their environments and access computers using their tongue motion. TDS translates specific tongue gestures to commands by detecting the position of a small permanent magnetic tracer on the users' tongue [6].

Up until now the TDS required a laptop PC to present the users with a graphical user interface (GUI), run the sensor signal processing (SSP) algorithm to detect user commands, and execute those commands on the same computer or deliver them to other devices such as wheelchairs, phones, or TV. In this paper, we present for the first time, a TDS prototype that runs entirely on a smartphone platform (iPhone/iPod Touch) with no need for a PC. This makes the TDS truly wearable and significantly more usable in various environments. While potentials are unlimited, the first application that we have targeted, tested our TDS-iPhone interface for, and reported in this paper, is the ordinary use of the smartphone as a cellular phone to make phone calls.

II. System Architecture

Fig. 2 shows the external TDS-iPhone wireless interface prototype, being used by one of the authors. This prototype consists of a standard eTDS headset, a permanent magnetic tracer attached to the user's tongue, an iPhone, a small RF module built in a 30-pin iPhone style connector (inset), and an application running on the iPhone with a GUI customized for our current trial. The GUI shows a standard telephone keypad along with graphical representation of the user's selected TDS commands on the bottom of the screen and the target/dialed numbers on top of the screen. In the following we explain the main system components in more detail.

A. External Tongue Drive System (eTDS) Headset

The current eTDS prototype consists of a pair of 3-axial magnetic sensors (Honeywell, Plymouth, MN) that are held bilaterally near the users' cheeks with a pair of flexible goosenecks to facilitate sensor positioning. The control unit and power source (a pair AAA batteries) were housed inside a small box on top of the headset (see Fig. 2). The control unit includes sensor interfacing and power management circuitry, a low-power system-on-chip microcontroller (CC2510, Texas Instruments, Dallas TX) with built-in 2.4 GHz RF transceiver, and a miniature surface mount antenna. The control unit samples each sensor at 50 Hz, while activating only one sensor at a time to save power. After all sensors are read, they are packed in a data frame, and wirelessly transmitted to the RF interface module. A more detailed description of the eTDS control unit can be found in [4]. A small disk-shaped ($\varnothing 4.8 \text{ mm} \times 1.5 \text{ mm}$) rare earth permanent magnet, attached near the tip of the users' tongues, is also needed for tracking the tongue position.

B. Sensor Signal Processing Algorithm

Every 20 ms, each 3-axial magnetic sensor provides three samples by measuring the components of the magnetic field vector at the sensor position along the X, Y, and Z axes. In

our real time SSP algorithm, a moving window picks up 18 values from three consecutive samples, taken from the two modules in a span of 60 ms, and passes them through a 3-D stereo-differential transformation, to cancel the effects of external magnetic interference (EMI) from a variety of sources particularly the earth magnetic field [4]. Transformed vectors are then reflected onto a virtual 3-D space using principal component analysis (PCA). Thus, every tongue position finds a unique representation in this 3-D PCA space.

In order to identify the six user-defined tongue positions which are associated with the individual TDS commands, users train the system by moving their tongues among these six positions, such as touching particular teeth with the tip of their tongues, and repeat this task 10 times, while the SSP algorithm collects training samples and reflects them onto the PCA space to create a cluster for each TDS command. There is also a 7th position, aligned with the tongue resting position, which is identified as the TDS neutral command. We refer to this procedure as the TDS training session (see Fig. 4).

During normal operation of the system (following the training session), every new incoming sample is transformed and reflected onto the PCA space to be classified using a method known as K-nearest-neighbors (KNN), which assigns that sample to one of the seven classes based on its proximity to the command clusters. A more detailed description of the TDS SSP algorithm can be found in [4].

C. Tongue Drive-iPhone RF Module

We have established the wireless connection between the eTDS headset and iPhone through a small RF module, placed in a 30-pin iPhone style connector, which is shown in Fig. 2 inset. The RF module electronics, which key components are a CC2510 microcontroller and a miniature 2.4 GHz antenna, are small enough to fit inside the connector's plastic encasing ($25 \times 25 \times 10 \text{ mm}^3$). The RF module consumes 71 mW from the iPhone battery when it is receiving data from the eTDS control unit at a data rate of up to 2 Mbps. The received data is sent to the iPhone through its serial port at a baud rate of 115.2 kbps. Even though this rate is much lower than the wireless link bandwidth, it is sufficient for the eTDS 50 Hz sampling rate, which is in turn sufficient for detecting the tongue motion in real time without noticeable delays.

D. Tongue Drive-iPhone Interfacing Software

The latest iPhone (3GS) and iPod Touch (3rd generation) are equipped with ARM Cortex-A8 CPU (S5PC100, Samsung, Korea) running at 600 MHz, 256 MB DRAM, and 3.5" multi-touch display with 480×320 pixel resolution (163 ppi). Before migrating the SSP algorithm from PC (laptop) to iPhone, we were concerned that the lower CPU power and memory in the iPhone may result in malfunction or noticeable delays in the detection and execution of the TDS commands. Therefore, we were ready to decrease the TDS sampling rate from 50 Hz down to 25 Hz or even 13 Hz, which was the rate used in earlier generations of the eTDS headset, to reduce the volume of computations.

The earlier versions of the SSP were developed in the LabVIEW and MATLAB environments [4]. In the current prototype, we first improved the efficiency and speed of the SSP algorithm by implementing the processing engine in Microsoft Visual C++ language on the PC, while maintaining the GUI shell in LabVIEW, and then migrating both to Objective-C language, provided in the iPhone software development kit (SDK v3.1.3, Apple Inc, Cupertino, CA). We found that the iPhone processor can properly handle the SSP algorithm with the TDS headset operating at 50 Hz. The TDS latency, running entirely on the iPhone, was measured to be less than 300 ms, which is acceptable for our real time applications and experiments.

Considering our first and foremost TDS-smartphone application, which is enabling TDS users to use their phones to make phone calls, we chose the iPhone's standard telephone keypad as the main component of the GUI for dialing phone numbers (see Fig. 2). Even though this standard 4×3 key arrangement, which is originally developed for fingers, may not be the easiest or most intuitive way of dialing numbers with the tongue motion, it is quite common in touchtone telephones and many other cellular phones. Hence, our TDS-iPhone interface and experimental results can be extended and validated across many platforms.

Fig. 3 shows the flowchart of the TDS-iPhone interfacing software. Its first task is to open the iPhone serial port to be prepared for receiving the sensor data from the RF module in real time. Communication between the CC2510 on the RF module and iPhone is via universal asynchronous receiver/transmitter (UART) port, which has a pair of dedicated pins on each device. Raw sensor data is transmitted from the CC2510 on the eTDS control unit in 28-byte packets to the CC2510 on the RF module, which adds two header bytes to it before relaying it to the iPhone. The iPhone then decodes the packet to recover the individual sensor data samples and feeds them into the SSP algorithm to identify one out of the seven TDS commands (0–6 in Table I).

Table I shows how TDS commands have been associated to the smartphone GUI functions for the task at hand, i.e. making a phone call. The tongue resting position is neutral (0) and does not have any effect on the iPhone. Four directional commands move the highlighted key on the keypad and Call button in four directions (see Fig. 2). The last two commands are dedicated to selecting (i.e. pressing) a key and deleting the last entered number (i.e. back space). The default highlighted key is #5 in the center of the keypad. In order to slow down the highlighted key movements and minimize unintentional commands, the SSP output was low pass filtered by requiring 25 consecutive detections of the same command before it was executed. Therefore, the maximum speed of entering new commands was reduced to two per second.

There are other blocks in Fig. 3, which are responsible for prompting the number to be dialed to the users, keeping track of time, correctly or incorrectly dialed numbers, and showing the selected TDS command at the bottom of the GUI screen.

III. Performance Evaluation

Four able-bodied volunteers, three males and one female aged from 26 to 29 years old, participated in this trial to evaluate the functionality of the TDS-Smartphone wireless interface, running on an iPod Touch. Two of the subjects had used TDS multiple times before but not on a regular basis. Subjects wore the TDS headset, as shown in Fig. 2, and a permanent magnetic tracer was attached on their tongue, 1 cm from the tip, using Cyanoacrylate adhesive (GluStitch Inc., Point Roberts, WA).

A dedicated GUI, shown in Fig. 4, was then launched for calibrating the TDS and training it as described in section II.D. We decided to facilitate the initial TDS calibration and training procedure by running it on a PC, using a 22" LCD monitor screen, even though these steps could also be conducted on the iPhone. Fig. 4 shows the training GUI in the LabVIEW environment for defining the tongue positions associated with each individual TDS command. During the training session, the computer provides the subject with a visual cue by randomly selecting one out of seven commands and turning on its light. The subject moves his/her tongue to that position and holds it stationary while a beep is played. Then a different light turns on. This cycle continues 70 times over the ~5 min training session until each command repeats 10 times. At the end, the collected data, which indicates command

clusters in the PCA space, is transferred to the iPhone to be used in the SSP algorithm in real time.

After training the system, subjects were asked to enter 10-digit phone numbers on the iPod and call the number using TDS while holding the iPod comfortably in their right hand, about 30 cm from their face (Fig. 2). We used three methods for prompting the subjects with the phone numbers to be dialed, and the goal was to compare the efficacy of these methods in terms of the completion time and rate of errors:

- Visual: target numbers appear in black on the top of the GUI screen, as shown in Fig. 2, and the subject simply enters the same number in blue in the following line.
- Auditory: subjects wear earphones attached to the iPod, which plays the target numbers in three blocks of 3, 3, and 4 digits, one digit at a time. Subjects can enter the numbers as soon as they hear them. They can also repeat each block by moving the highlighted key to the right of the Call button and selecting it (see Fig. 2).
- Cognitive (using memory): target numbers are randomly selected from three numbers that subjects know by heart, often their mobile, home, and parents' phone numbers. The GUI prompts "mobile", "home", or "parents" on the screen and the subjects enter the numbers in the following row.

Once all 10 digits were entered, regardless of the method of prompting, subjects were supposed to select the Call button to make a phone call.

In each round, subjects were randomly assigned one out of the three prompting methods and entered 6 phone numbers. In the case of the visual and auditory, these numbers were all different, and no digits were repeated twice (all 10 digits were present in all phone numbers). However, in the cognitive method, each one of the three memorized phone numbers was repeated twice to reach a total of 6. Each subject went through 10 rounds, the first of which was for practice, followed by 3 randomly selected rounds of each prompting method (60 phone numbers in total). The entire session took 70~80 min including the TDS calibration and training.

Experiment Results

All subjects successfully completed the assigned task and their performance improved over time. The TDS-iPhone interface worked seamlessly as if the TDS SSP was running on a PC. Fig. 5 shows a summary of the experimental results. The average time for entering a 10-digit phone number and selecting the Call button was ~42.7 s. Fig. 5a shows that the method of prompting the numbers had little effect on the speed of dialing with visual, auditory, and cognitive methods taking 41.76 ± 7.25 s, 43.57 ± 8.02 s, and 42.88 ± 7.11 s, on average, respectively. In terms of digits per minute (DPM), these methods yielded 15.8, 15.1, and 15.4, respectively [7].

Fig. 5b shows that the percentage of errors over 540 digits entered by each subject was quite low. This is probably because the TDS was slowed down (only two commands per second) and no limitation was imposed on the subjects in terms of dialing duration or total number of commands other than asking them to dial the phone numbers as fast and accurately as they can. In fact the only error was a result of a mistake in recalling a phone number by one of the subjects in the cognitive mode. While these results do not provide enough information about the most intuitive method of prompting or presenting phone numbers to TDS users, they show the functionality, high level of accuracy, and reliability of the current TDS-iPhone prototype, which is necessary for real life usage of the TDS by those with severe disabilities.

IV. Conclusion

Tongue Drive System is now wirelessly linked to a smartphone (iPhone) by a small RF module that is slightly larger than a quarter and connects to its 30-pin charging port. We have also developed firmware that takes care of reliable wireless communication between the eTDS headset and the RF module using a pair of system-on-chip wireless micro-controllers, as well as efficient software that handles the entire SSP algorithm and the GUI in real time using the iPhone's limited processing resources. The significance of this TDS-smartphone interface is that it makes the TDS truly wearable by relieving its users from the burden of carrying a PC on their wheelchairs. Even though we have implemented the present TDS interface on an Apple iPhone platform due to its popularity and powerful programming environment, it is relatively straight forward to migrate to other popular platforms such as Windows Mobile, Google's Android, Nokia's Symbian, and even more powerful and compact systems to come. Further evaluation of the TDS-smartphone interface is needed by potential end users not only in the controlled laboratory or hospital environments but also in real life situations, in their homes, workplaces, and outdoors.

Acknowledgments

This work was supported in part by the National Institute of Biomedical Imaging and Bioengineering grant 1RC1EB010915 and the National Science Foundation award CBET-0828882 and IIS-0803184.

References

1. National Institute of Neurological Disorders and Stroke (NINDS). NIH, Spinal cord injury: Hope through research, [Online]. Available: http://www.ninds.nih.gov/disorders/sci/detail_sci.htm
2. Carlson, D.; Ehrlich, N. Report of US Department of Education, National Institute on Disability and Rehabilitation. Washington, D.C.: 2005. Assistive technology and information technology use and need by persons with disabilities in the united states.
3. AbleData, [Online]. Available: <http://www.abledata.com/>
4. Huo X, Wang J, Ghovanloo M. A magneto-inductive sensor based wireless tongue-computer interface. *IEEE Trans Neural Syst Rehabil Eng*. Oct.2008 16:497–504. [PubMed: 18990653]
5. Huo X, Ghovanloo M. Using unconstrained tongue motion as an alternative control surface for wheeled mobility. *IEEE Trans on Biomed Eng*. June; 2009 56(6):1719–1726.
6. Huo X, Ghovanloo M. Evaluation of a wireless wearable tongue computer interface by individuals with high level spinal cord injuries. *Journal of Neural Engineering*. Apr.2010 7(2):026008.
7. MacKenzie, IS. Mobile text entry using three keys. *Proc. 2nd Nordic Conf. on Human-Computer Interaction, NordiCHI*; New York. Oct. 2002; 2002. p. 27-34.

Fig. 1.
Block diagram of the external Tongue Drive System (eTDS) [4].

Fig. 2.

Tongue Drive-iPhone wireless interface prototype consists of a small RF module, built on a 30-pin iPhone connector (inset), and an application running on the iPhone with a GUI customized for our trial that shows target/dialed numbers, a telephone keypad, and selected TDS commands.

Fig. 3.
Flowchart of the Tongue Drive-iPhone interfacing software.

Fig. 4. GUI shown on a 22" PC screen for the TDS training session to define the tongue positions associated to each TDS command.

Fig. 5. (a) Average completion time to enter a 10-digit telephone number with 95% confidence interval. (b) Error rate over a total of 54 trials (540 digits).

Table 1

Tongue Drive-Smartphone Control Commands

Command	Function	Command	Function	Command	Function
0	Neutral	3	Up	5	Select
1	Left	4	Down	6	Delete
2	Right				