DESIGNED INTERFACE FOR USERS OF POWERED WHEELCHAIR WITH ARMS OR HANDS DEFICIENCY

Paulo Henrique de Araújo Madeira, paullo.h@pop.com.br
Franco Giuseppe Dedini, dedini@fem.unicamp.br
UNICAMP, R. Sérgio Buarque de Holanda, 251, FEM (Faculdade de engenharia mecânica), Laboratório de Projeto Mecânico

Abstract. *Analyzing the problems which wheelchair users where have any arms or hands deficiency and looking for more freedom on movements, this project will develop a user interface by a more intuitive commands as possible for powered wheelchair. This interface consists in two modules, one for voice recognization and another for navigation; all build in a microcontroller associated with external sensors and control drivers. The voice recognition is programmed to understand the more usual words like "frente", "trás" or another sound if the user has problems to speak, by a sequence of frequency combination used to build a captured frequency vector. This test sequence is compared with data vectors from a word library, programmed for each user, alloing interpretation and reaction to a programmed command. For a safe navigation control, sonar sensors can be used to alert or disable some directions of movement when an object is detected ahead, and allow obstacle avoidance by the user. This user interface will be implemented in a wheelchair motorization module; to avoid a replace from the user's actual wheelchair, resulting in a significant cost reduction.*

Keywords*: voice recognize, wheelchair, deficiency, interface, microcontroller*

1. INTRODUCTION

In cases of wheelchair users which another physical limitations that impossibility them to control a common joystick, like repetitive strain injuries, muscular dystrophy, handicapped and other cases, is needed to employ a interface sensible to another possible kinds of interaction by the user, like voice, eye position or sip and puff devices (may be uncomfortable).

A pricing study evidenced that systems using eye position, like the eye-gaze from LC Technologies (2007), may have an inaccessible price for the users, around 13'500 dollars. In this case the better solution looks to be a voice recognition system, because it is cheaper than an eye-gaze and more comfortable than a sip and puff device.

2. VOICE INTERFACE

Voice can be described like a combination of frequencies and amplitude to build phonemes and, consecutively, words. In a natural speech, these worlds may have some phonemes omitted, by the contraction of syllables, or changes in frequency or amplitude, according with the speaker habits and humor.

A voice recognition system needs to identify these phonemes, and consecutively processes then to recognize the words where spoke to finally processes this utterance (or either, a set formed by word and function).

There are four levels of sophistication in recognition process of utterance described by Cook (2002):

- Isolated words: each utterance is recognized isolated; but they need a small pause time between then, to slice the data.
- Connected words: is the same as isolated words, but the utterances don't need a pause time between then.
- Continuous speech: is the next step, allowing a continuous speech with more than one utterance at the same time, likes computer dictation.
- Spontaneous speech: the system is capable to identify a continuous speech, and some natural sounds from the user in the words like "ums", "ans".

For this voice recognition approach, two architectures are used: the first uses voice recognition software, and the second is implemented on a microcontroller. An isolated words model is used to simplify the recognition algorithm and reduce hardware requirements when the microcontroller is used.

Is important to remember about each user has your own habits when speak, and the used software must to be trained before to adapt and understand the utterances.

2.1. Using an Computer and Voice Recognition Software

The first phase of this project was done with focus on develop a simple and intuitive voice controlled interface for wheelchair users, and define the better syntax for the utterances to be used.

If too many kinds off utterance are used, the user will be confused when try to remember the right one for the desired command, and the most functions probably will never be used. In this project, with focus on wheelchair control, only six utterances are used: “Liga” (turn on), “Desliga” (turn off), “Frente” (go ahead), “Trás” (go back), “Esquerda” (turn left), “Direita” (turn right), and “Testando” (testing, used only in developing phase, meaning no action).

Before to implement utterances in a real motorized wheelchair, this first prototype was developed to study how the relations between user and interface are transparent, and search for new applications. Its architecture consists in tree elements: input interfaces (microphone or keyboard), processing elements (microcomputer and software) and system to control (simulated by microcontroller and led’s).

Using a keyboard or dictation software, like IBM Via Voice®, is possible to acquire the utterances in a textual form. These utterances allow a dedicated text interpreter software do the interface with a microcontroller, allowing a simulation between commands and utterances.

If necessary, a joystick can be placed associated directly with the microcontroller to substitute the control from text interpreter in emergency cases, like illustrated in block diagram of Fig. 1.

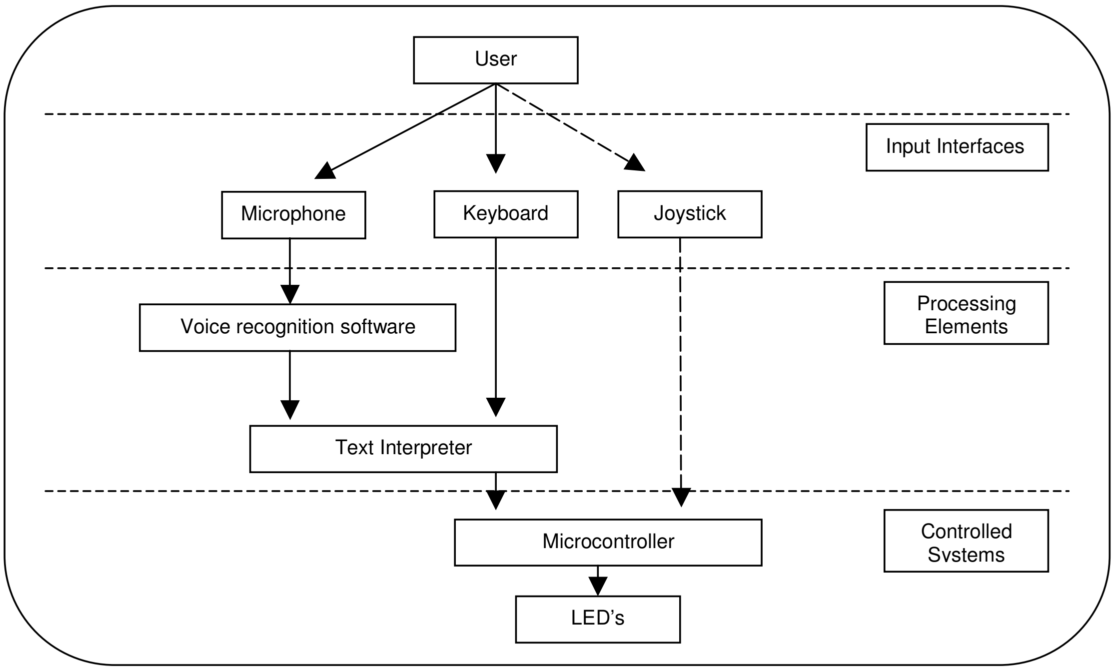

Figure 1. Block diagram of computer interface system

This prototype results allow finding other applications for voice recognition systems, like lamps control in residential automation, household-electric control and open and close doors. Although for mobile systems, like wheelchairs, there are problems associated with the large size and power consumption from computer.

Another problem observed is the need to buy voice recognition software and laptop computer, increasing the complexity and costs, becoming this solution cheaper than an eye-gaze, but however still inaccessible for the majority of the users.

The solution found is to develop a simpler system, removing the computer and voice recognition software by placing these functions in microcontroller firmware.

2.2. Using an Microcontroller and Voice Recognition Firmware

The expected results for this new implementation are a cheaper, small size and significant reduction on power consumption. Although the hardware used (microcontroller) is very resource limited if compared with a laptop computer, resulting in constrains for voice recognition firmware.

This second prototype consists in a microphone, amplifier, microcontroller and drivers for motor control, like showed in Fig. 2.

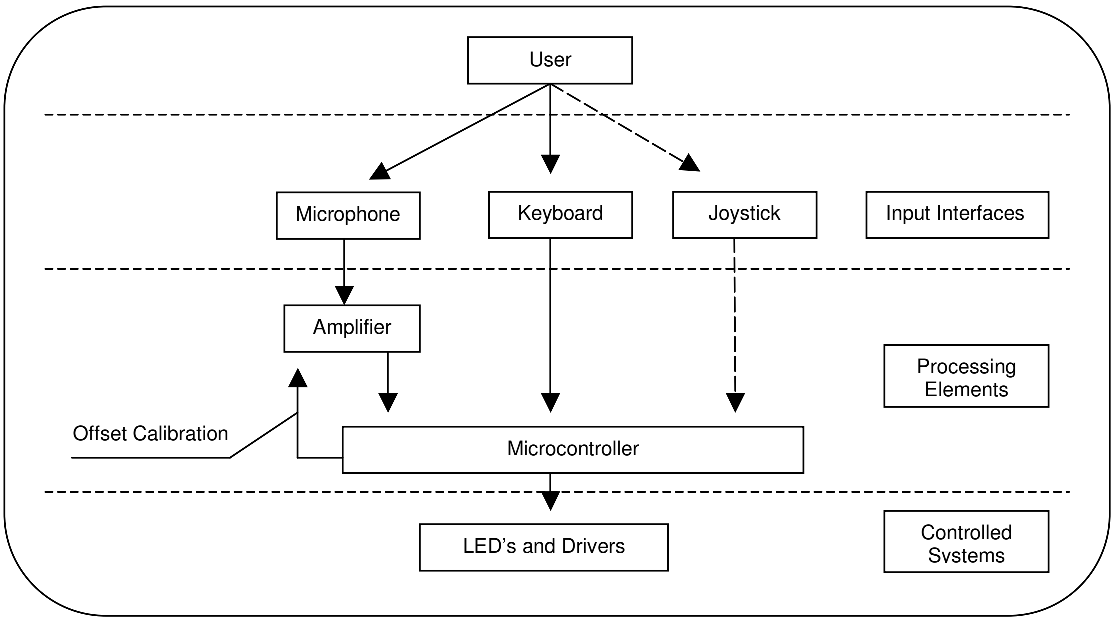

Figure 2: Block diagram of microcontroller interface system

To interface a low costs IBM -PC microphone with the microcontroller, an amplifier must be used following the Sound Blaster protocol, described by Engdahl (2000). The amplifier uses a LM741 operational amplifier, with a constant gain, and offset defined by the PWM from microcontroller by an automatic calibration cycle. This circuit is showed in Fig. 3:

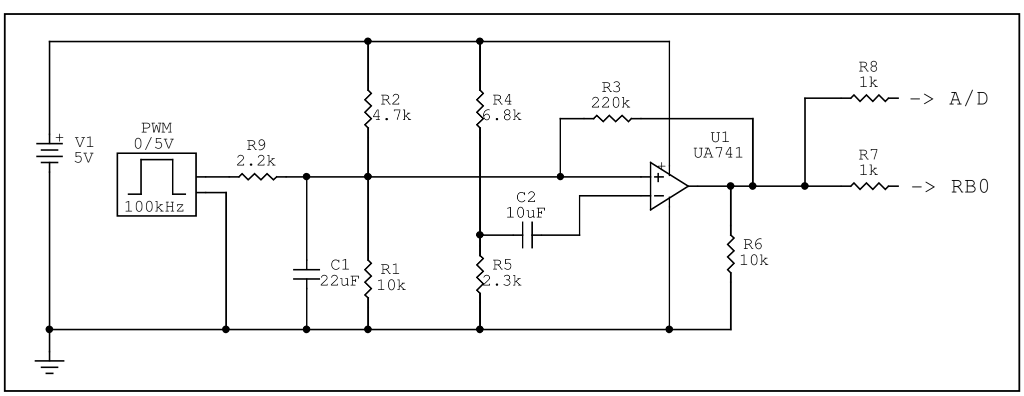

Figure 3: Amplifier Circuit

In this calibration cycle, the microcontroller uses eight mean value captured from the A/D output from amplifier and increase or decrease the PWM duty cycle, allowing a fine tuned offset.

To process the voice signal, a real time frequency capture algorithm is used. This algorithm captures the number of raising edges (2.0 volts up to 2.5 volts) from the microphone-amplified signal in eight cycles of 100 milliseconds to build the captured frequency vector. This captured frequency vector is send for the interpretation module.

The interpretation module has two sub-modules: one to build a word library and a second to compare the captured words with the generated word library.

World library consists in a group up to eight words, where each is represented by one frequency vector. These frequency vectors are each word or any sound that can be easily repeated by the user, formed by eight frequencies and his respective command.

To generate a library vector is necessary to repeat about ten times the same word, allowing a reliable representation for each frequency captured, like showed in Fig. 4 by the white squares. In Fig. 4 is possible to see about each word having a unique representative frequency vector by the comparison of these vectors. These library vectors are stored in EEPROM microcontroller memory to avoid any loss of data in any power down or turn off situation.

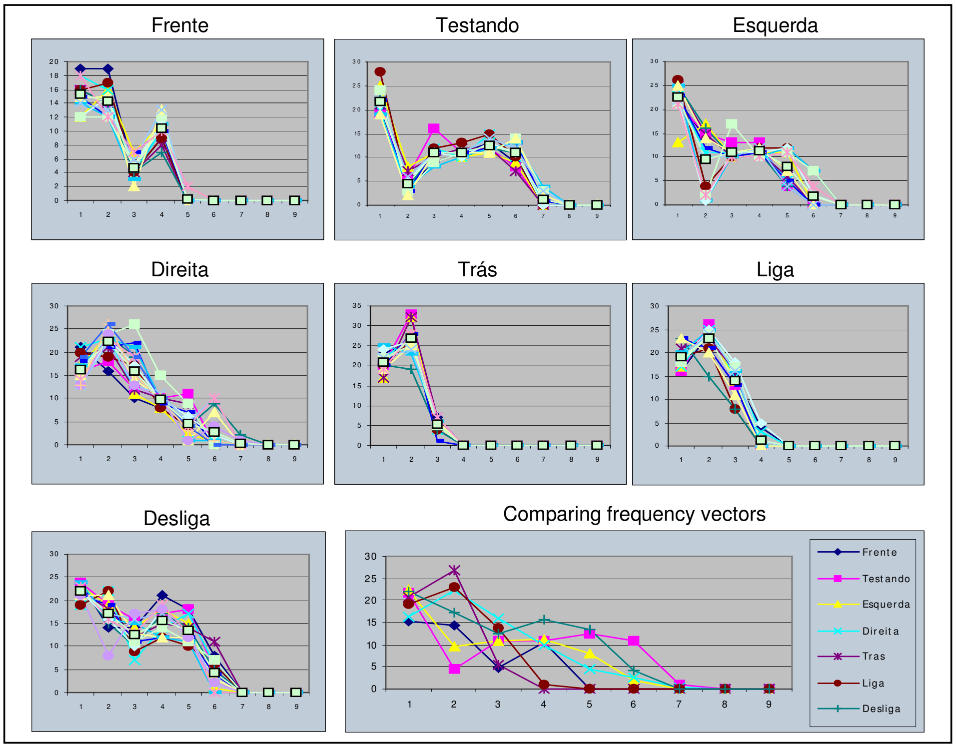

Figure 4: Frequency vector and associated word

Legend (for all graphs):

Y axis: Number of rising edges

X axis: Period Interval

Line with white squares: means the representative captured frequency vector

The word library builder has a user interface firmware that needs a PC with RS232 interface software to program new words in library and to provide the graphical interface and keyboard input, after this phase, computer is no longer necessary.

After all needed words and respective associated command was recorded, the second module can interpret these utterances by comparing all received frequency vector with the library built in microcontroller EEPROM.

This comparison starts after any new capture of vector words occur, analyzing if there are any similar vectors in the library. This similarity study is done by subtracting the new acquired frequency vector from each library vector (reference), resulting in a difference vector for each comparison. All values in each difference vector is converted to module and summed to obtain the total dispersion.

Each comparison results in a different dispersion and, for this configuration of sampling time, dispersions below the value 15 is assumed as match. If two or more library words match with the captured vector, the word with the small dispersion is used as utterance.

The Fig. 5 illustrates a comparison event graphically, when a captured frequency vector match with the utterance “frente”. All points that describe the captured vector are closer than the library vector with represents “frente” and few points are closer than the library vector that represents “esquerda”.

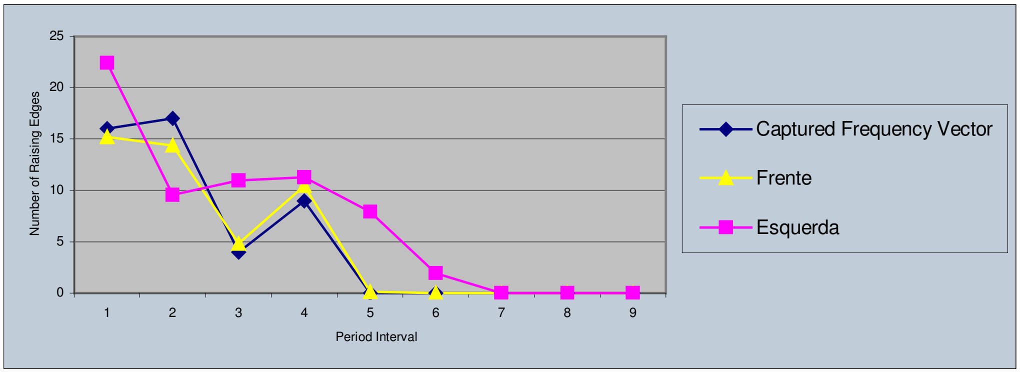

Figure 5: Analyzing a captured frequency vector

The advantage of this algorithm is his low memory and processor time consumption, allowing an implementation in a mid range microcontroller, like PIC 16F877, and a significant reduction in costs, weight and power consumption. This solution trades off reduction in accuracy and word library size, if compared with any PC commercial voice recognition software, but has a significant size, power consumption and costs reduction.

3. USING A HIERARCHICAL CONTROL LOGIC FOR INPUT MANAGEMENT

When different kinds of interfaces and inputs are used, a consistent integration is necessary between them. The integration system could use a hierarchical logic, meaning that are some inputs predominant than others when used, and stays in an idle condition until an event occurs.

Hierarchical logic can be used for input management or obstacle avoidance, when an unsafe command is performed, like to drive closest of a wall, an alert by sonar sensors about an unsafe condition will be generated and the navigation system will disable some movement’s directions.

This hierarchical control is defined by placing the easier interfaces for the user (less robust) in a lower priority level and the more robust interface in a high priority level, like Tab. 1.

Table 1. Interfaces priority in a hierarchical control

Interface	**Kind of Event**	**Priority**
Sonar	Obstacle detected	1 – High
Joystick	Move out of zero position	2
Keyboard utterance	New word recognized	3
Voice utterance	New utterance recognized	4 – Low

To evaluate this logic, a prototype was built in reduced scale, but conserving all sensors and control architecture characteristics. In this prototype, hierarchical logic could be used for multiple inputs management by choosing the right input by his priority and event occurrence to control the movement output. For example: if an obstacle ahead alert event occurs, all forward movement from voice, keyboard and joystick will be disabled until there are no objects ahead.

The reliability from interface systems in the first priority levels has a great contribution in all system performance. Tests realized with the prototype shows about if the sonar implemented is too susceptible to cross reference signals, like another sound noise interference, the system will not be able to move in the direction protected by this sonar, or if a joystick have a poor regulation, noise movements will be generated and utterances by voice or keyboard will be disabled until the joystick zero position is detected.

4. ANOTHER INTERFACE METHODS

Is possible to describe more solutions for interfaces, as example a virtual keyboard, from Microsoft (2006), allows a one input digitations. This virtual keyboard continuous scans through lines (Y axis) until an input is detected, then scans through the letters and commands (X axis of selected line) and execute the command or character when a second input is detected. The scan speed could be adjusted allowing trades off by digitations speed and easiness to select a desired character or line.

In wheelchair interfaces, virtual keyboards could be made with only the needed inputs, like forward, left, right and back, allowing a simpler and quicker interface. The input could be a simple push button or a sip and puff device, like the used by Speropoulos and Townson (2001) for a wheelchair control.

An improved switching interface is presented by Felzer (2002), in this interface is possible to scan the options actively by commands from one or double muscle contractions, like mouse clicks. This interface allows to control a conputer or a whellchair in less time than traditional switching interfaces.

Sup and puff devices can be used for household-electric turn-on turn-off controls, like the “Medical Sip and Puff Device” developed by Cunningham (2006).

For users where lost the movements of superior and inferior members, and needs to do simple tasks like a small text digitations or interact with interfaces by buttons, a wand fixed in the head is a simplest solution but a poor aesthetical concept.

A less intrusive interface concept developed by Bergasa *et al.* (1999) detects head position and expression by image recognition to control a wheelchair using a microcomputer processing. A simpler implementation of the same concept by Adremo (2006) uses an adjustable head support to detect head position and control wheelchair movements.

5. APLICATIONS

The presented sound and words recognition approach allows to user send commands by words or other pre-memorized sounds, avoiding the constant need of a carer to do simple tasks like to movement the wheelchair.

With more freedom in movements, the user will be able to have a better social relationship, like going to school or accompanying his friends. A review into another interface approaches has showed about other applications for this interface beyond the wheelchair control, like Eagle Eyes project, user modeling techniques and Pebbles project.

5.1. Eagle Eyes Project and Initial Learning Phases

In cases where the impairment levels disallow to use traditional interfaces, like when the user has only facial movements, the Eagle Eyes interface presented by Gips (1998) can be used. This interface uses five electrodes placed on user’s head to acquire electro-oculographic (EOG) potentials and make an interface with a computer.

To acquire the EOG potential, four electrodes are placed in two pairs and a fifth for voltage referential. The horizontal pair acquires horizontal eye position and vertical pair acquires the vertical eye position.

The time spent for learning to use this interface varies from 15 minutes up to months, depending on user characteristics. Gips (1998) describes about “children who are completely paralyzed from birth are not used to physically controlling anything, much less the cursor on the screen with their eyes” in these cases the time to learn will be increased.

In these cases, a vocalized interface could be used allowing a more user friendly interface on first learning phases. The verbalization of the commands could be easier than an eye interface, and can allow interaction with computer software for reading, writing and other applications.

5.2. Obtaining User Model for Adequate Interface Design

Knowing user model for timing response is necessary before to develop an adequate interface for him. Each user has his own timing response and if he has any disabilities these times tends to increase. An adequate user timing response model will be useful to determine scanning time and compatible wheelchair speeds.

An approach to obtain these timing models is presented by Keates *et. al.* (1999) when adaptations to focus users with disabilities on methodology proposed by Card *et. al.* (1983) is done. The methodology of Card *et. al.* (1983) proposes two user timing modeling techniques: Model of Human Processor (MHP) and Keystroke Level Model (KLM).

These two techniques consider the response time as the time between the perception and reaction to an event, differing in how the response time can be subdivided and measured.

The Model of Human Processor (MHP) methodology subdivides the response time in a sum of three events: perception, cognition and motor (movement or action). Each one of these three events is multiplied by a constant associated with how many times they occur. The times for the events are defined by empirical experiments with the user.

The Keystroke Level Model (KLM) methodology subdivides the response time in a sum of six events: press the bottom, point, route, drag, mental operations and response. This methodology is used in more complex cases, when there are more operations involved on response time.

These two user models, with are proposed by Card *et. al.* (1983), allow understand how the user thinks and provides a standard to choose the best interface for each user, based on his response times. The vocal interface presented on this paper can be evaluated on MHP model like a motor event to be compared with another interfaces, like scanning associated with sip and puff devices, for the same user to define where interface fits better for each case of disability.

By the method of MHP, proposed by Card *et. al.* (1983), and the vocalized interface, proposed in this paper, is possible to discover where the user spends more time (perception, cognition or motor) and then make improvements like changing the feedback interface for the user (vision), simplifying the interface (cognition) or reducing the size of the words (motor timing).

Educational aspects are associated with modeling the user abilities and posterior training of the user weak points to improve a better response time.

5.3. Bringing the School for the User

Estimative showed by Pebbles project (2007) about long term hospitalized children inform about 50% of then have difficulties to readapt and return to school, and finish his course on suitable period. Traditional methods in these cases consist on deployment of a teacher on the hospital, but unfortunately there is no acquaintance with school friends.

The solution developed by Pebbles project (2007) consists to deploy a system of videoconference between the hospitalized child and his school class. This technique differs from videoconference, in an aspect intended as video presence.

In video presence, described by UCSF Children's Hospital (2007), the user is represented by video and camera on school in a structure that looks like the child, to simulate his physical presence and allow a face to face interaction. To control the movements and zoom from the camera and video on class, the child uses a computer's joystick on hospital.

Since in higher disabilities cases, which could be temporary or permanent, the child may have difficulties or impossibility to operate a computer joystick, then a voice interface should be used. If the child has no higher impairment levels, this voice interface should help him to operate the camera by hands free, while holds or write on a notebook.

Allowing a better integration between school and hospitalized child will motivate the educational processes and social relationships.

6. CONCLUSIONS

This new voice recognition interface with microcontroller allow a significant cost reduction and mobility increase, because the computer is only used as a graphical interface during the programming phase of the library of frequency vectors.

However, given limited architecture of the microcontroller if compared with computers, there are some problems to recognize the utterances, specially when the user speak a word in a different period from the word in database, resulting in a mismatch from the frequency vectors. New filtering techniques in the amplifier, like band pass filters; and algorithms improvements, like recording in analog mode for post processing, will allow a more efficient utterance processing.

Since hierarchical control is too dependent of the reliability from the higher priority inputs, other interfaces more liable could be used in these higher levels, like head position detection or sip and puff devices to secure a good command interpretation.

Development of user models for interfaces must to concern about disabled people and his especial characteristics, like differences on timing responses. Learning about these response times will help to improve traditional interfaces or build new ones designed for these user needs.

In another point of view the study of user response times could help to understand more about the effects of each disability. This knowledge about the user disabilities and educational theories could help to find ways to improve each weakness of the user on the three basic fields: vision, cognition and motor.

This approach for vocalized interface will allow new kinds of human machine interaction, helping disabled children and adults in social integration and respective learning by classroom and acquaintance.

7. REFERENCES

Adremo wheelchair, 2006. "Adremo wheelchair". 9 March 2007. <http://www.adremo.nl/english/index.php>

Bergasa., L.M., Mazo, M., Gardel, A., Sotelo, M.A., García, J.C., 1999. "Guidance of a Wheelchair for Handicapped People by Head Movements". 12 March 2007. <http://ieeexplore.ieee.org/iel5/6597/17656/00815344.pdf>

Card, S. K., Moran, T. P., Newell A., 1983, “The Psychology of Human-Computer Interaction, Lawrence Erlbaum Associates”.
Cook, S., 2002. “Speech Recognition HOWTO”. 20 January 2007 <http://www.ibiblio.org/pub/Linux/docs/HOWTO/other-formats/pdf/Speech-Recognition-HOWTO.pdf>
Cunningham, S., 2006. “Medical Sip and Puff Device”. 15 April 2007 <http://www.parallax.com/html_pages/resources/custapps/app_medical_sip_puff_device.asp>.
Engdahl, T. 2000. “Powering microphones”. 5 December 2007 <http://www.epanorama.net/circuits/microphone_powering.html>
Felzer, T., Freisleben, B., 2002, “HaWCoS: The "Hands-free" Wheelchair Control System”, Scotland, Edinburgh, 8-10
Gips, J., 1998, “On Building Intelligence into EagleEyes”, Computer Science Department Boston College Chestnut Hill, 50 - 58, 18 april 2007, <http://www.springerlink.com/index/LLC1PMYUCQ4NHVTJ.pdf>
Keates, S., Clarkson, P. J., Robinson, P., mai. 1999, “Developing a Methodology for the Design of Accessible Interfaces”, University of Cambridge
LC Technologies, 2007. “Eyegaze Communication System”. 23 March 2007. <http://www.eyegaze.com/2Products/Disability/DisPricing.htm>
Microsoft, 2006. “Types of Assistive Technology Products”. 23 January 2007, <http://www.microsoft.com/enable/at/types.aspx>.
Speropoulos, C. A., Townson, A. F., 2001, “Assistive Technology Assessments in a Tertiary-care Rehabilitation Facility”, BC Medical Journal Volume 43, Number 2, March, 82-85. <http://www.bcma.org/public/bc_medical_journal/BCMJ/2001/march_2001/AssistiveTechnology.asp>.

8. RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.