The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

Real-time Implementation of a Non-invasive Tongue-based Human-Robot Interface

Michael Mace, Khondaker Abdullah-Al-Mamun, Ravi Vaidyanathan, Shouyan Wang and Lalit Gupta

***Abstract*— Real-time implementation of an assistive human-machine interface system based around tongue-movement ear pressure (TMEP) signals is presented, alongside results from a series of simulated control tasks. The implementation of this system into an online setting involves short-term energy calculation, detection, segmentation and subsequent signal classification, all of which had to be reformulated based on previous off-line testing. This has included the formulation of a new classification and feature extraction method. This scheme utilises the discrete cosine transform to extract the frequency features from the time domain information, a univariate Gaussian maximum likelihood classifier and a two phase cross-validation procedure for feature selection and extraction. The performance of this classifier is presented alongside a real-time implementation of the decision fusion classification algorithm, with each achieving 96.28% and 93.12% respectively. The system testing takes into consideration potential segmentation of false positive signals. A simulation mapping commands to a planar wheelchair demonstrates the capacity of the system for assistive robotic control. These are the first real-time results published for a tongue-based human-machine interface that does not require a transducer to be placed within the vicinity of the oral cavity.**

***Index Terms*— Tongue-movement ear pressure signals, real-time classification, assistive HMI.**

I. Introduction

This paper builds on previous work by introducing initial real-time results for a tongue-movement based communication concept to generate, detect and classify signals online [1], [2]. The output of this classification process can then be used for control of peripheral assistive devices, aiding in the independence of the user. Hands-free communication and control of assistive devices is essential if an individual who has limited upper body mobility is to be more autonomous and therefore less reliant on carer's, family and society. Beneficiaries of said devices include but are not limited to, individuals suffering from spinal cord injury, amputation/disarticulation, stroke, congenital limb deformities and arthritis. The prevalence of these afflictions has been well documented in previous papers [1], [2], as well as across the assistive technology literature.

The user expresses their intention by impulsive actions of the tongue, creating acoustic signals within the ear canal. These signals have been coined as tongue-movement ear pressure (TMEP) signals, due to the nature of their generation and evolution within the oral and auditory regions. The tongue has been chosen to express the user's intention due to it rarely being affected by said afflictions, with it also allowing for a multitude of distinct actions to be chosen when devising the task specific instruction set. It is also known that the tongue has rich sensory and motor cortex representation that rivals that of fine finger control [3]. The actions themselves involve placement of the tip of the tongue at the base of the central incisor, left or right first molar and flicking the tongue up (up/backward, left or right action) and placing the tip of the tongue against the palate and flicking down (down/forward action). The right action is shown pictorial in Fig. 1b and the associated waveform consisting of one second of data sampled at 8 kHz is shown in Fig. 1a.

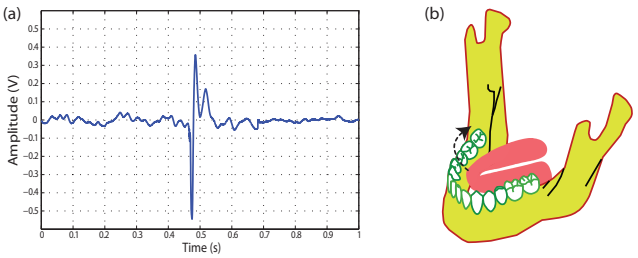


Fig. 1. (a) Waveform corresponding to right action (1 sec of data sampled at 8 kHz). (b) Pictorial representation of the right action during execution (arrow shows trajectory of tongue tip).

Using the tongue as command input is not a new concept, with the large number of tongue-based assistive HMI's testament to its functionality. An overview of the existing assistive HMI technologies which use both the tongue and alternative input methods are presented in [2]. The fundamental difference of this unique system to others being the manner in which these tongue actions are executed and monitored. Although the execution of these tongue actions feel natural they are not normal movements, allowing them to be differentiated from potentially interfering signals that occur in everyday life. These interfering signals can be categorized into internal artifacts such as speech, swallowing and coughing and also external noise such as conversation

This research was supported by the UK Engineering and Physical Sciences Research Council (EPSRC), grant EP/F01869X

Michael Mace is with the Department of Mechanical Engineering, Faculty of Engineering, University of Bristol, UK, BS8 1TR (email: mike.mace@bristol.ac.uk)

Khondaker Abdullah-Al-Mamun and Shouyan Wang are with the Institute of Sound and Vibration Research (ISVR), University of Southampton, UK, SO17 1BJ (email: [kam1e08,sy.wang]@soton.ac.uk)

Ravi Vaidyanathan Bristol Robotics Laboratory, Department of Mechanical Engineering, University of Bristol, Bristol, BS8 1TR, UK and the US Naval Postgraduate School, Monterey, CA, USA, 93943 (email: r.vaidyanathan]@bristol.ac.uk)

Lalit Gupta is with Department of Electrical and Computer Engineering, Southern Illinois University, IL, USA, 62901 (email: lgupta@siu.edu)

and road noise. Initial off-line results which consider these options are presented in [4], [5]. The benefits of this system include the lack of intrusion whilst donning the earpiece, simplicity in terms of execution of the natural instruction set, cost, portability, hygienic to use and unobtrusive to don.

An overview of the real-time system is presented in terms of its instrumentation and acquisition, preprocessing (filtering, detection and segmentation) required and feature extraction and classification methods. Two classification strategies are outlined and empirical results are presented and compared for both. The system is then shown applied in simulation to real-time control of a wheelchair moving through a constrained 2D environment with conclusions drawn in Section V.

II. System Overview

A. Instrumentation and acquisition

The hardware is centered around a small microphone positioned comfortably within the outer ear canal (external acoustic meatus) by an associated earpiece housing. This housing can either be of generic stock utilising a replaceable foam tip as seen in Fig. 2 or custom moulded from a silicon impression of the user's ear canal, with the moulded hollow acrylic housing commissioned from a hearing aid manufacturer [6]. This work focuses on the generic earpiece which is advantageous in terms of cost and time of manufacture with the trade-off being decreased passive noise rejection and microphone localization which is important due to the sensitive nature of the signals. Fig. 2 shows the generic earpiece both out of the ear and inserted into the ear of an individual.

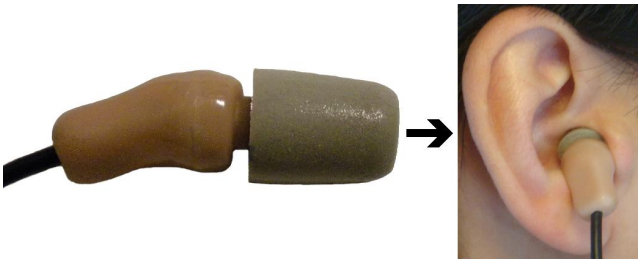


Fig. 2. Generic earpiece - showing it both out of the ear (left) and inserted into the ear canal (right)

The hardware as a whole consists of the microphone transducer converting TMEP signal to electrical signal, an analogue-to-digital (AD) card converting this electrical signal to a digital signal and the processor which turns this discretized signal into a digital output (due to the signal processing and classification stages). This hardware can be represented as an open loop block diagram shown in Fig. 3.

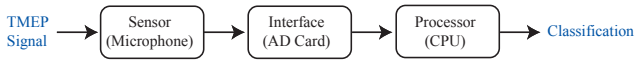


Fig. 3. Block diagram showing the system's required hardware.

TABLE I
Conditions for detecting a TMEP signal

Condition	Reason
$E_{p,q} > T_L$	The energy must exceed and recede a lower threshold to inhibit segmentation of lower energy signals
$E_{p,q} < T_U$	The energy can't exceed an upper threshold to inhibit high energy signals such as coughing
$q - p > D_L$	The time to exceed and recede this lower threshold must be greater then a lower detection duration window to inhibit impulsive signals
$q - p < D_U$	The time to exceed and recede this lower threshold must be less then an upper detection duration window to inhibit longer signals such as swallowing

B. Preprocessing: filtering, detection and segmentation

The main stages for processing and classifying the incoming digital TMEP signal and outputting a classification class are shown in Fig. 4. This representation not only applies to this specific system but is a generalisation to any communication/classification system which takes an arbitrary analogue signal as input and requires classification to 1 of 'C' classes. The filtering block is optional and can simply be an anti-alias analogue filter realised in the hardware.

Before classification can proceed the continuous stream of digital data has to be monitored so that when a TMEP action occurs it can be detected and the signal segmented appropriately. The segmentation process involves creating a data segment of finite dimension that can then be processed. The detection phase is based on similar methods used in automatic speech recognition (ASR) systems and uses the short-term energy (STE) of the incoming signal [7]. The STE can be calculated causally from:

$$E_n = \sum_{k=n-W_E+1}^{n} (x[k])^2$$

where $x[k]$ is the incoming digital stream, E_n is the STE at time index n and W_E is the finite window length that the STE is calculated in.

Due to this causality in the calculation, it is required that a circular buffer is implemented so that historical time domain data can be saved for the required computation, analysis and visualisation. The STE is continually monitored and when certain criteria outlined in Table. I are fulfilled, a TMEP signal is detected.

The segmentation process also requires the use of the STE, allowing very fast computation during the detection and segmentation phases (as the STE only requires $2W_E - 1$ operations). This is especially beneficial as the STE calculation and detection checking is occurring continually and thus faster computation will avoid any latency issues. To segment a signal, historical STE data is required and therefore not only is a circular buffer required for the time domain information but also for the STE data. This second buffer reduces computation further as a new STE value can be updated using:

$$E_{n+1} = E_n + x[n+1]^2 - x[n-W_E]^2$$

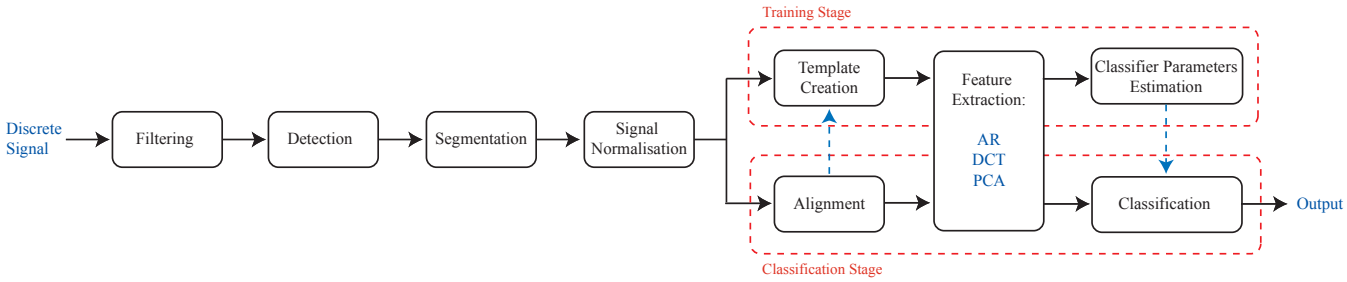


Fig. 4. Block diagram representing the various processing stages required, taking the digitized TMEP signal as input and outputting a classification class.

thus reducing the computation to only 6 operations. As the STE is also essentially a moving average filter (with W_E taps), it can be calculated at a much lower frequency then the actual sampling rate, further reducing the load on the processor.

The signal is initially segmented using the position, t, of maximum STE, E_t, located in the range $[p,q]$ where E_n exceeds and then recedes the lower threshold T_L while still fulfilling the remaining detection conditions. Thus the initial endpoint is given by

$$t = \arg\max_n \{E_p, E_{p+1}, \ldots, E_q\}$$

The initial start point is then given by $r = t - N$, where $N = 0.2fs$ i.e. the time a TMEP signal typically takes to execute and it's associated waveform to completely decay. The signal is then shifted so that the midpoint of the 1D distribution $x[k]^2/\sum_{j=r}^{t} x[j]^2$ is located at the centre of the segment window. The signal is finally normalised by $\hat{x}[k] = (x[k] - \bar{x}[k])/\sigma_x[n]$ to ensure that no class dominates the classification process.

C. Feature extraction and classification

For the initial real-time testing, two different classification algorithms have been implemented, namely the decision fusion classifier (DFC) and a univariate Gaussian classifier utilising discrete cosine transform (DCT) coefficients as input. Before classification can begin training vectors need to be collected so that:

- A template for each action class can be created. This is non-specific to the classification strategy.
- Any classifier or feature extraction parameters can be estimated from the training set. This is specific to the classification strategy.

The steps associated with the training stage are highlighted in Fig. 4.

*1) **Decision fusion classification**:* The DFC algorithm is based around a re-ranking of the time domain samples in order of classification probability on the training set. The classification metric used is a Euclidean norm nearest mean discriminant function which takes a segmented and aligned TMEP vector as input $x[k]$, and outputs a scalar value associated with each sample point for each template $\bar{x}_c[k]$, i.e.

$$f_c\{x[k]\} = x[k]\bar{x}_c[k] - (1/2)\bar{x}_c[k]^2$$

Initial classification is then achieved by finding the argument of the maximum of these 'C' scalar values across the action set:

$$m_k^* = \arg\max_c \left\{ f_c\{x[k]\} \right\}$$

Initially this discriminant methodology is applied across the entire training set and for each sample the probability of correct and misclassification to each of the remaining classes is aggregated over the number of training vectors in each class. Thus a log-likelihood matrix, $\mathscr{L}_{N\times c\times c}$ can be built using the probability of sample k, from an action c, being classified as class m_k^*, with each element calculated using

$$\mathscr{L}_k(c/m_k^*) = \log\left(\frac{\sum_{i=1}^{N_{tr}} \delta_k^i}{N_{tr}}\right)$$

where δ_k^i is a kronecker delta function of the form:

$$\delta_k^i = \begin{cases} 1 & \text{if } m_k^* = c \\ 0 & \text{if } m_k^* \neq c \end{cases}$$

The sample space is then ranked based on these probabilities, with the indices which give the highest correct classification averaged across the C-actions, given the highest ranking.

For classification of an incoming test vector, it is initially aligned to each template and then reordered according to the ranking stipulated within the training phase. This allows for a number of the end ranks to be removed (post-reordering) as these do not actually aide in the classification, with the length D, of the remaining feature vector determined empirically and dependant on required computational efficiencies. For this paper, as computational speed was not being considered and due to the fact that the probabilities inherently weight the effect that each sample has on the classification i.e. no over-fitting occurs, the entire re-ranked vector was used as input into the classifier (i.e. $D = N$). This reordered (and reduced) vector is then input into the Euclidean norm nearest mean discriminant function with each template after the associated cross-correlation alignment. The corresponding likelihood probabilities of each of the scalar outputs for correct classification and the misclassification to one of the '$C-1$' remaining actions, as found in the training phase, are concatenated at each sample to form 'C' probability vectors. This allows a marginal likelihood probability to be found for each action set, by summing down the probability vectors as

given by

$$P_c\{x[k]\} = \sum_{k=1}^{D} \mathscr{L}_k(m_k^*/c)$$

with the test vector ultimately classified to the class which maximises this marginal likelihood value, given by

$$m_D^* = \arg\max_c \left\{P_c\{x[k]\}\right\}$$

The DFC algorithm is explained with more rigor and in finer mathematical detail in [1].

*2) **Univariate Gaussian classifier with DCT features**:* The second classification scheme implemented is based around a maximum likelihood classifier assuming that each feature is normally distributed and statistically independent. DCT features are used as input into the classifier. This assumption of statistical independence implies that each feature within the input vector is uncorrelated and equivalently that the covariance matrix associated with the multivariate case is diagonal i.e the multivariate distribution simplifies to a univariate Gaussian. A univariate Gaussian maximum likelihood classifier is of the form:

$$g_c\{s[k]\} = -\sum_{k=1}^{D} \left\{ \frac{(s[k]-\bar{s}_c[k])^2}{2\sigma_c[k]^2} + \frac{1}{2}\log\sigma_c[k]^2 \right\}$$

where $s[k]$ is the discrete input feature vector at sample k, $\bar{s}_c[k]$ is the mean of the feature vector and $\sigma_c[k]$ is the standard deviation of the feature vector, both estimated from the training set for each class. Classification of an incoming test vector is then achieved by assigning to the class which gives the highest probability:

$$m_D^* = \arg\max_c \left\{g_c\{s[k]\}\right\}$$

The input feature vector to the univariate Gaussian classifier uses a DCT to transform the time domain information of a segmented and aligned TMEP signal to a frequency domain representation. The DCT is of the form:

$$s[k] = \sqrt{\frac{2}{N}} C(k) \sum_{n=0}^{N-1} x[n] \cos\left(\frac{\pi(2n+1)k}{2N}\right)$$

where $x[n]$ is the discrete time domain signal, $s[k]$ is a vector containing the DCT coefficients, $C(k)$ is a constant dependent on whether k is at the boundary and N is the length of the TMEP segment. The DCT was selected due to its ability to compact more of the signal energy onto fewer coefficients and is widely used in lossy data compression applications due to this superior compaction capacity [8]. In this regard, it has comparable performance to principle components analysis which is considered optimal in the least mean error sense, with the additional benefit that its basis matrix is independent of its training set and therefore it is not susceptible to associated estimation, generalisation and dimensionality issues.

The DCT can be represented as a matrix operation, $S = \Phi X$ where S and X are column vectors and Φ is the DCT basis matrix. This allows for rows of Φ to be discarded and thus certain selected DCT features to be removed using a 2-phase feature extraction procedure. This procedure incorporates a feature ranking phase followed by a feature selection phase, both utilising a K-fold cross-validation subroutine. Cross-validation has been widely used in model parameter estimation and is known to reduce over-fitting to the training set [9]. Prior to the cross-validation procedure the training set is split into K-disjoint subsets (folds). To distribute the signals evenly among the folds the following equation is used:

$$F_K[i] = \begin{cases} \lfloor \frac{N_{tr}}{K} \rfloor + 1 & \text{if } i \leq rem \\ \lfloor \frac{N_{tr}}{K} \rfloor & \text{if } i > rem \end{cases}$$

where

$$rem = N_{tr} - \lfloor \frac{N_{tr}}{K} \rfloor K$$

and $\lfloor\rfloor$ represents the floor function and F_K is a vector containing the number of signals in each fold. Each fold can then be iteratively used as separate validation sets, while the rest are used to train the classifier. The goodness-of-fit of the current model configuration is then the averaged classification accuracy across the validation sets. The first phase involves ranking the features independently based on their individual averaged classification accuracies across the folds. This can be calculated from

$$\alpha[k] = \frac{1}{KC} \sum_{c=1}^{C} \sum_{i=1}^{K} \frac{1}{F_K[i]} \sum_{j=1}^{F_K[i]} \delta_{ij}^c[k]$$

where $\delta_{ij}^c[k]$ is a kronecker delta function:

$$\delta_{ij}^c[k] = \begin{cases} 1 & \text{if } m_D^* = c \\ 0 & \text{if } m_D^* \neq c \end{cases}$$

and is applied to the (univariate gaussian) classification of each DCT coefficient $s_{ij}^c[k]$ (post cross-correlation alignment to each template c), of each validation signal j, in each fold i. With m_D^* indicating classification of a scalar input, as a single feature is being input into the classifier. The N features are then ranked in order of largest $\alpha[k]$. The statistical independence assumption negates the need for a search algorithm as the combination of features is irrelevant.

The second phase is to cross-validate the number of features, D, that is to be input into the classifier using this previously found feature ranking. This allows the optimal number of features to also be selected based on the same cross validation criterion. This is done in a similar manner as the first phase except the input to the classifier is no longer an individual feature but a vector containing the best k features. The number of features D is then selected based on the value which gives the highest α. The corresponding D rows can be pulled from the DCT basis matrix creating a new transformation matrix $\Phi_{D\times N}^*$. For fast calibration, a training set of $N_{tr} = 30$ was used. [9] provides further details as to the impact of varying the number of folds and its impact on training.

III. Real-time results

Preliminary real-time results are presented from one individual for three tongue actions, namely a left, right and up

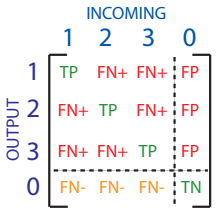
movement. These three actions have been chosen as these readily map to the wheelchair control scheme outlined in the adjoining section, with the individual familiar with making the actions prior to testing. The empirical procedure has been rigorously designed to allow not just for the segmentation and classification of TMEP signals but also equal opportunity of interfering non-TMEP signals to be classified. A sampling rate of 8 kHz was chosen due to interfering signals such as speech and coughing being in the frequency range of 0-4 kHz, with future interference rejection schemes requiring this higher frequency content to differentiate it from the controlled TMEP actions. Initially a short calibration period was necessary so that the various segmentation thresholds which are unique to the individual could be selected.

A. Testing procedure

A training ensemble is necessary from the individual which involved the collection and storage of 30 of each action for later processing. The testing was carried out in a normal office environment with no restriction on the user to swallowing or coughing. Also gentle head movement was permitted but sudden motions could be detected as false positives due to vibrations along the connecting cable and were kept to a minimum. These issues are to be addressed fully in the future with the introduction of interference rejection and a wireless earpiece.

B. Potential outcomes

There are a variety of potential outcomes (PO) which can occur when a TMEP action is intended or not intended. These are highlighted in Fig. 5 as a confusion matrix alongside corresponding explanations.



PO	Explanation
TP	Controlled TMEP signal correctly classified
TN	Un-intended TMEP signal classified as interference
FN-	User intention but no classification
FN+	Controlled TMEP signal incorrectly classified
FP	Interfering signal classified as controlled action

Fig. 5. Confusion matrix highlighting potential outcomes (PO) alongside associated explanations

The columns represent the intention (incoming/input) and the rows represent the classifier output. The input zero column represents the total number of interfering signals that were segmented and classified as action 1, 2 or 3, while the output zero row represents the total number of missed vectors which should have been classified as 1, 2 or 3

C. Empirical data

Results for one subject using a generic earpiece are presented in Tables II and III. Included are results of 4 test runs for the two classification schemes outlined earlier. Each run associated with a classifier uses the same training set as the corresponding run of the other classification scheme,

TABLE II
CONFUSION MATRIX FOR THE DECISION FUSION CLASSIFIER FOR ONE SUBJECT SUMMATED OVER 4 RUNS

m_D^*/c	**1**	**2**	**3**	**0**
1	88.43	0.93	1.67	4
2	1.37	93.02	0.42	6
3	9.72	4.65	97.91	4
0	0.46	1.40	0	0
total	216	215	240	14
Classifier Acc. = 93.12%				

TABLE III
CONFUSION MATRIX FOR THE DCT UNIVARIATE GAUSSIAN CLASSIFIER FOR ONE SUBJECT SUMMATED OVER 4 RUNS

m_D^*/c	**1**	**2**	**3**	**0**
1	95.87	1.84	2.14	3
2	1.65	97.24	0.85	2
3	0.46	0.92	95.72	1
0	2.07	0	1.28	0
total	242	217	234	6
Classifier Acc. = 96.28%				

implying that only 4 training sets were required for the 8 test sets. This allows for a more direct comparison to be made between the two classification schemes. The results are presented in confusion matrices (based on Fig. 5), showing the average accuracy of class c classified as class m across the 4 runs. Also included is the total number of signals classified and the average classification accuracy for the three actions.

The initial results show that the TMEP signals can be classified online accurately with an average classification accuracy of 96.28% for the DCT univariate gaussian classification and 93.12% for the DFC classification. For a more direct comparison between the classifiers, the FP and FN- values can be ignored, increasing the classification accuracies to 97.37% and 93.70%. Even though the DFC has a higher number of FP's and FN-'s, its classification accuracy increases by a smaller amount when these effects are ignored. The total number of FP's across the eight trials was 20 and the total number of FN-'s was only 12. The total time for all eight trials was 5893.28 seconds (≈100 minutes), implying the false positive rate was approximately one every 5 mins and a missed action approximately every 8 minutes.

IV. CASE STUDY: REAL-TIME SIMULATION AND CONTROL OF A WHEELCHAIR MOVING THROUGH A CONSTRAINED 2D ENVIRONMENT

To highlight the feasibility of this HMI integrated within an assistive system in real-time, a simulation has been designed and tested which allows for initial quantitative results in terms of misclassification of TMEP signals, false positives and missed tongue signals and how these relate to actual control of an assistive device. The scenario devised involves the control of a wheelchair moving through a constrained 2D environment, with the user having to navigate through a set of obstacles between a start and endpoint. The layout of the environment, shown in Fig. 6 is meant to mimic an average

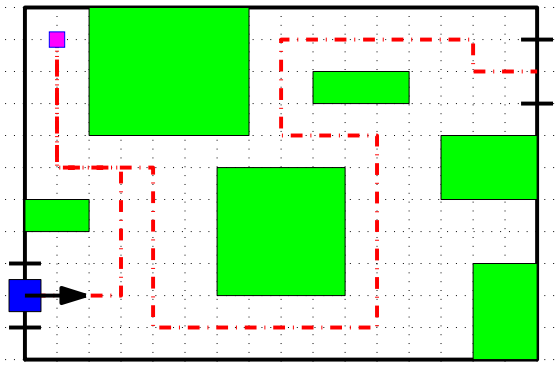


Fig. 6. Plan view of the constrained wheelchair simulation environment

TABLE IV
WHEELCHAIR SIMULATION RESULTS

Run	Classifier	Classification Accuracy				# Cols	Total # Moves	Total Time (s)
		1	2	3	Avg			
1	DFC	80	100	98.15	95.83	0	71	212.84
	DCT	100	100	100	100	0	67	199.76
2	DFC	100	72.73	92.98	91.14	2	79	226.51
	DCT	100	93.75	84.13	86.36	0	87	246.85
3	DFC	78.57	57.14	98.18	88.46	0	77	228.86
	DCT	90	100	100	98.53	0	67	202.75
4	DFC	100	100	88.33	91.34	0	77	219.02
	DCT	76.92	100	98.15	94.56	0	73	216.53

room topology, as seen from above. The optimal path of the wheelchair is shown with the red dotted line and involves the wheelchair moving to a way-point represented by the square in the top left of the figure. This can be considered as the user collecting an item e.g. a book, before leaving the room.

The chosen control scheme involves a top or down tongue motion (preference given to the user) to move the wheelchair forward one grid space and a left or right action to rotate the wheelchair by 90 in that direction. Thus the control scheme co-ordinate system is local to the wheelchair rather than global to the environment and means that only 3 actions need to be defined for this particular task. If a collision occurs it is recorded and the wheelchair is located back at the previous node. Table. IV presents results in terms of accuracy, number of collisions, total number of actions required and total time to complete the simulation run.

As can be seen from the table only two collisions occurred in total and was probably due to the topology of the environment, i.e. at specific locations a single misclassification would cause a collision. The DCT univariate Gaussian classifier had one perfect run and another near perfect run reinforcing it's previous results. Overall the DCT univariate Gaussian classifier outperformed the DFC in terms of classification accuracy, number of collisions, total number of moves and time to finish the course.

V. CONCLUSION

This paper has presented online implementation of a tongue based communication system, for use within an assistive technology setting. The detection, segmentation and classification processes have been outlined for realisation within an real-time human-machine interface. This includes the presentation of efficient algorithms for computation of the short-term energy of the signal and implementation of additional detection parameters for increased artifact rejection. A new algorithm is presented which combines the DCT for feature extraction as input into a univariate Gaussian maximum likelihood classifier alongside a 2-phase cross-validation procedure for feature selection and dimensionality reduction. This work will act as a platform for future study's that will be carried out in this area, allowing for a wider subject base to be gathered and tested.

Future work aims to unite this research with other work being carried out in the area of interference rejection. Current work on interference rejection utilises a wavelet packet transform to separate rhythmic bursting activity from sustained tonic activity (babble) and has helped to increase the classification rate when experiencing a decrease in signal-to-noise ratio due to external interferences [4], [5]. Also work is being carried out on internal artifact rejection (e.g. speech, coughing and other physiological signals), these signals can potentially pass through the initial detection block and would therefore otherwise be classified as a FP. Once the systems have been combined, it will allow for a rigorous test program to be devised to test for classification and misclassification rates, false positive and true negative rates, computational speed and robustness.

VI. ACKNOWLEDGEMENTS

The authors gratefully acknowledge Think-A-Move, Ltd. of Cleveland, OH, USA for their commercial research in this area. This work was supported by the UK Engineering and Physical Sciences Research Council (EPSRC).

REFERENCES

[1] R. Vaidyanathan, B. Chung, L. Gupta, H. Kook, S. Kota, and J. D. West, "Tongue-Movement Communication and Control Concept for Hands-Free Human-Machine Interfaces," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 37, no. 4, pp. 533–546, 2007.

[2] M. Mace, R. Vaidyanathan, S. Wang, and L. Gupta, "Tongue in cheek: A novel concept in assistive human-machine interface," *Journal of Assistive Technologies*, vol. 3, pp. 14–26, 2009.

[3] C. Salem and S. Zhai, "An isometric tongue pointing device," in *Proc. of the SIGCHI conf. on Human factors in computing systems*, (New York, USA), pp. 538–539, ACM Press, 1997.

[4] R. Vaidyanathan, S. Wang, and L. Gupta, "A wavelet denoising approach for signal action isolation in the ear canal," in *Proc. of the 30th Annual International Conf. of the IEEE EMBS*, pp. 2677–2680, IEEE, 2008.

[5] K. Mamun, M. Mace, M. Lutman, R. Vaidyanathan, and S. Wang, "Pattern classification of tongue movement ear pressure signal based on wavelet packet feature extraction," in *Proc. of 5th UK & RI Postgraduate Conf. in Biomedical Engineering & Medical Physics*, (Oxford), pp. 33–34, 2009.

[6] R. Vaidyanathan, M. P. Fargues, R. Serdar Kurcan, L. Gupta, S. Kota, R. D. Quinn, and D. Lin, "A Dual Mode Human-Robot Teleoperation Interface Based on Airflow in the Aural Cavity," *The International Journal of Robotics Research*, vol. 26, no. 11-12, pp. 1205–1223, 2007.

[7] L. R. Rabiner, "Algorithm for determining the endpoints of isolated utterances," *The Journal of the Acoustical Society of America*, vol. 56, no. S1, p. S31, 1974.

[8] K. Sayood, *Introduction to data compression*. San Diego: Morgan Kaufmann, 2nd ed., 2000.

[9] R. Kohavi, "A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection," in *Proc. of the Int. Joint Conf. on Artificial Intelligence (IJCAI)*, (Montreal), pp. 1137–1143, Morgan Kaufmann, 1995.