We are IntechOpen,
the world's leading publisher of
Open Access books
Built by scientists, for scientists

6,900
Open access books available

185,000
International authors and editors

200M
Downloads

154
Countries delivered to

Our authors are among the

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS
BOOK CITATION INDEX
INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Wheelchair and Virtual Environment Trainer by Intelligent Control

Pedro Ponce, Arturo Molina and Rafael Mendoza

Additional information is available at the end of the chapter

1. Introduction

There are many kinds of diseases and injuries that produce mobility problems. The people affected with any disability must deal with a new lifestyle, specifically people with tetraplegia. According to ICF [1], people with tetraplegia have damages associated to the power of muscles of all limbs, tone of muscles of all limbs, resistance of all muscles of the body and endurance of all muscles of the body. The main objective of this project was to help disabled people to move any member of their body, although this wheelchair can be used for persons with mobility problems, doctors should not recommend it to all patients because it reduces muscle movement, which could lead to muscular dystrophy.

Currently, there is not an efficient system that covers the different needs that a person with quadriplegia could have. Their mobility is reduced by physical injury and, depending on the extent of the damage nursing and family assistance is required. Even though many platforms have been developed to address this problem, there is not an integrated system that allows the patient to move autonomously from one place to another, thus limiting the patient to remain at rest all the time. In previous research projects completed in Canada and in the United States, such as wheelchairs controlled with the tongue [2] and a wheelchair controlled with head and shoulder movements [3]. Those systems provide mobility for the person with any injury in functions related to muscle strength. This work offers a different alternative for the patient and aims to build an autonomous wheelchair that can afford enough motion capacity to transport a person with quadriplegia. Different kinds of controls are provided, so the trajectories required by the patient must be controlled using ocular movements or voice commands, among others.

An existing brand of electric wheelchair was used (the commercial Quickie wheelchair model P222 [4] with a Qtronix controller).

2. Eye movement control system

2.1. General description

The eye control is based on the magnetic dipole generated by the eye movement; therefore a voltage signal is produced allowing us to sense these voltages using clinical electrodes. Those signals, microvolts, come with noise. A biomedical differential amplifier was used to sense the desired signal in the first electronic stage and simple amplification in the second stage. The signals are digitalized and acquired into the computer in the range of volts via data acquisition hardware for further manipulation. Once the signal is filtered and normalized, the main program based on artificial neural networks learns the signals for each eye movement. This allows us to classify the signal, so it can be compared against the next signals acquired. In this manner, the system could detect which kind of movement was made and assign a direction command to the wheelchair.

2.2. Physiological facts

There is a magnetic dipole starting between the retina and cornea that generates differences of voltage around the eye. This voltage ranges from 15 to 200 microvolts depending on the person. The voltage signals also contain noise with the fundamental of a base frequency between 3 and 6 hertz. This voltage can be plotted over time to obtain an electro-oculogram (EOG) [7] which describes the eye movement. Fig. 1 shows a person with the electrodes.

Figure 1. Patient using the EOG and training the signal recognition system.

Prior to digitalizing the signals, an analogical stage of amplifiers, divided in two basic parts, was used. The first p is a differential amplifier AD620 for biomedical applications, a gain of 1000x was calculated using the equation (1):

$$R_G = \frac{49.4k\Omega}{G-1} \tag{1}$$

Where G is the gain of the component and R_G is the resistance. The digitalization of the amplified signal is carried out with the National Instruments DAQ [8] data acquisition hardware.

2.3. EOG LabVIEW Code

This section will present an overview of the program implemented for the EOG Signal acquisition and filtering stages. Fig. 2 shows three icons, the first one (data) represents a local variable which receives the values from the LabVIEW utility used to connect the computer to a DAQ. The second icon represents the filter which is configured as written in part B. The third icon is an output that will display a chart in the front panel in which the user can see the filtered signal in real time.
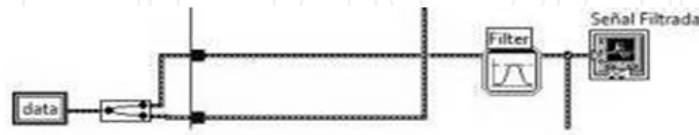

Figure 2. Signal filter

After the signal was filtered, it was to be divided into frames with 400 samples each. This stage is very important because the signals are not all the same length, thus by using the framing process all the signals will have the same length.

Once the length of our data arrays was normalized, the amplitude of the signal had to be normalized as well. After all the aforementioned stages, there will be six normalized arrays, which are then connected to their Neural Network, as shown in Figure 3.
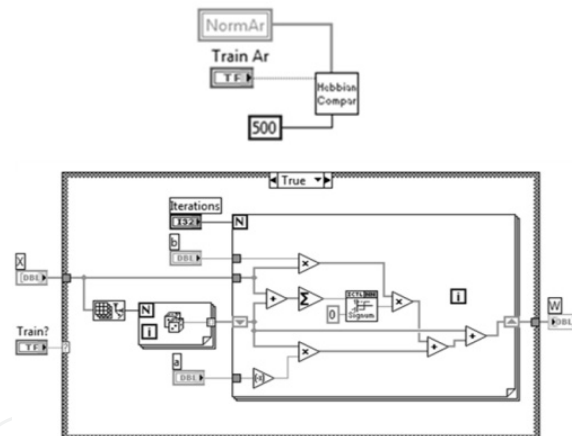

Figure 3. Training System LabVIEW Code

In Figure 3, it can be seen that this section will receive the variables and give the calculated error. In the case of the recognition signals where two different kinds of neural networks, trigonometric neural networks and Hebbian neural networks were tested, both of them gave good results. Figure 3 shows the configuration for the Hebbian Learning. The diagram of the configuration of this kind of network is shown inside the block diagram of Figure 3.

This network will take the point from the incoming signal and will make an approximation of this value and save it into W. If train is false the network will only return the values already saved in W; by doing this "Hebbian Comparison" will calculate the error between the incoming signal and W.

The Front Panel on LabVIEW shown in Figure 4 is the screen that the user will see when training the wheelchair. On the upper half of the screen are all the charts of the EOG (Figure 5) and the upper two charts refer to the filtered signals of both the vertical and the horizontal channels. The lower two charts refer to the length normalized signals. The right-hand chart will show the detected signal. On the lower half of the screen (Figure 6), the user will see all the controls needed to train the system. A main training button will activate and set all the systems in training mode. Then the user will select which signal will be trained. Doing so will open the connection to let only that signal through for its training. Once recognized, the signal will appear in the biggest chart and the user will push the corresponding switch to train the Neural Network. When the user has trained all the movements, the program needs to be set in comparison mode, for this case the user must deactivate the principal training button and put the selector on Blink (Parpadeo). Then the system is ready to receive signals. To avoid problems with natural eye movements, the chair was programmed to be commanded with codes. To get the program into motion mode the user must blink twice, which is why the selector was put on Parpadeo, so that the first signals that go through the system could only be Blink signals. After the system recognizes two Blink signals the chair is ready to receive any other signal. By looking up the chair will move forward, looking left or right will make the chair turn accordingly. The system will stay in motion mode until the user looks down; this command will stop the chair and reset the program to wait for the two blinks. Embedding this code into a higher level of the program will allow the EOG system to communicate with thecontrol program that will receive the Boolean variables for each eye movement direction. Depending on the Boolean variable received on true, the control program will command the chair to move in that direction.
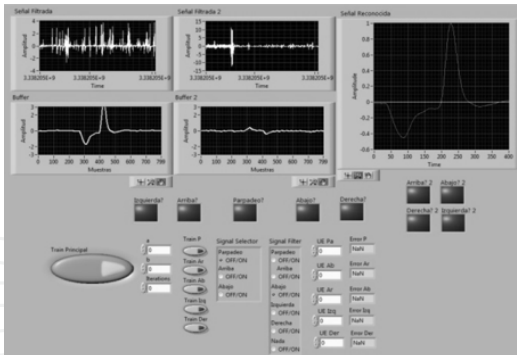

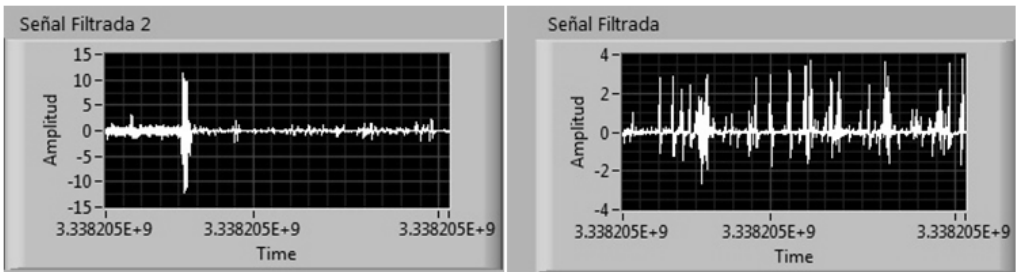**Figure 4.** Figure 4. Frontal panel for training signal

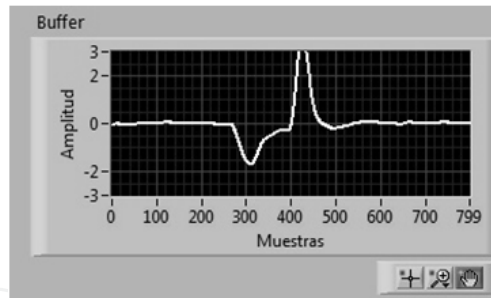**Figure 5.** Filtered signal for vertical and horizontal channels

Figure 6. Eye movement

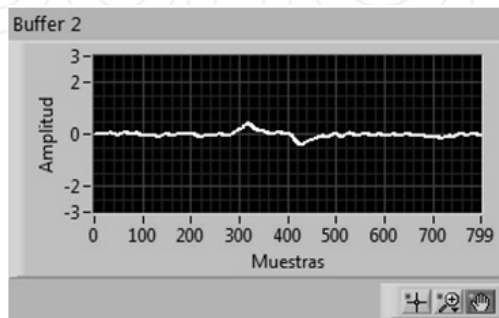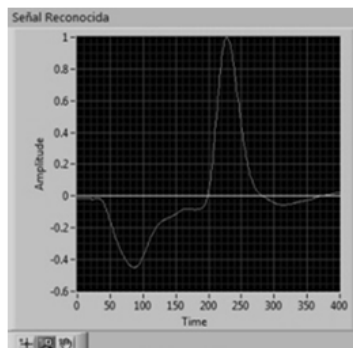**Figure 7.** Analog received input from eye movements

Figure 8. Signal generated when the eye moves up

3. Voice control

The wheelchair in this study is intended to be used by quadriplegic patients; it was programmed to add a voice message system to the chair. This system, will also allow the user to send pre-recorded voice messages by means of the EOG system, as another way to assist the patient. Figure 9 shows the main LabVIEW code.

3.1. EOG and voice message system coupling

In this section of the project, all the EOG system and programming remained the same as in the previous direction control system. But instead of coupling the EOG system to the motor control program, it was coupled it to a very simple program that allows the computer to play pre-recorded messages, such as: I am hungry, I am tired, etc. The messages can be recorded to meet the patients' needs to aid them in their communication with their

environment. The EOG program will return a Boolean variable which will then select the message corresponding to the eye movement chosen by the user. The selection will search for the path of the saved pre-recorded message and will then play it.
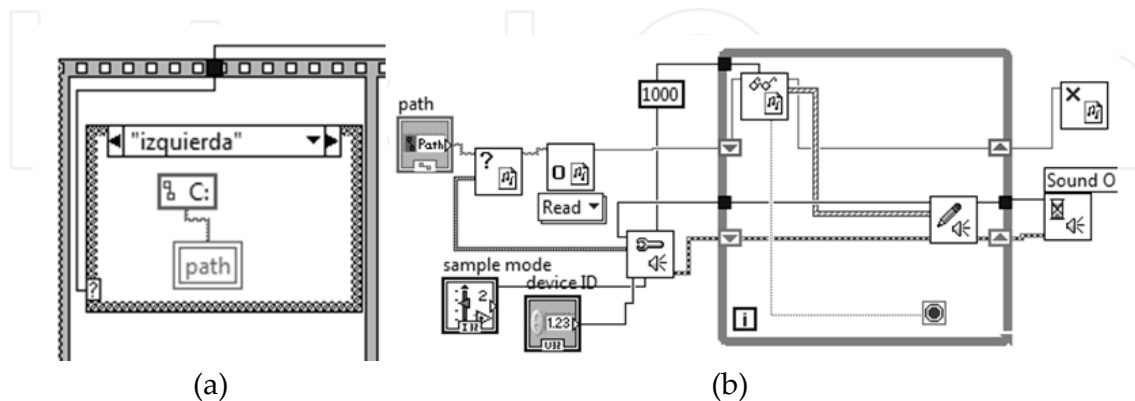

(a)   (b)

Figure 9. a.Shows the activated Boolean received into a case structure that selects the path used for opening each file. Figure 9.b.Sound playback.

The second stage shown on Figure.9.b of the structure works by opening a *.wav file, checking for errors and preparing the file to be reproduced, a while structure is used to play the message until the end of the file is reached, then the file is closed and the sequence is over.

3.2. Voice commands

For patients with less severe motion problems, it was decided to implement a Voice Command system. This allows the user to tell the chair in which direction they want to move.

3.3. Basic program

For this section two separate programs were used, Windows Speech Recognition and Speech Test 8.5 by Leo Cordaro (NI DOC-4477). The Speech Test program allows us to modify the phrases that Windows Speech Recognition will recognize. By doing so and coupling Speech Test to our control system, it is possible to control the chair with voice commands.

The input phrases can be modified by accessing the Speech Test 8.5 Vi. By selecting speech (selection box) the connection between both programs can be activated. Then Speech Test 8.5 will receive the variable from Speech Recognition and by connecting it to our control system it is possible to receive the same variable and control the chair.

At first, the user must train the Windows Speech Recognition. This is strongly recommended, because although it can differentiate different people's voices, the system

fails continuously if many people use the same trained configuration. On the other hand, the different tests that were performed in a closed space without any source of noise were 100% satisfactory.

This system runs in the same way as the EOG, by saying to the chair *derecho*, the chair will start moving. Saying the words *derecha* or *izquierda* will turn the chair right or left and by saying *atrás* the chair will stop. The Boolean variable is received into our control system the same way as in the EOG.

4. Electric wheelchair navigation system

After the user sends the reference command by voice or eye movement, the electric wheelchair uses fuzzy logic and neural networks for taking over the complete electric wheelchair navigation system.

For transferring the vague fuzzy form of human reasoning to mathematical systems a fuzzy logic system is applied.

The use of IF-THEN rules in fuzzy systems gives us the possibility to easily understand the information modeled by the system. In most of the fuzzy systems the knowledge is obtained from human experts.

Artificial neural networks can learn from experience, but most of the topologies do not allow us to understand the information learnt by the networks. ANN's are incorporated into fuzzy systems to form Neuro-Fuzzy systems which can acquire knowledge automatically by learning algorithms of neural networks. Neuro-Fuzzy systems have the advantage over Fuzzy systems that the acquired knowledge is easy to understand -more meaningful- to humans. Another technique used with Neuro-Fuzzy systems is clustering, which is usually employed to initialize unknown parameters such as the number of fuzzy rules or the number of membership functions for the premise part of the rules. They are also used to create dynamic systems and update the parameters of the system.

4.1. The neuro-fuzzy controller

The position of the wheelchair is taking over by the Neuro- Fuzzy controller, thus it will avoid crashing against static and dynamic obstacles.

The controller takes information from three ultrasonic sensors which measure the distance from the chair to an obstacle located in different positions of the wheelchair, as shown in Figure 10.

The outputs of the Neuro-Fuzzy controller were the voltages sent to a system that generates a PWM to move the electric motors and the directions in which the wheel will turn. The controller is based on trigonometric neural networks and fuzzy cluster means. It follows a Takagi-Sugeno inference method but instead of using polynomials on the defuzzification process it also uses trigonometric neural networks (T-ANNs). The diagram of the neuro-fuzzy controller is shown in Figure 11.
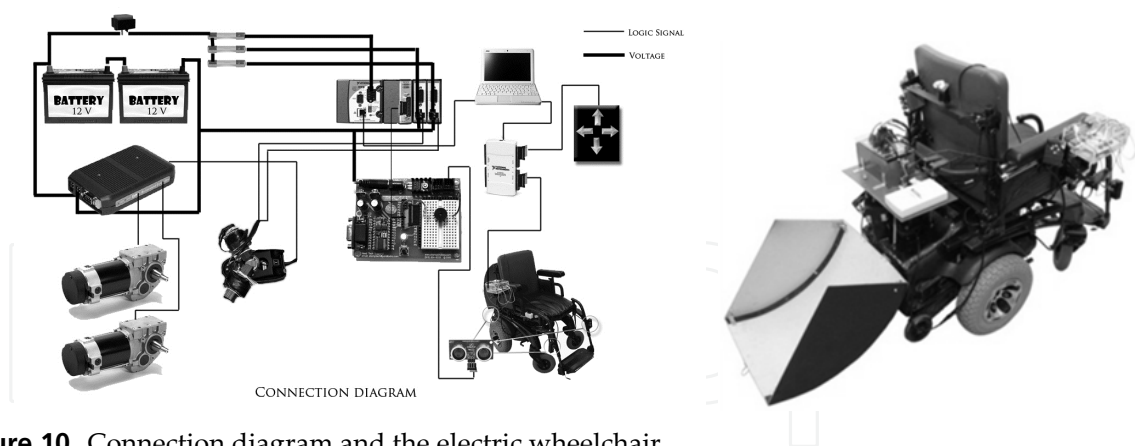
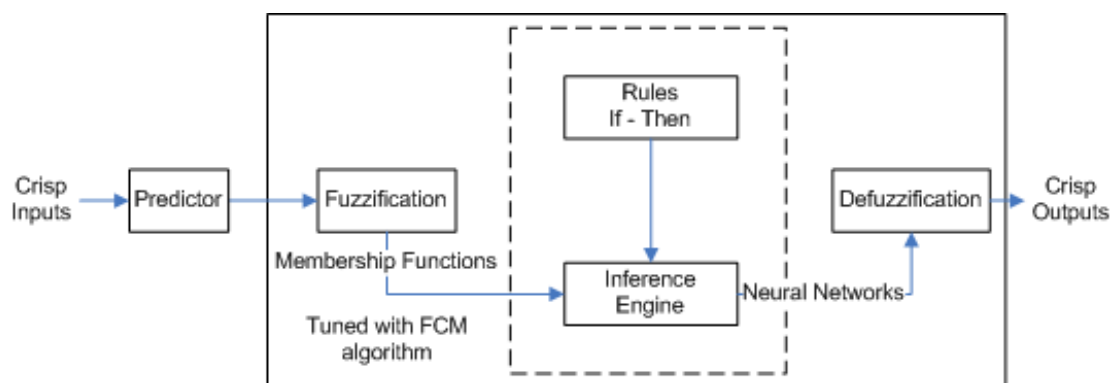**Figure 10.** Connection diagram and the electric wheelchair

Figure 11. Basic diagram of the Neuro-Fuzzy controller

Theory of Trigonometric Neural Networks

If the function $f(x)$ is periodic and integrable in Lebesgue (continuous and periodic functions (2π) in $[-\pi,\pi]$ or $[0,2\pi]$). It must be written as $f \in C*[-\pi,\pi]$ or just $f \in C*$. The deviation -error- of $f \in C*$ from the Fourier series at the   point or from a trigonometric polynomial of order $\leq n$.

$$E_n(f) = \min_{\tau_n} \qquad (2)$$

$$\max|f(x) - \tau_n(x)| = \min\|f - \tau_n\| \qquad$$

$$0 \leq x \leq 2\pi \qquad (3)$$

Using Favard sums of   falling in its extreme basic property, give the best approximation for trigonometric polynomials of a class (periodic continuous functions) as follows:

$$\|f'\| = \max_x |f'(x)| \leq 1 \qquad (4)$$

Fourier series have been proven to be able to model any periodical signal in [2]. For any given signal $f(x)$ it is said to be periodic if $f(x) = f(x+T)$ where T is the fundamental period of the signal. The signal can be modeled using Fourier series:

$$f(x) \sim \frac{a_0}{2} + \sum_{n=1}^{\infty} \left(a_n \cos(nx) + b_n \sin(nx) \right) = \sum_{n=1}^{\infty} A_k(x) \tag{5}$$

$$a_0 = \frac{1}{T} \int_0^T f(x) dx \tag{6}$$

$$a_n = \frac{1}{T} \int_0^T f(x) \cos(n\omega x) dx \tag{7}$$

$$b_n = \frac{1}{T} \int_0^T f(x) \sin(n\omega x) dx \tag{8}$$

The trigonometric Fourier series consists of the sum of functions multiplied by a coefficient plus a constant, so a neural network can be built based on the previous equations.

The advantages of these neural networks are that the weights of the network can be computed using analytical methods as a linear equation system. The error on the solution decreases when the number of neurons is augmented which corresponds to adding more harmonics according to the Fourier series.

To train the network we need to know the available inputs and outputs. The traditional way to train a network is to assign random values to the weights and then wait for the function to converge using the gradient descendent method. Using this topology the network is trained using the least-squares method fixing a finite number of neurons and arranging the system in a matrix formAx=B. We will use cosines for approximating the function with pair functions and sine in the case of impair functions.

Numerical example of T-ANN's

Figure 12 shows the ICTL and the trigonometric neural networks icons. The trigonometric neural networks inside the ICTL include examples. The front panel and block diagram of the example can be seen in fig. 13. In the block diagram the code is related to training and evaluation of the network. The signals from the eye or voice could be recognized using trigonometric neural networks, as shown in figure 14 in which the example presents a signal approximation.
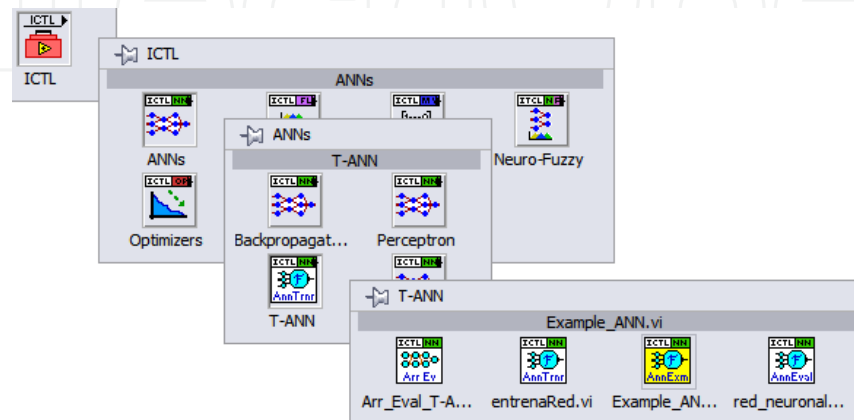

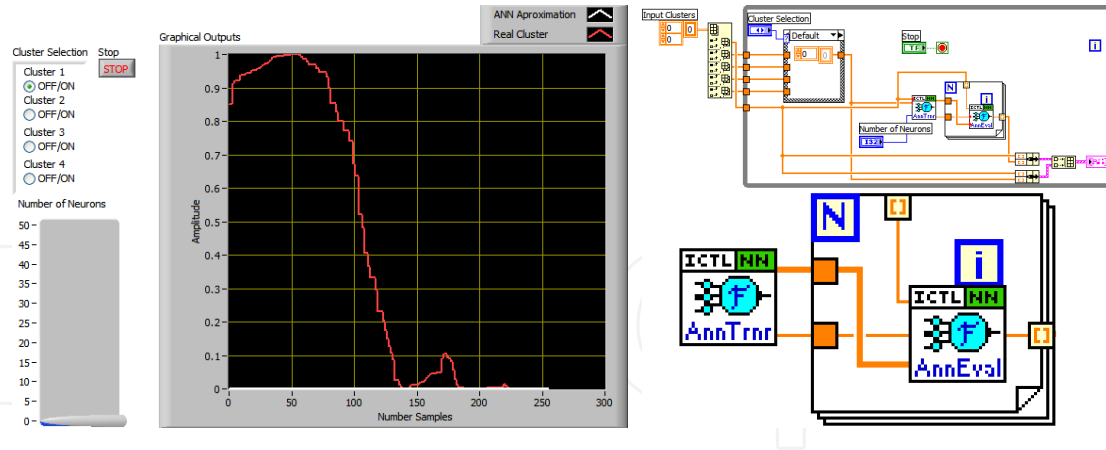**Figure 12.** The ICTL showing the Trigonometric Neural Networks icons

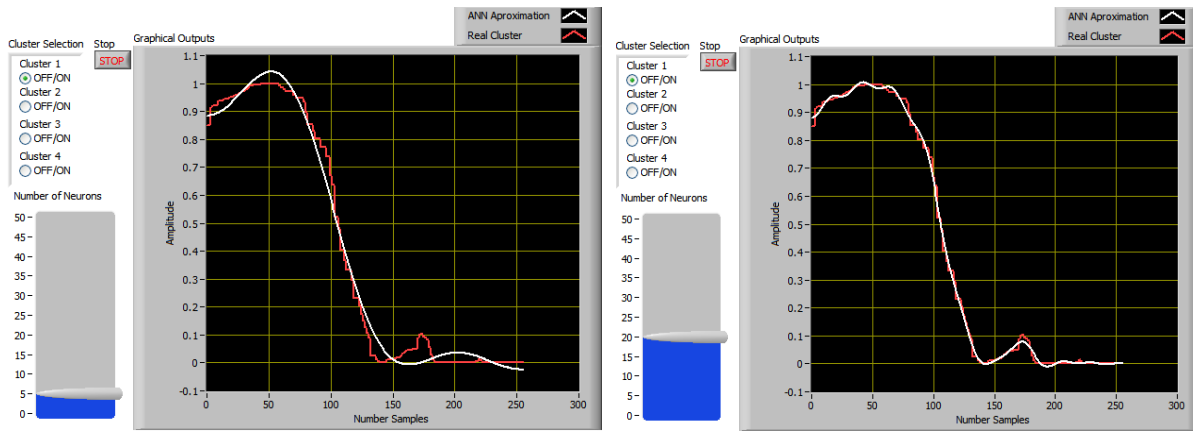**Figure 13.** The front panel and block diagram

Figure 14. Trigonometric neural network example network using 5 (left) and 20 neurons (right)

Fuzzy Cluster Means

Clustering methods split a set of N elements $X = \{x_1, x_2 \ldots, x_n\}$ into a c group denoted $c = \{\mu^1, \mu^2, \ldots \mu^n\}$. Traditional clustering set methods assume that each data vector can belong to one and only one class, though in practice clusters normally overlap, and some data vectors can belong partially to several clusters. Fuzzy set theory provides a natural way to describe this situation by FCM.

The fuzzy partition matrices M, for c classes and N data points were defined by three conditions: $M = \{U \in V_{cN} 1, 2, 3\}$

- The first condition: $\forall \quad 1 \leq i \leq c \quad \mu_{ik}[0,1], \quad 1 \leq k \leq N$
- The second condition: $\sum_{k=1}^{c} \mu_{ik} = 1 \quad \forall \quad 1 \leq k \leq N$
- The third condition: $\forall \quad 1 \leq i \leq c \quad 0 < \sum_{k=1}^{c} \mu_{ik} < N$

The FCM optimum criteria function has the following form:

$$J_m(U, V) = \sum_{i=1}^{c} \sum_{k=1}^{N} \mu_{ik}^m d_{ik}^2 \tag{9}$$

Where d_{ik} is an inner product norm defined as:

$$d_{ik}^2 = ||x_k - v_i||_A^2 \tag{10}$$

Where A is a positive definite matrix, m is the weighting exponent $m\epsilon[1,\infty]$. If m and c parameters are fixed and define sets then (U,V) may be globally minimal for $Jm(U,V)$ only if:

$$\begin{matrix} \forall \\ 1 \le i \le c \\ 1 \le k \le N \end{matrix} \quad u_{ik} = \frac{1}{\sum_{j=1}^c \left(\frac{||x_k - v_i||}{||x_k - v_j||} \right)^{\frac{2}{m-1}}} \tag{11}$$

$$\begin{matrix} \forall \\ 1 \le i \le c \end{matrix} \quad v_j = \frac{\sum_{k=1}^N (u_{ik})^m x_k}{\sum_{k=1}^N (u_{ik})^m} \tag{12}$$

FCM Algorithm:

The fuzzy c-means solution can be described as:

1. Fix c and m, set $p=0$ and initialize $U^{(0)}$
2. Calculate fuzzy centers for each cluster using $V^{(p)}$ (12)
3. Update fuzzy partition matrix $U^{(p)}$ for the p-thiteration using (11)
4. If $\left\| U^p - U^{(p-1)} \right\| < \epsilon$ then, $j \leftarrow j+1$ and return to the second step

In this algorithm, the parameter m determines the fuzziness of the clusters; if m is large the cluster is fuzzier. For $m \rightarrow 1$ FCM the solution becomes the crisp one, and for $m \rightarrow \infty$ the solution is as fuzzy as possible. There is no theoretical reference for the selection of m, and usually $m = 2$ is chosen. After the shape of the membership functions are fixed, the T-ANN's learn each one of them.

Predictive Method

Sometimes the controller response can be improved by using predictors, which provide future information and allow it to respond in advance. One of the simplest yet most powerful predictors is based on exponential smoothing. A popular approach used is the Holt's method.

Exponential smoothing is computationally simple and fast at the same time, this method can perform well in comparison with other more complex methods. The series used for prediction is considered as a composition of more than one structural component (average and trend) each of which can be individually modeled. We will use series without seasonality in the predictor. This type of series can be expressed as:

$$y(x) = y_{av}(x) + py_{tr}(x) + e(x); p = 0 \tag{13}$$

Where: $y(x)$, $y_{av}(x)$, $y_{tr}(x)$, and $e(x)$ are the data, the average, the trend and the error components individually modeled using exponential smoothing. The p-step ahead of prediction [3] is given by:

$$y*(x + p|k) = y_{av}(x) + py_{tr}(x) \tag{14}$$

The average and the trend components are modeled as:

$$y_{av}(x) = (1-\alpha)y(x) + \alpha\left(y_{av}(x-1) + y_{tr}(k-1)\right) \tag{15}$$

$$y_{tr}(x) = (1-\beta)y_{tr}(x-1) + \beta\left(y_{av}(x) + y_{av}(x-1)\right) \tag{16}$$

Where   and   are the average and the trend components of the signal. Where $y_{av}(x)$ and $y_{tr}(x)$ are the smoothing coefficients, its values range (0,1). y_{av} and y_{tr} can be initialized as:

$$y_{av}(1) = y(1) \tag{17}$$

$$y_{tr}(1) = \frac{\left(y(1) - y(0)\right) + \left(y(2) - y(1)\right)}{2} \tag{18}$$
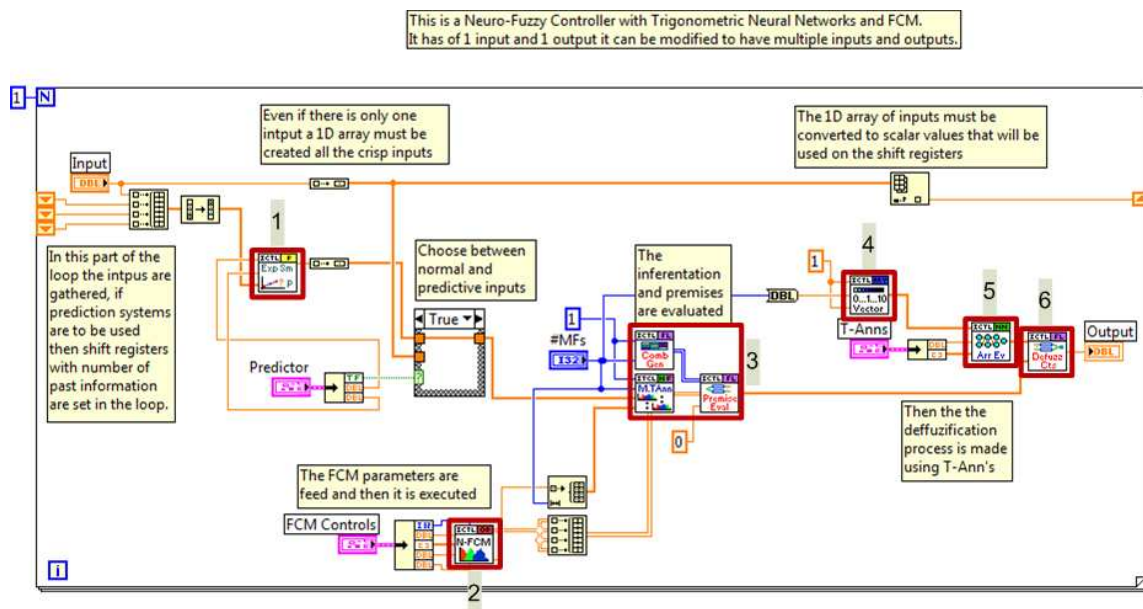

Figure 15. Block diagram of the neuro-fuzzy controller with one input, one output

The execution of the controller depends on several VI's (more information in [4]), which are explained in the following steps:

Step 1. This is a predictor VI based on exponential smoothing; the coefficients alpha and beta must be fed as scalar values. The past and present information must be fed in a 1D array with the newest information in the last element of the array.

Step 2. This VI executes the FCM method; the information of the crisp inputs must be fed as well as the stop conditions for the cycle; the program will return the coefficients of the trigonometric networks, the fundamental frequency and other useful information.

Step 3. These three VI's execute the evaluation of the premises. The first is on the top left generator of the combinations of rules that depends on the number of inputs and membership functions. The second one on the bottom left combines and evaluates the input membership functions. The last one on the right uses the information on

the combinations as well as the evaluated membership functions to obtain the premises of the IF-THEN rules.

Step 4. This VI creates a 1D array with the number of rules of the system: where n is the number of rules, it is used on the defuzzification process.

Step 5. This VI evaluates a T-ANN on each of the rules.

Step 6. This VI defuzzifies using the Takagi method with the obtained crisp outputs from the T-ANN.

This version of one input, one output of the controller was modified to have three inputs and four outputs; the block diagram is shown in figure 16.
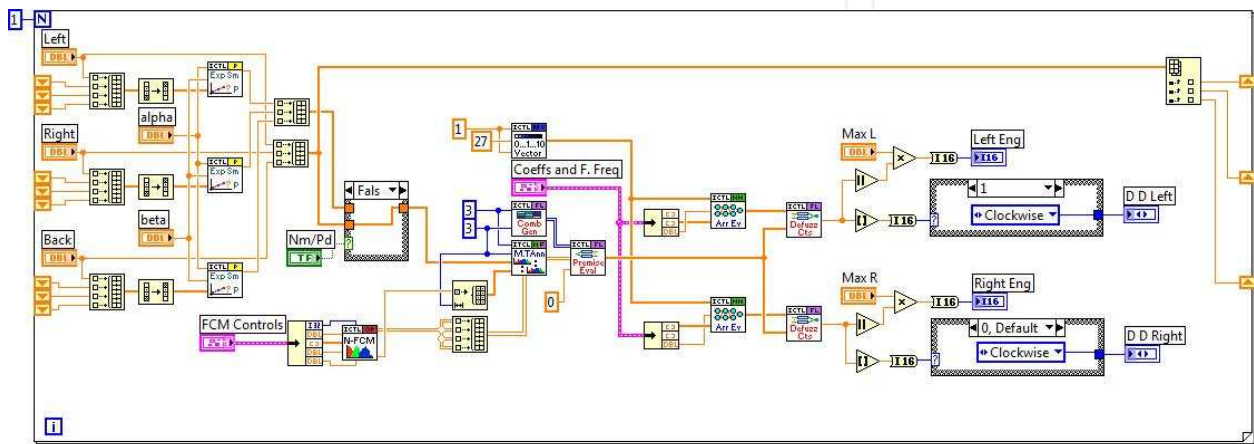

Figure 16. Neuro-Fuzzy controller block diagram

Each input is fuzzified with four membership functions whose form is defined by the FCM algorithm. The crisp distances gathered by the distance sensors are clustered by FCM and then T-ANN's are trained. As can be seen in figure 17, the main shape of the clusters is learnt by the neural networks and no main information is lost.
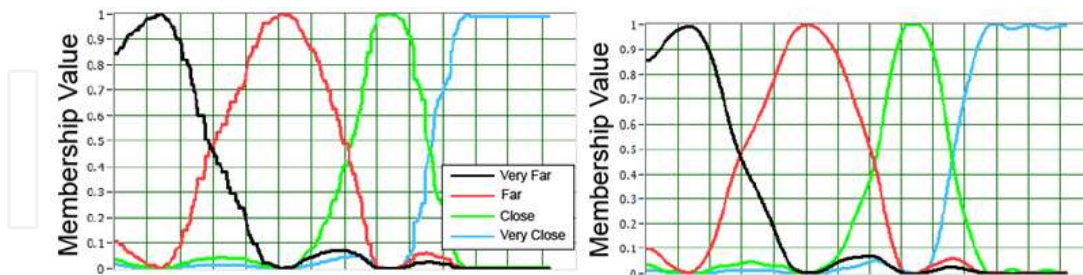

Figure 17. Input membership functions

With three inputs and four membership functions, there is a total of sixty four rules that can be evaluated. These rules are *IF THEN* and have the following form: **IF** x_1 is μ_{in} & x_2 is μ_{in} & x_3 is μ_{in} **THEN** *PWM LeftEngine, Direction Left Engine, PWM Right Engine, Direction Right Engine.*

The value of each rule is obtained through the inference method min that consists of evaluating the $\mu_{in's}$ and return the smallest one for each rule. The final system output is obtained by:

$$Output = \frac{\sum_{i=1}^{r}\left[\min\left(\mu_{i1,2,3}\right)NN\left(x_1,x_2,x_3\right)\right]}{\sum_{i=1}^{r}\min\left(\mu_{i1,2,3}\right)} \tag{19}$$

For the direction of the wheel, three states are used: clockwise (1), counterclockwise (-1) and stopped (0). The fuzzy output is rounded to the nearest value and the direction is obtained.

5. Results using the controller

The wheelchair was set on a human sized chessboard and the pieces were set in a maze, presented in Figure 18 with some of the trajectories described by the chair.
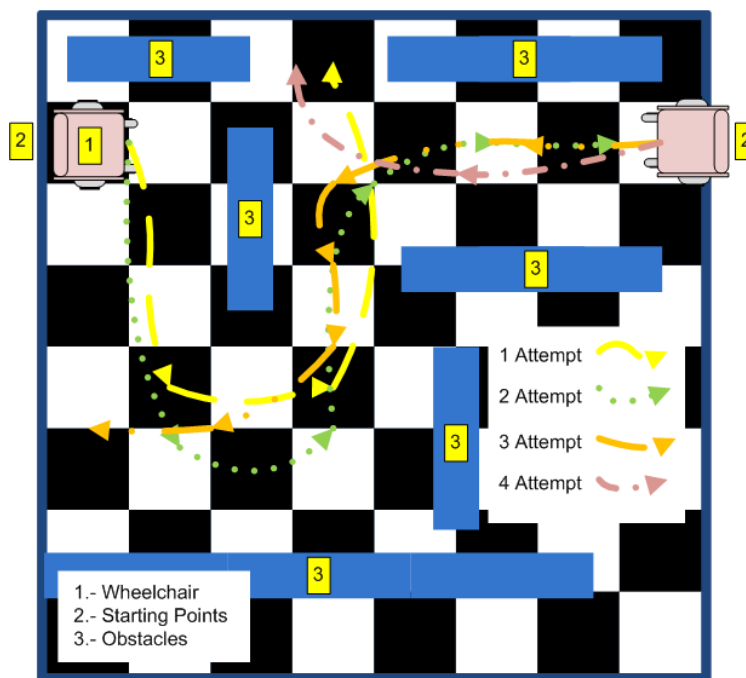

Figure 18. Wheelchair maze and trajectories

The wheelchair always managed to avoid obstacles, but failed to return to the desired direction. It also fails to recognize whether the obstacle is a human being or an object and, thus, have different behaviors to avoid them.

Controller Enhancements

Direction Controller

As can be seen from the previous results, the wheelchair will effectively avoid obstacles but the trajectories that it follows are always different, sometimes it may follow the directions we want but other times it will not. A direction controller can solve this problem, so we need a sensor to obtain feedback from the direction of the wheelchair. A compass could be an option to sense the direction, either the 1490 (digital) or 1525 (analog) from images SI [8]. After the electric wheelchair controller avoids an obstacle the compass sensor will give it information to return to the desired direction, as shown in Figure 19.
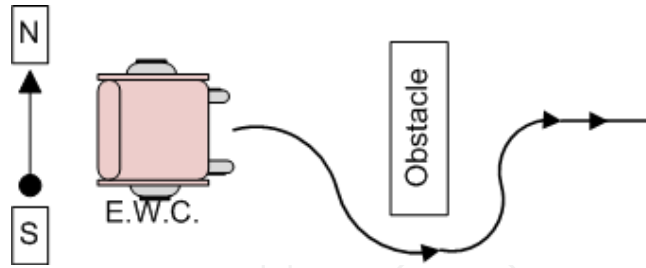
Figure 19. The wheelchair recovering the direction with the Direction Controller.

A fuzzy controller that controls the direction can be used in combination with the obstacle avoidance controller. The directions controller will have as input the difference between the desired and the current direction of the wheelchair. The direction magnitude describes how many degrees the chair will have to turn and the sign indicates if it has to be done in one direction or the other. The output is the PWM and the direction that each wheel has to take in order to compensate.

Three fuzzifiying input membership functions will be used for the degrees and the turning direction, as shown in figure 20. The range for the degrees is [0, 360] degrees and the turning direction is [-180, 180], also in degrees.
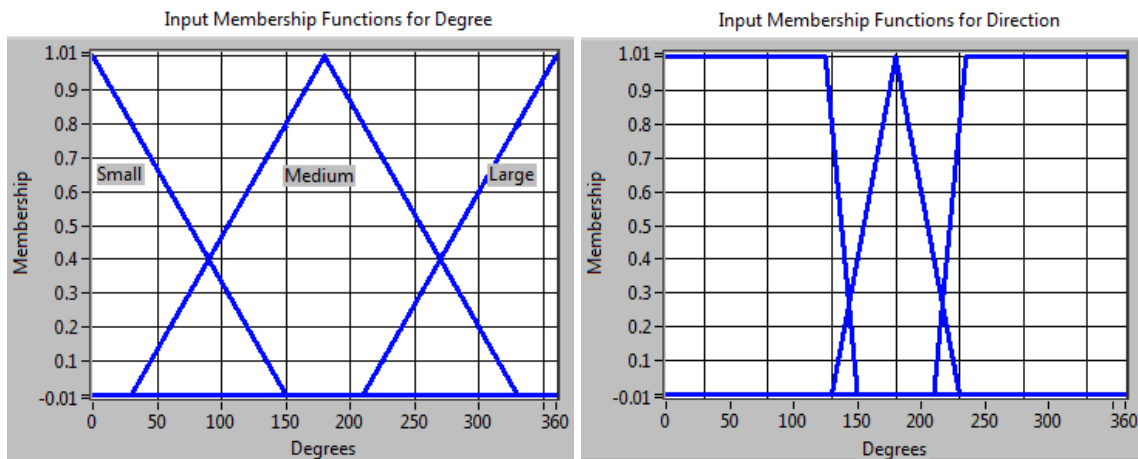

Figure 20. Input membership functions for Degrees and Direction

The form of the rule is the following: **IF** *degree* is A_{in} & direction is B_{in} **THEN** *PWM Left Engine , Direction Left Engine , PWM Right Engine , Direction Right Engine .*

Figure 21 shows the rule base with the nine possible combinations of inputs and outputs. The outputs are obtained with the rule consequences using singletons.

The IF-THEN Rules:

- CCW : Counterclockwise
- CW: Clockwise
- NC: No Change
1. IF Degree is Small & Direction is Left THEN PWMR IS Very Few, PWML IS Very Few, DIRR is CCW, DIRL is CW.

2. IF Degree is Small & Direction is Center THEN PWMR IS Very Few, PWML IS Very Few, DIRR is NC, DIRL is NC.
3. IF Degree is Small & Direction is Right THEN PWMR IS Very Few, PWML IS Very Few, DIRR is CW, DIRL is CCW.
4. IF Degree is Medium & Direction is Left THEN PWMR IS Some, PWML IS Some, DIRR is CCW, DIRL is CW.
5. IF Degree is Medium & Direction is Center THEN PWMR IS Some, PWML IS Some, DIRR is NC, DIRL is NC.
6. IF Degree is Medium & Direction is Right THEN PWMR IS Some, PWML IS Some, DIRR is CW, DIRL is CCW.
7. IF Degree is Large & Direction is Left THEN PWMR IS Very Much, PWML IS Very Much, DIRR is CCW, DIRL is CW.
8. IF Degree is Large & Direction is Center THEN PWMR IS Very Much, PWML IS Very Much, DIRR is NC, DIRL is NC.
9. IF Degree is Large & Direction is Right THEN PWMR IS Very Much, PWML IS Very Much, DIRR is CW, DIRL is CCW.
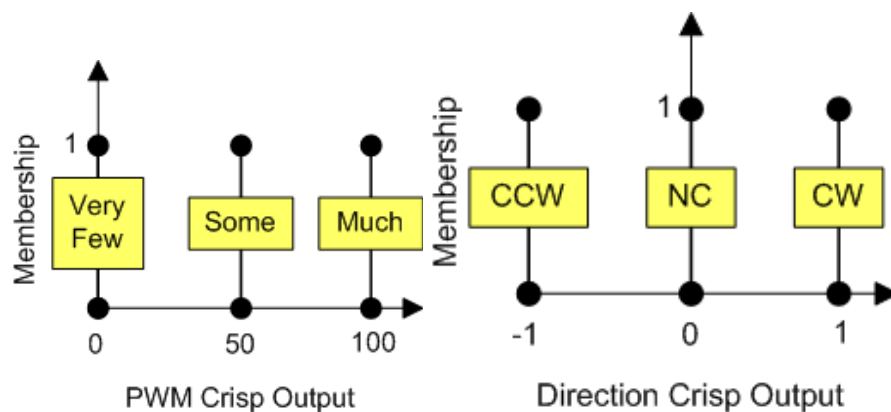

Figure 21. Rule Base and output membership functions for the Direction controller

The surfaces for the PWM and the direction are shown in Figure 22. For both PWM outputs the surface is the same, while for the direction the surfaces change and completely invert from left to right.
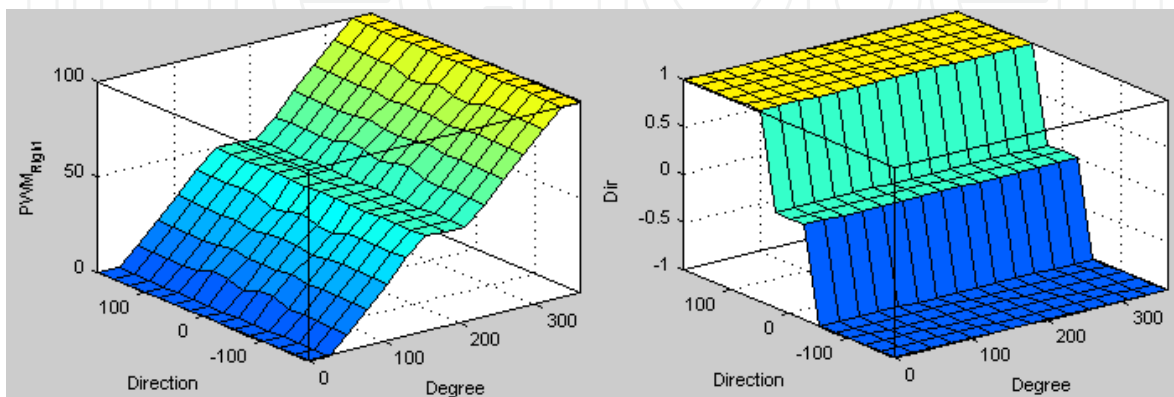

Figure 22. Surfaces for PWM and Direction outputs

This controller will act when the distances recognized by the sensors are Very Far, because the system will have enough space to maneuver and recover the direction that it has to follow, otherwise the obstacle avoidance controller will have the control of the wheelchair.

Obstacle Avoidance Behavior

Cities are not designed to be transited by disabled people, thus one of their main concerns is the paths and obstacles they have to cope with to get from one point to another. Big cities are becoming more and more crowded, so moving around on streets with a wheelchair is a big challenge.

If temperature and simple shape sensors are installed in the wheelchair (Figure 23 shows the proposal) then some kind of behavior can be programmed so the system can differentiate between a human being and a non human being. Additionally the use of a speaker or a horn is needed to ask people to move out of the way of the chair.
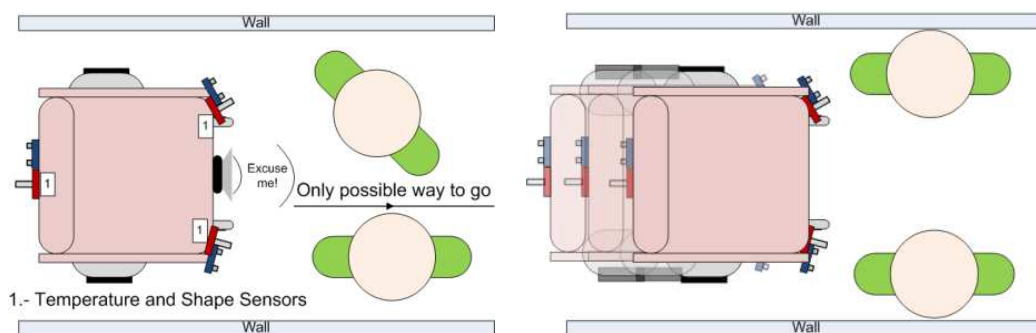

Figure 23. Wheelchair with temperature sensors for obstacle avoidance

The proposed behavior is based on a fuzzy controller which has as input the temperature in centigrade's degrees of the obstacle and as output the time in seconds the wheelchair will be stopped and a message or a horn will be played. It has three triangular fuzzy input membership functions as shown in Figure 24. The output membership functions are two singletons, as can be seen in Figure 25.
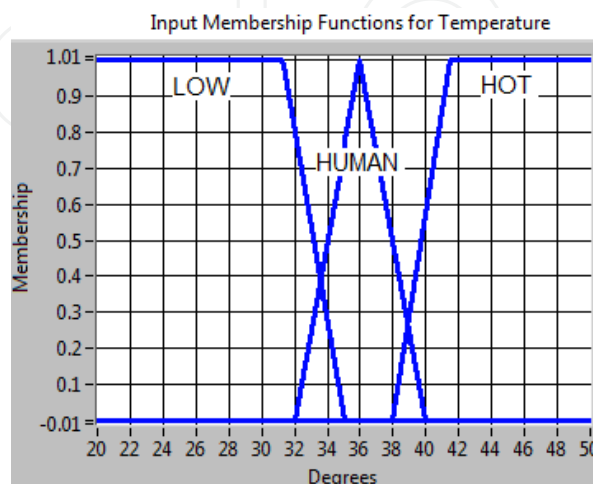

Figure 24. Input membership function for temperature

The IF-THEN Rules:

1. IF Temperature is Low THEN TIME IS Few.
2. IF Temperature is Human THEN TIME IS Much.
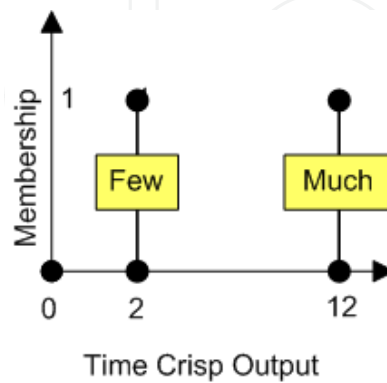3. IF Temperature is Hot THEN TIME IS Few.

Figure 25. Singleton outputs for the temperature controller

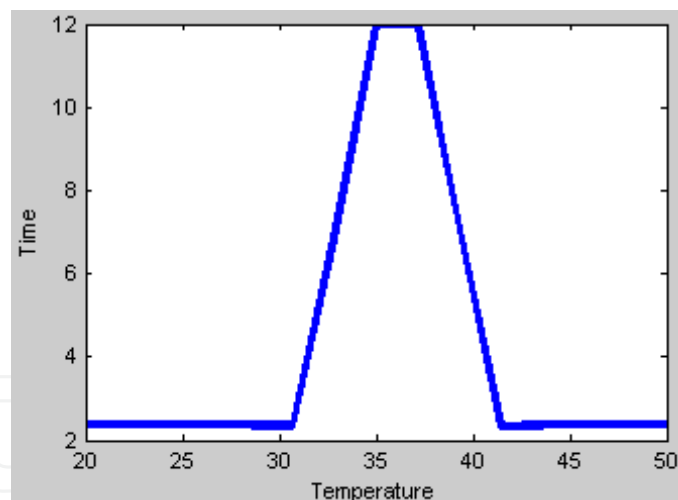The controller response is shown in Figure 26.

Figure 26. Time controller response

6. Structural design

The full system was built on a Quickie Wheelchair. The full system diagram is shown in Figure 27. Using LabVIEW, three different kinds of controls were programmed:

1. Voice control
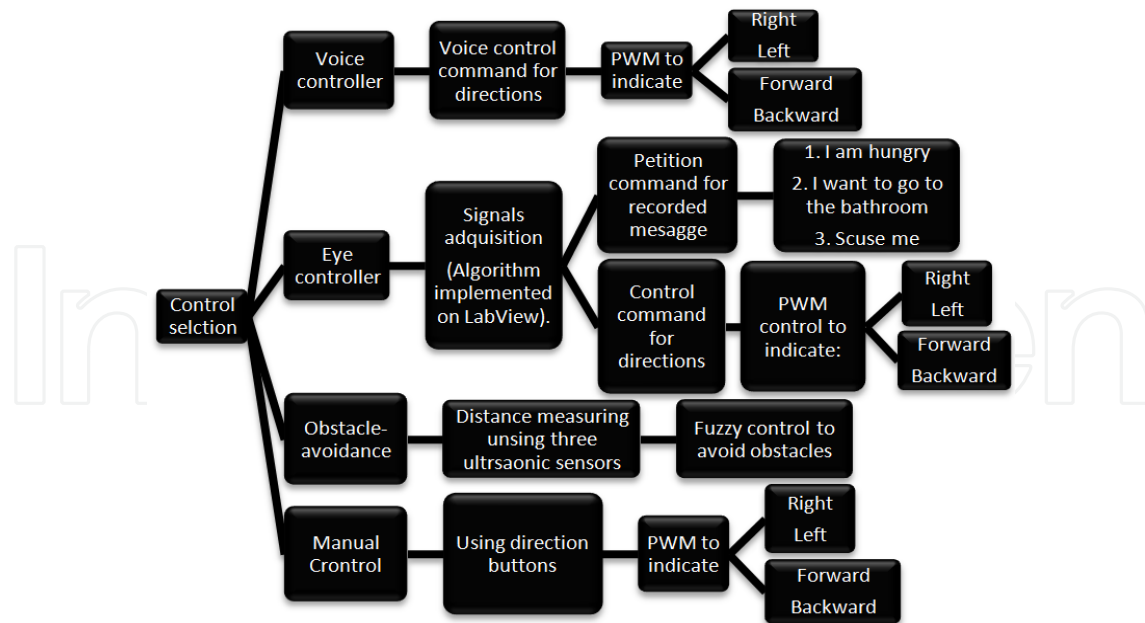2. Eye-movements control
3. Keyboard control

Figure 27. Control structure. The user can select any of the controls depending on his/her needs and the surrounding environment.

The wheelchair is controlled using two coils to generate an electromagnetic field which could be detected by two sensors. Depending on the density of the magnetic field and its intensity, the motors could be controlled to move the wheelchair in any direction. However this solution also requires that both coils should be fixed in place and they cannot be moved with respect to the sensors, so the sensed field will be always the same for one determined configuration. The use of fuzzy logic to design an obstacle-avoidance system and the Hebbian network used to determine the different kinds of eye movements were the tools that helped us to obtain efficient answers. Specific hardware is required to obtain the signals that would be processed with these systems. Each control was programmed on LabVIEW in different files and all of them were included in a LabVIEW project. Each one must be executed separately, so when using the voice controller, the eye controller cannot be used. In future, the obstacle-avoidance system will have higher priority than the eye and voice controllers.

The hardware used is:

1. DAQ: for sensing the different voltage signals produced by the eyes and to set the directions using the manual control. Each one gets values from different ports of a NI USB-6210. In the case of the voltages generated by the eyes, the data acquisition was connected to the analog port and the manual control to the digital input port.
2. CompactRIO: This device was used to generate PWM to the coils allowing us to have control for the different directions. The cRIO model 9014 had the following modules:
a. a.Two H-bridges: for controlling the PWM.
b. b.5 V TTL Bidirectional digital I/O Module.
3. BasicStamp[12]: This device is used to acquire the signals detected from three ultrasonic distance sensors.

4. Three ultrasonic sensors that measure distance and are used to help the obstacle-avoidance system. Two of them are placed in front of the wheelchair and one at the back.

5. A laptop to execute the programs and visualize the different commands introduced by the user.

7. Virtual trainning

7.1. Augmented reality

The simulator has a variety of modules that make up the augmented reality of the virtual environment. Augmented reality refers to those aspects that help to represent real physical situations and whose data is printed on screen to help users to understand them.

Distance to close objects

Taking the user's position in the virtual world as the center, a circle with a 5-meters radius is generated; then the distance to all objects within this area is computed with vector operations and the screen displays the distance to the nearest one and its name. For this, a file with specific information about all the objects at the simulator stage is consulted. This information allows the user to gain experience of how to move in small spaces, know the speed of the chair and the time between the moment when the command is indicated and the moment when it is executed (system response time).

Variation of the wheelchair's movement according to the terrain

Depending on the terrain where the patient moves in the virtual world, different physical phenomena are represented. If the terrain's surface is uneven, such as grass or pavement, vibrations are displayed on the screen and there is a change in the traction of the intelligent wheelchair. If the terrain is tilted, the perspective tilts and the corresponding acceleration changes are re-created as shown by the following formula obtained from the analysis of forces in Figure 28.
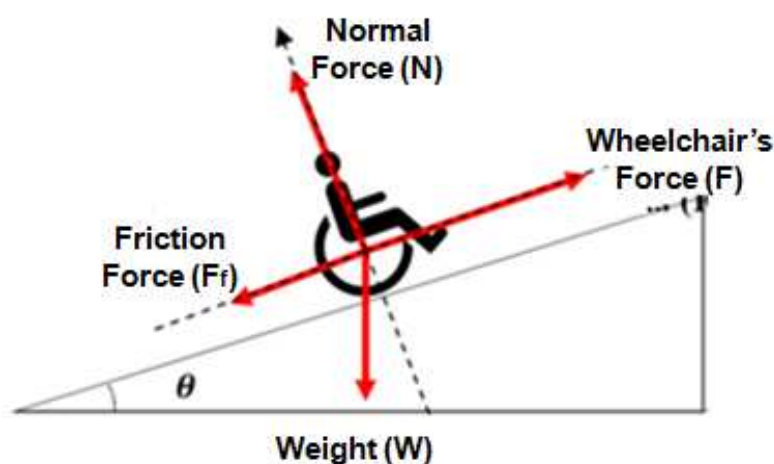

Figure 28. Forces that act upon the user and the intelligent wheelchair while moving on a tilted plane.

$$a = g \cdot \left(sen\,\theta + \mu_k \cos\theta \right) \tag{20}$$

Where a is the acceleration in the inclined plane, g is gravity, μ_k is the kinetic friction coefficient and θ is the inclination angle.

Static and dynamic obstacles

The 3D virtual trainer presents the dynamic obstacles: pedestrians, animals; and the static obstacles: furniture, walls, etc. that would most likely be seen in the locations in which the patient conducts his/her activities. The simulator offers online recommendations on how to avoid these obstacles or interact with them.

Within the virtual world, sounds and noises, both from dynamic objects and the environment itself, are presented. The user can listen to conversations when he/she is outside: barking, car engines, among others. These sounds vary depending on the patient's position in the virtual world and interaction with certain objects. For example, the volume of dynamic objects (such as people or animals) increases or decreases proportionally according to their distance from the user in the virtual environment.

3D Perspectives

As a way to help users, they can see the virtual world from different perspectives. The first perspective, or first person, presents the objects to the patient as seen from the intelligent wheelchair. This is the most useful because it is how the user will see them in real life. The third-person perspective offers a view of the user's avatar and the objects closest to it, that is, the camera is placed above and behind, five meters in each direction, with a 45° tilt. Thus, the user can see more details of its virtual environment. Finally, a perspective that is independent from the user's movement is offered, making it possible to explore the entire virtual world to get a look at the objects to interact with later.

Statistics: augmented reality

Figure 29. This image shows the virtual trainer along with the augmented reality interface. In the upper left corner, information about the performance of the user is shown: elapsed time, found targets. In the upper right corner there are more statistics such as number of collisions and the name of the nearest object. In the lower area data that can help the user to navigate more properly or about the closest

As mentioned in the description of some other modules of the virtual trainer, the user will be shown at any time information about objects in the simulator: distance to the nearest object, its name and the distance to the nearest target; as well as information about his/her performance: time spent at each level of the simulator and the number of collisions. Likewise, the screen's lower left corner providesinformation on the nearest object, such as extra precautions to consider, material and dimensions, or what would be the best instruction on driving the wheelchair through the location the user is currently at. Also, if relevant, the user is informed about the changes in terrain and sound as well as the change of perspective. These statistics are shown in Figure 29 above.

In this approach, the user is provided with not only a way to learn how to control the intelligent wheelchair, but also a space where he/she can find cultural information about the objects that constitute the virtual world. Likewise, the augmented reality interface facilitates and makes the user's training more pleasant.

7.2. Simulator performance

Due to the intense computing conducted in real time to show the aforementioned statistics and the level of detail present in virtual environments, such as the interior of houses, the implementation of algorithms that provide greater game performance and visual quality is necessary.

In simulators, video games and other interactive media, 3-D models and animations are crucial components. Games like Gears of War and Half-Life 2 would not be as striking if not for the vivid, detailed models and animations. Games within the XNA Framework, which are able to take advantage of the GPU (Graphics Unit Processor) of the Xbox 360 or PC, are not the exception. Many advanced rendering techniques can be exploited through the XNA Framework, such as hardware instancing.

Traditional implementations of 3-D models require a large header on the CPU (Control Unit Processor) and are not efficient or completely instantiated. Many processes are performed in the CPU and each part of the model requires its own call to Draw method and sometimes multiple calls if the model has a large number of polygons. This means, in the XNA framework, that creating a large number of models in the CPU creates a bottleneck.

The aim of this work is to present an alternative to "traditional" techniques to instantiate 3-D models. This technique even makes it possible to render animated models and, depending on its complexity, we can draw more than 45 models with a single call to Draw. Since the Xbox has a powerful graphics card, this is very desirable, and any way the level of processing on the CPU is reduced.

Hardware instancing works by sending two streams of vertexes to the video card, in order to send information about the objects' vertexes (which is called a regular vertex buffer) and information for the instances (position and color) to the GPU simultaneously. The advantage of hardware instancing is that a flag can be used to indicate which data flow (stream) contains information about certain vertexes and instances. Thus, only the original mesh has

to be introduced in the data sequence of vertexes and XNA ensures that these data will be repeated for each instance. Thissaves memory and is generally easier to handle, since you can use the original mesh and the only thing needed is a call for each subset. This is because the main bottleneck in the GPU occurs when transformations are made to the pixels, such as the application of textures. With hardware instancing, the common operations for these transformations are not made every time an object is going to be drawn but only a relationship between the data flows through the flags. The arrangement of data streams is shown below in Figure 30.
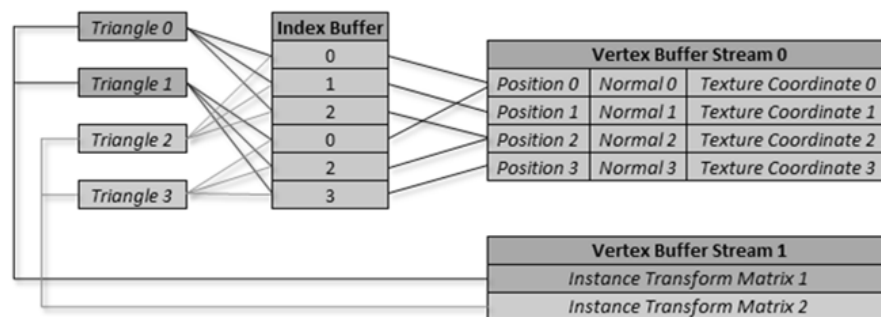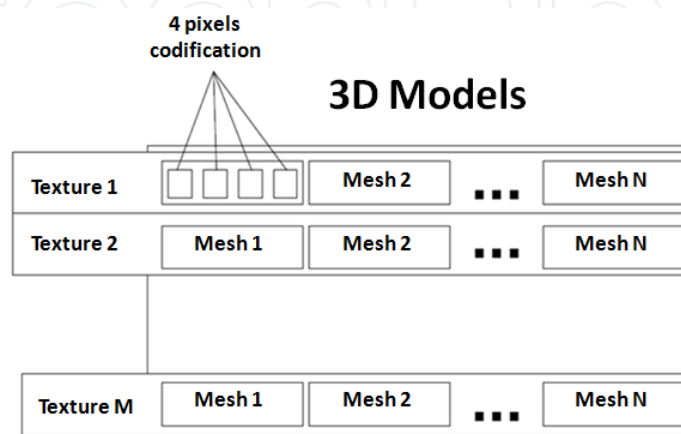

Figure 30. Array of the data sequences of the vertex buffers and their pointers
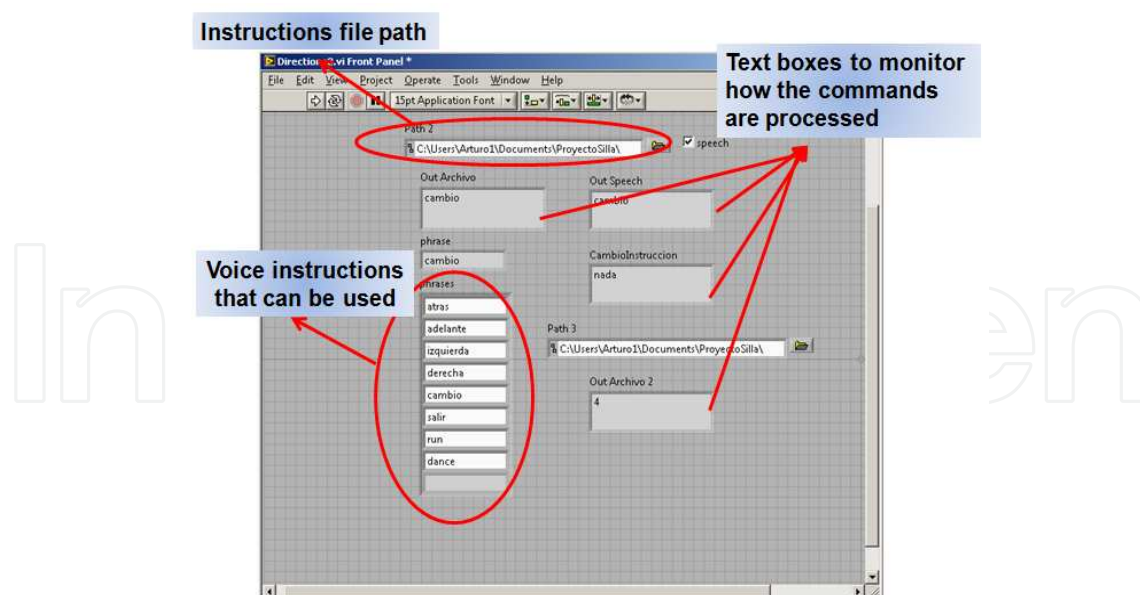
To accomplish hardware instancing, the following classes, processors and pipelines were used:

- InstancedModelProcessor.- This is the processor that does the heavy work of converting the model's data into vertexes, and giving as output an Instanced Model Content. Basically a cycle goes through the vertexes and assigns them a corresponding texture.
- InstanceSkinnedModelPart.- It is actually a wrapper around the ModelMesh class, with additional functionality to support instancing.

The last point to consider in hardware instancing is that each row of the processed mesh transforms is encoded as a pixel, and because the matrix dimensions are 4x4, 4 pixels represent a transform. Then a texture is assigned to each transformation according to the vertexes that represent them. This form of encoding is depicted in Figure 31.

Part I

Part II

Figure 31. Encoding of the mesh transforms as in the Hardware Instancing method

7.3. Labview Interface for eye and voice control

The Labview interface that verifies the operation and control of Windows Speech Recognition is shown Figure 20. This window appears in the background of the computer when the game starts if the user wants to verify the voice instructions identified (these commands are recognized only when the Labview program has started). This window appears by means of an object of type Process which calls the Labview file (Virtual Instrument) from XNA.

To communicate with the two applications (Labview and XNA) and control the virtual trainer through the acquired data by Labview, a parallel access to a file which contains the instructions executed by the user in string format was implemented; then these commands are encoded to generate an event in XNA. This part of the project can be improved in order to move the information between the two applications more efficiently and with lower exception handles.

7.4. Web page and data base

In this section the web page of the project where the new simulator's users can register to monitor their activity and performance is explained. The patient's progress is measured through the mentioned statistics, such as number of collisions, instructions used, elapsed time in each level, among other variables. Based on them, a score is set to the user according to the weights assigned to each statistic. This page is also available for the patients' doctors; the page was developed using Microsoft Web Developer 2008.

Once registered, users can log in to access their information as well as that of the other players. Likewise, the user can also share information with other people and send them a

message through the web page. This is also a quick means of communication for specialists who care for patients with disabilities and, at the same time, a space where users can share the information not only about the virtual trainer but about their life experiences which can be useful for other users. With this, a community can be created where patients can identify with and help each other.

The screen in Figure 32 shows the storage of users' data through SQL Express Edition. The general table that keeps a record of all the registered patients shows their best time (time in which they completed a level of the simulator), best score, number of games played, as well as their score average.

Similarly, more detailed statistics for each individual user are stored, which can be seen in Figure 33: scores, number of collisions, best time and most used instruction. In this way the patients can monitor their progress and work on the command that is most difficult for them. To help them, the targets that they have to collect in each level of the virtual trainer are placed in such a way as to let the users strengthen the instructions with which they collide the most and lower the percentage of the instruction they most widely use.
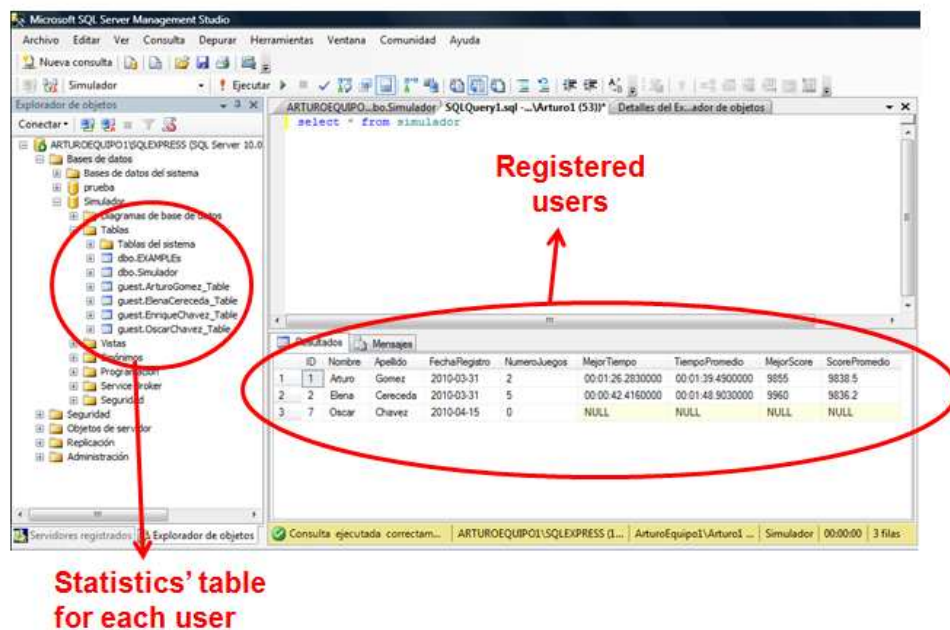

Figure 32. Information kept in the SQL Express data base about the statistics of the registered users
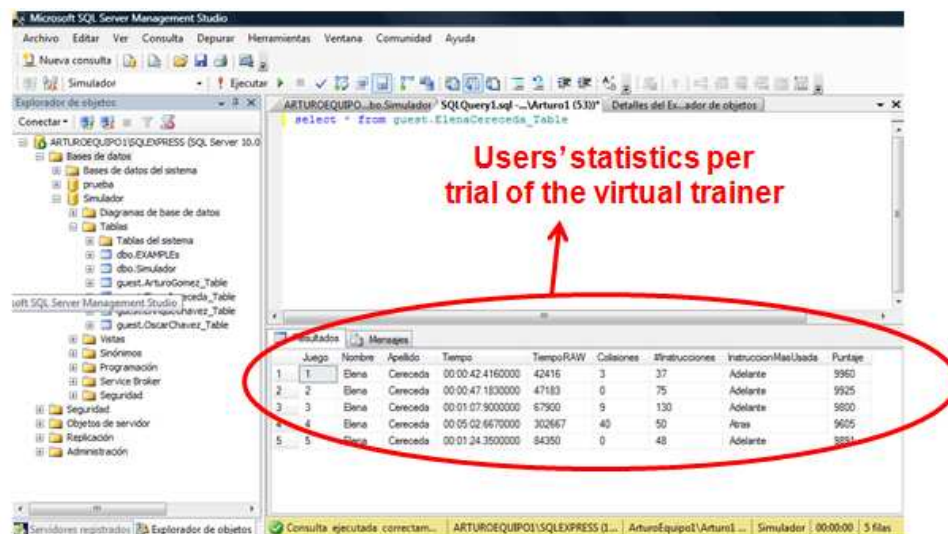
Figure 33. Statistics kept for every user and on which the virtual trainer based the way it distributes the targets among the virtual world to help the patients to strengthen their skills by using the intelligent wheelchair.

All the previous information is transmitted to the database using the libraries System.Data.SqlClient and System.Data.OleDb once a user has completed a level of the virtual trainer. The information consists of the statistics mentioned in section IV.1 and in shown in Figure 5, and some mathematical calculations to obtain averages and keep count.

8. Results

The following results show the controller performance. The voice controller increases the accuracy; if the wheelchair needs to work in a noisy environment, a noise cancelation system has to be included. The results are quite good when the wheelchair works in a normal noise environment, with the average value ranging from 0 to 70 db, and an average value of around 90% is obtained. The EOG recognition system changes the precision value according to the time because the user requires a certain length of time to move the eye in a correct way in order to recognize the signal. At first, the user is not familiarized with the whole system, thus the signal is not well defined and the system has a medium performance. After six weeks, the system increases the precision to around 94 %. Fig 34 shows the results.
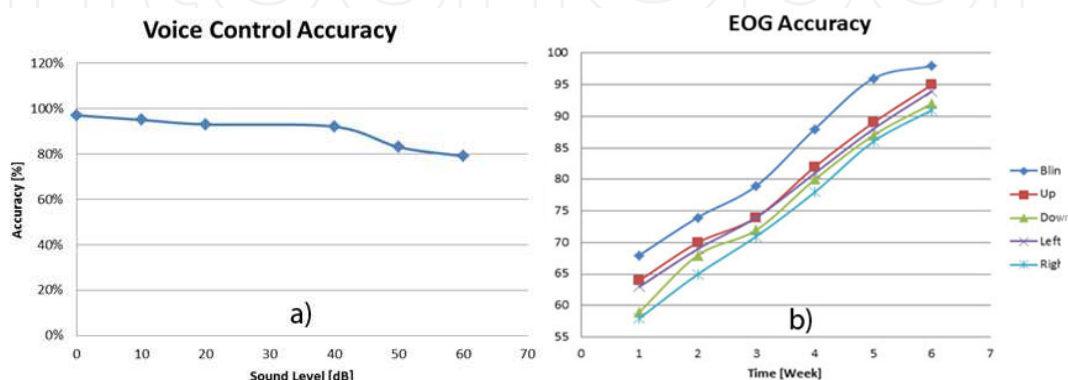

Figure 34. a. Tests on the voice control system response. b. EOG system response.Experimental Results

9. Conclusions

The complete system works well in a laboratory environment. The signals from eye movement and voice commands are translated into actual movements of the chair, allowing people who are disabled and cannot move their hands or even their head to move freely through spaces. There is still not a full version that can run an avoidance system at the same time as the chair is been controlled with eye movement. This should be the next step for further work. As was intended, the four main control systems that give the wheelchair more compatibility and adaptability to patients with different disorders were successfully completed. This allowed the chair to be moved with the eyes for those who cannot speak. Speech recognition was included for people who cannot move and directional buttons (joystick) for any other users. Many problems were presented when trying to interfere with the systems already built by the manufacturer; the use of magnetic inductors is one of the temporal solutions that should be eliminated even though the emulation of the joystick is good and works well. The use of these inductors produces a lot of power loss, thus considerably reducing the in-use time of the batteries. It also generates a small retardation in the use of Windows Vista Speech Recognition software and enters some faults into our system, since it is well known that this user interface is not well developed and sometimes does not recognize what it was expected to do, which is not good enough for a system like ours that requires a quick response to commands. This project demonstrated how intelligent control systems can be applied to improve existing products. The use of intelligent algorithms broadened the possibilities of interpretation and manipulation.

Author details

Pedro Ponce, Arturo Molina and Rafael Mendoza
Escuela de Ingeniería y Arquitectura, Instituto Tecnológico y de Estudios Superiores de Monterrey, Mexico City, Mexico

Acknowledgement

10. References

Book Chapter

[5] P. Ponce and F. D. Ramirez.*Intelligent Control Systems with LabVIEW*.UnitedKingdom, Springer, 2009.

[7] R. Barea, L. Boquete, M. Mazo, E. López and L. M. Bergasa, *Aplicación de electrooculografía para ayuda a minusválidos.*; Alcalá de Henares. Madrid, Spain: Universidad de Alcalá.

[15] F. D. Ramirez and D. Mendez, *Neuro-FuzzyNavigationSystemfor Mobile Robots. Electronics and CommunicationsEngineering Project,* Instituto Tecnologico y de Estudios Superiores de Monterrey Campus Ciudad de México, 2007.

Journal

[1] International classification of functioning, disability and health: ICF. World health Organization, Geneva 2001, 228 págs.

[2] G. Krishnamurthy and M. Ghovanloo, *Tongue Drive: A Tongue Operated Magnetic sensor Based Wireless Assistive Technology for people with severe Disabilities*, IEEE Circuits and Systems, 2006. ISCAS 2006.Proceedings. 2006 IEEE International Symposium

[13] P. Ponce, et al. *A Novel Neuro-Fuzzy Controller Based on Both Trigonometric Series and Fuzzy Clusters.*IEEE International Conference an Industrial Technology. India. ICIT, Dec 15-17, 2006, 2006.

[14] P. Ponce, et al. *Neuro-Fuzzy Controller Using LabVIEW*. Paper presented at the Intelligent Systems and Control Conference by IASTED, in Cambridge, MA, Nov. 19 - 21, 2007.

Online Journal

[3] United States Department of Veterans Affairs, " New Head Control for Quadriplegic Patients, 1970," Rehabilitation Research & Development Service, http://www.rehab.research.va.gov/jour/75/12/1/lozach.pdf ,   (accesed October 21, 2009)

[4] Quickie-Wheelchair.com, "Quickie P222 SE – Wheelchair – Quickie-Wheelchair.com", http://www.quickie-wheelchairs.com/products/Quickie-P222-SE-2974.html,   (accesed October 16, 2009)

[6] National Instruments Corporation, "NI LabVIEW – The Software That Powers Virtual Instrumentation – National Instruments", http://www.ni.com/labview/ , (accesed October 16, 2009).

[8] National Instruments Corporation, "NI USB-6210 – National Instruments", http://sine.ni.com/nips/cds/print/p/lang/en/nid/203189 , (accesed October 16, 2009).

[9] National Instruments Corporation, "NI Crio-9014 – National Instruments", http://sine.ni.com/nips/cds/print/p/lang/en/nid/203500 , (accesed October 16, 2009).

[10] National Instruments Corporation, "NI 9505 – National Instruments", http://sine.ni.com/nips/cds/print/p/lang/en/nid/202711 , (accesed October 16, 2009).

[11] Parallax Inc. "BASIC Stamp Discovery Kit – Serial (With USB Adapter and Cable)", http://www.parallax.com/StoreSearchResults/tabid/768/txtSearch/bs2/List/0/SortField/4/ProductID/320/Default.aspx , (accesed October 21, 2009).

[12] XNA libraries, http://msdn.microsoft.com/en-us/aa937791.aspx