A WEARABLE NEURAL INTERFACE FOR REAL TIME TRANSLATION OF SPANISH DEAF SIGN LANGUAGE TO VOICE AND WRITING

R. Villa-Angulo[1] & H. Hidalgo-Silva[2]

[1]Instituto de Ingeniería
Universidad Autónoma de Baja California
Calle de La Normal S/N y Blvd. Benito Juárez,
Fracc. Insurgentes Este. Mexicali, Baja California,
21280, México
ravilla@uabc.mx

[2]Departamento de Ciencias de la Computación, CICESE,
Km. 107 Carr. Tijuana-Eda,
Ensenada, Baja California
22800 México
Hugo@cicese.mx

Received: December 11th 2002. Approved: November 30th 2005

ABSTRACT

This paper describes a work related to the design and implementation of a communication tool for persons with speech and hearing disabilities. This tool provides to the user a Human-Computer interface capable of the capture and recognition of gestures belonging to the Mexican Spanish Sign Alphabet. To capture the manual expressions, a data-glove constructed to sense the position of fifteen articulations of one of the user's hand is described. A location system that detects the position and movements of the hand with respect to the user's body is also constructed. The data-glove and location system signals are processed by a pair of programmable automatons. The automaton's outputs are sent to a personal computer that realizes the gesture recognition and interpretation tasks. Artificial neural network techniques are utilized to implement the mappings of the space of information generated by the instruments to the interpretation space, where the representation of the gestures are found. Once a gesture is captured and interpreted, it is presented in written form through a screen mounted in the clothes of the user, and in verbal form by a speaker.

RESUMEN

En este documento se presenta el trabajo de diseño e implementación de una herramienta de comunicación para personas discapacitadas del habla y del oído. Esta herramienta es conceptualizada como una interfaz humano-computadora capaz de capturar y reconocer ademanes del lenguaje español signado de México. Para realizar la captura de las expresiones manuales, el sistema utiliza un guante de datos basado en sensores de flexión capaces de medir la posición de quince articulaciones de una de las manos de un usuario, y un sistema basado en sensores de ultrasonido para detectar la posición y movimientos de la mano con respecto al cuerpo del usuario. Un par de autómatas programables realiza el tratamiento de la información proveniente del guante de datos y del rastreador de movimientos y una computadora personal realiza el trabajo de reconocimiento e interpretación de los ademanes.

Para realizar el procesamiento de la información se utilizan técnicas de redes neuronales artificiales con las cuales se realizan mapeos del espacio de información generado por los instrumentos a un espacio de soluciones donde se encuentran los significados de los ademanes representados por el usuario. Una vez capturado e interpretado un ademán, éste es presentado en forma escrita a través de una pantalla montada en la ropa del usuario, y en forma sonora a través de una bocina la cual pronuncia la letra que ha sido representada.

KEYWORDS: Neural Networks, Human-Computer Interfaces, Signed Languages.

1. INTRODUCTION

Human-Computer Interaction (HCI) has become a very important part of our daily life. The main objective of HCI systems has been to transfer the natural ways of human communication to the HCI systems [1]. With this motivation, during the last years the interest in introducing some Human-Human communication forms has been increasing, as signed languages, which involve hand and arms movement. Signed languages are conceived as some kind of non verbal communication between people and involve a wide range of actions beginning with the simple pointing to something and ending with more complex movements that express feelings and allow to set up structured conversations among people.

A number of works have been done related with sign language recognition over the last two decades, some based on traditional processing tools and methods, e.g. Hidden Markov Models [2], other on neural nets [3, 4 & 5], rule extraction and also temporal concept learners [6]. Some reviews are provided in [1 & 7].

After extensive linguistic analysis of signed languages, Johnston [8] found that gestures can be described in terms of four basic manual features: 1) the hand shape, which defines the configuration of the joints of the hand, 2) the orientation, which specifies the direction where the hand and fingers are pointing, 3) the place of articulation, which specifies the hand position related to the body, and 4) the movement that is the most complex feature and consists of the change trough time of any combination of the other three features.

When working with hand gestures in a HCI system it is necessary to provide the meaning of each gesture. To realize a gesture interpretation, the computer should be capable of measuring static and dynamic configurations of the hand, arm or other parts of the body. These measurement and recognition problems have been undertaken in two different approaches: recognition based on computer vision techniques [9, 10, 11 & 12], and interpretation based on data glove devices [3, 4, 13 & 14].

The computer vision approach suggests the use of a video-camera and computer vision techniques to recognize and interpret the gestures. The data glove based implementation requires for the user to wear a device covering his hand (in this case a glove). This device is generally connected to a set of cables that carry optic or electric signals to the computer.

Both techniques require the control and adequate use of physical devices capable to transform the position and movements of the user into digital information. When the purpose of the device is to be utilized by persons with hearing and speech disabilities, then the problem must be undertaken from the construction of an adaptable and portable interface point of view. It should work in real time to enable a user to establish a normal conversation. Also, it must incorporate more than one channel of communication, for example, in visual and sonorous ways for transmitting the information to the receiving person. Considering the portability issue, the computer vision approach suffers from the disadvantage of being tied to a camera-computer system, with added portability and high computing power requirements. On the other hand, the data-glove based approach allows portability; require less processing and it is easier to be attached to the user's hand. For those reasons in this application we consider the data-glove approach.

Fels et al. [3 & 4] observe that adaptable interfaces are a natural and very important class of applications for artificial neural networks. They constructed a system called Glove-Talk that implemented a neural network interface between a commercial data-glove and a DECTalk speech synthesizer [3]. Their system was constructed on a Silicon Graphics workstation due to the shared memory architecture required to process communication, making the system non-portable. In the Glove-Talk II, Fels et al. [4] constructed other neural network interface, but now mapping hand gestures to control parameters of a parallel format speech synthesizer. It was based on a Cyberglove, a polhemus sensor, a keyboard and a foot pedal. The complexity of the system made the learning task difficult even for an accomplished pianist [4]. Also, the system had portability issues due to the pedal needed to control the volume.

When a system must provide a wide and extensive band of control for some complex physical devices, an adequate mapping among the person's movements and the device behavior becomes crucial. Neural networks make possible to build interfaces where the mapping is automatically adapted during the training phase. This kind of interface simplifies the design process of a compatible mapping and facilitates the adaptation to different users.

The Mexican Spanish Sign Language is composed by a group of twenty seven gestures pertaining to the representation of each of the letters that form the alphabet, and another great group of gestures called ideograms, in which complete words are represented [15].

In this paper we present a device designed for the capture and recognition of the twenty seven gestures of the manual alphabet. This is due to two reasons: on the one side, with the manual alphabet any word can be spelled no matter how complicated it is. On the other, to execute any ideogram we have to gesture the first letter of the word and then make a movement or a gesture change to complete the representation. For this purpose we require only one hand of the user, and then we need to sense the movements and the articulations of one hand.

In section two, we present an analysis of the hand's physiology and the articulations utilized to represent the manual alphabet, then we describe the instrumentation developed for sensing the position of the articulations (the shape of the hand) and the hand's movements related to the body of the user. In section three, an analysis of the data distribution is presented in order to detect class overlapping in the data generated by the devices. To visualize this distribution a Kohonen Neural Network is utilized. In the same section, the design of the neural classifiers is considered. In section four the integration of the different parts of the system is analyzed, and a scheme for the analysis of the space-time feature of gestures is proposed. In section five, the representation of gestures in written and sonorous form is presented. In section six the real time translation system is described. In section seven, the evaluation of the system working with different neural network classifiers is presented. In section eight, an analysis of portability is presented showing that the system can easily become a wearable tool for speech and/or hearing impaired persons. And, finally in section nine we present our final remarks.

2. INSTRUMENTATION

2.1 Hand joints and Spanish Sign Alphabet representation

The set of gestures to be recognized consists of the twenty seven letters of the Spanish Sign Alphabet. The device that captures these representations should consider the articulations that intervene when representing each one of the gestures. Fig. 1 shows an image of the human hand, the bones that compose it and the degree of freedom (DOF) associated to each one of the unions.

The total degree of freedom that the hand possesses in its articulations are 23, plus six degrees of freedom that correspond to the axes x, y, z in the space where the hand can be moved, and the rotation upon each one of these axes. Then, we can say that the human hand articulation system contains 29 degrees of freedom.

Table I shows an image of each of the manual representations of the alphabet and the articulations of the fingers that should be flexed when representing these gestures. From this table we observe that there are a total of fifteen articulations that intervene to represent all the set of gestures. The device should sense these 15 degrees of freedom plus the coordinates x, y, z of the position of the hand in space with respect to the body of the user. The device to be designed should be able to sense all the parameters that characterize each gesture of the signed alphabet.

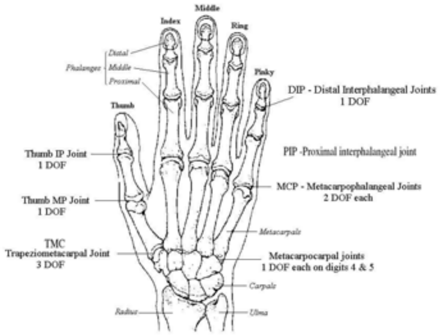
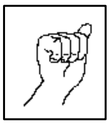
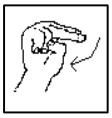
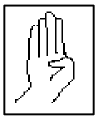
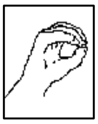
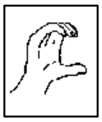
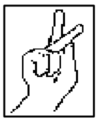
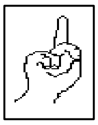
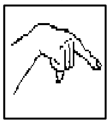
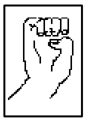
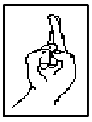


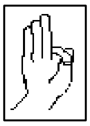
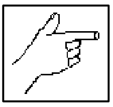
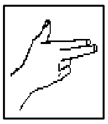
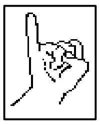
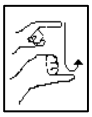
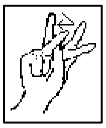
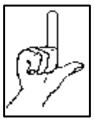
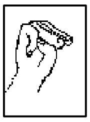
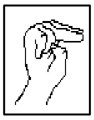
Figure 1. The human hand, the bones that compose it and the DOF belonging to each one of the joints

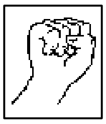
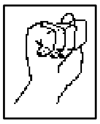
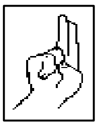
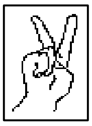
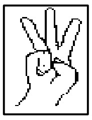
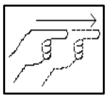
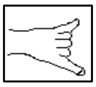
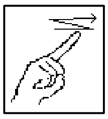
2.2 Joints position sensing

When representing a gesture, the position of the articulations defines the form of the hand, called hand posture. In the case of representations that do not involve movement, the posture defines by itself the gesture. For gestures that involve space-time features, the posture can vary through time initiating with a specific posture and changing to another to finish the gesture. This characteristic of change is present in two of the gestures of the alphabet (these are the gestures for character K and Ñ), as can be seen in table I. The other gestures that incorporate movement utilize a fixed posture and require the movement of the whole hand within the space facing the user (these gestures correspond to the letters J, X and Z).

Table I.
The Spanish sign alphabet and the joints flexed for each sign

	THUMB	INDEX	MIDDLE	RING	PINKY		THUMB	INDEX	MIDDLE	RING	PINKY
A						Ñ					
	*	DIP	DIP	DIP	DIP		*	*	*	DIP	DIP
	*	PIP	PIP	PIP	PIP		IP	*	*	PIP	PIP
	*	MCP	MCP	MCP	MCP		MCP	MCP	MCP	MCP	MCP
	*	*	*	*	*		TMC	*	*	*	*
B						O					
	*	*	*	*	*		*	DIP	DIP	DIP	DIP
	IP	*	*	*	*		IP	PIP	PIP	PIP	PIP
	MP	*	*	*	*		*	*	*	*	*
	TMC	*	*	*	*		TMC	*	*	*	*
C						P					
	*	DIP	DIP	DIP	DIP		*	*	*	*	*
	*	PIP	PIP	PIP	PIP		*	*	*	PIP	PIP
	*	*	*	*	*		*	*	MCP	MCP	MCP
	TMC	*	*	*	*		TMC	*	*	*	*
D						Q					
	*	*	*	DIP	DIP		*	*	DIP	DIP	DIP
	IP	*	*	PIP	PIP		IP	*	PIP	PIP	PIP
	MP	*	MCP	MCP	MCP		MP	*	MCP	MCP	MCP
	TMC	*	*	*	*		TMC	*	*	*	*
E						R					
	*	DIP	DIP	DIP	DIP		*	*	*	DIP	DIP
	IP	PIP	PIP	PIP	PIP		IP	*	*	PIP	PIP

	MP	*	*	*	*
	TMC	*	*	*	*
F					
	*	*	*	*	*
	*	PIP	*	*	*
	*	MCP	*	*	*
	*	*	*	*	*
G					
	*	*	*	*	*
	*	*	PIP	PIP	PIP
	*	*	MCP	MCP	MCP
	*	*	*	*	*
H					
	*	*	*	*	*
	*	*	*	PIP	PIP
	*	*	*	MCP	MCP
	*	*	*	*	*
I					
	*	DIP	DIP	DIP	*
	IP	PIP	PIP	PIP	*
	MP	MCP	MCP	MCP	*
	TMC	*	*	*	*
J					
	*	DIP	DIP	DIP	*
	IP	PIP	PIP	PIP	*
	MP	MCP	MCP	MCP	*
	TMC	*	*	*	*
	*	*	*	*	*
	*	*	*	PIP	PIP
	*	MCP	MCP	MCP	MCP
	TMC	*	*	*	*
L					
	*	*	*	*	*
	*	*	PIP	PIP	PIP
	*	*	MCP	MCP	MCP
	*	*	*	*	*
M					
	*	*	*	*	DIP
	IP	*	*	*	PIP
	MP	MCP	MCP	MCP	MCP
	TMC	*	*	*	*
N					
	*	*	*	DIP	DIP
	IP	*	*	PIP	PIP
	MP	MCP	MCP	MCP	MCP
	TMC	*	*	*	*

	MP	*	*	MCP	MCP
	TMC	*	*	*	*
S					
	*	DIP	DIP	DIP	DIP
	IP	PIP	PIP	PIP	PIP
	MP	MCP	MCP	MCP	MCP
	TMC	*	*	*	*
T					
	*	DIP	DIP	DIP	DIP
	*	PIP	PIP	PIP	PIP
	MP	MCP	MCP	MCP	MCP
	TMC	*	*	*	*
U					
	*	*	*	DIP	DIP
	IP	*	*	PIP	PIP
	MP	*	*	MCP	MCP
	TMC	*	*	*	*
V					
	*	*	*	DIP	DIP
	IP	*	*	PIP	PIP
	MP	*	*	MCP	MCP
	TMC	*	*	*	*
W					
	*	*	*	*	DIP
	IP	*	*	*	PIP
	MP	*	*	*	MCP
	TMC	*	*	*	*
X					
	*	DIP	DIP	DIP	DIP
	IP	PIP	PIP	PIP	PIP
	MP	*	MCP	MCP	MCP
	TMC	*	*	*	*
Y					
	*	*	*	*	*
	*	PIP	PIP	PIP	*
	*	MCP	MCP	MCP	*
	*	*	*	*	*
Z					
	*	*	DIP	DIP	DIP
	IP	*	PIP	PIP	PIP
	MP	*	MCP	MCP	MCP
	TMC	*	*	*	*

In order to detect the bending of the articulations and sense the position of the hand posture, a data-glove was constructed. The data-glove is based on ten flex sensors, which configuration was obtained after some testing with different number and position of sensors. The flex sensor is an electrical resistance that changes when bent, connecting it into a circuit provides a voltage related to the angle of articulation. Fig. 2 shows the form and position in which the sensors were positioned with respect to the articulations of the hand and upon the data-glove.

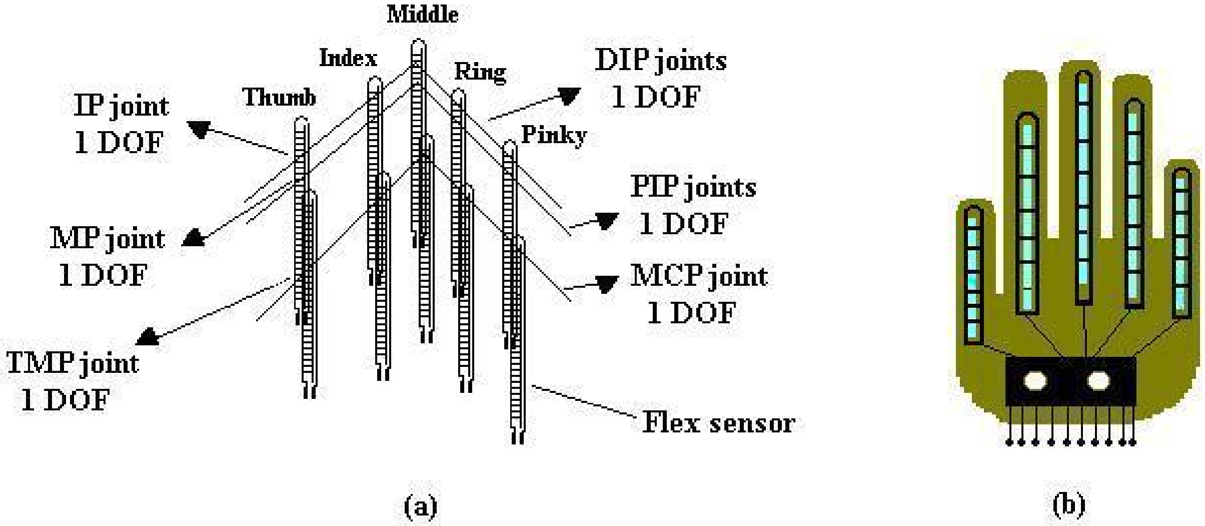


Figure 2. Hand posture sensing
(A) The ten flex sensors and their placement upon the articulations of the hand
(B) The data-glove

In Fig. 2, we can observe that the flex sensors are placed upon the articulations MCP, PIP and DIP of the pinky, ring, middle and index fingers, and upon the articulations TMP, MCP and IP of the thumb finger. It is important to note that the degree of bending in each one of these articulations is measured only toward the front of the hand and not to the sides, for this reason the abduction angle is not measured. The implemented glove is able to sense fifteen degrees of freedom, measured from fifteen articulations of the hand. The data-glove provides as response a vector of ten elements where each element is the voltage measured by one sensor. Two sensors for each finger are arranged, in order to provide enough information to the detecting device.

2.3 Tracking system

As mentioned in previous section, the movement is a very important feature of some gestures of the manual alphabet. For this reason the system has to measure the position *x, y, z* of the hand in front of the user and the movements executed when each one of the gestures is represented.

To integrate this capacity a tracker system was incorporated consisting of two ultrasound transmitters located in the data-glove and three receivers positioned in a triangular set in the clothes of the user. Fig. 3 shows the plane with origin (*0, 0, 0*) utilized as reference for locating the transmitters and receivers.

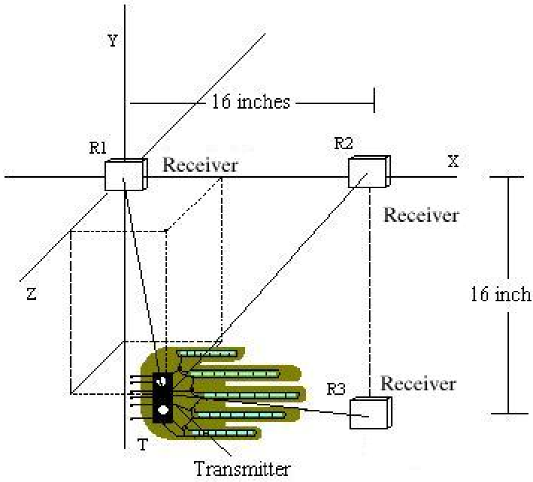


Figure 3. The tracking system consists of two ultrasound transmitters in the data-glove and three receivers located on a jacket wore by the user

Taking the position of the receiver R_1 as $x = 0, y = 0, z = 0$ (origin), the position of the receivers R_2 and R_3 are ($16, 0, 0$) and ($16, 16, 0$) respectively. To obtain the coordinates x, y, z of the data-glove, we do the following calculations:

- Each receiver R_1, R_2 and R_3 can be considered at the center of three spheres with radius given by the equations

$$R_1^2 = x^2 + y^2 + z^2 \tag{1}$$
$$R_2^2 = (x\text{-}16)^2 + y^2 + z^2 \tag{2}$$
$$R_3^2 = (x\text{-}16)^2 + (y\text{-}16)^2 + z^2 \tag{3}$$

- Using the location of the receivers, we can solve the equation system for the variables x, y and z, and obtain the equations 4, 5 and 6 that define the position x', y' and z' of the data-glove regarding to the origin situated in the receptor R_1.

$$x' = (R_1^2 - R_2^2 + 256)/32 \tag{4}$$
$$y' = (R_2^2 - R_3^2 + 256)/32 \tag{5}$$
$$z' = (R_1^2 - x'^2 - y'^2)^{1/2} \tag{6}$$

When the user starts a signed conversation using the data-glove and the tracker, the reference system coordinates are set up through a push button, positioning the hand in the area in front of the user, where the signs will be executed. The reference system remains fixed during all the conversation.

2.4 Signal treatment

In this section we analyze the acquisition and conditioning of the signals coming from the data-glove and the tracking system. The signals correspond to the value of the ten flex sensors and the three distances of the data-glove regarding to each one of the ultrasound receivers (R_1, R_2 and R_3). In total there are thirteen signals acquired by this module. The signals R_1, R_2 and R_3 have digital values, so we have to decode them to find the values x', y', z' of the data-glove's real position regarding to the ultrasound receivers. For the ten signals of the flex sensors, an analog to digital conversion is necessary. The conversion and decoding work is implemented using two programmable automatons of the microcontroller (MCU) kind. The MCU are programmable integrated circuits, each one contains internally an 8 bits microprocessor, four ports for parallel communication, a port for serial communication, 512 bytes of Random Access Memory (RAM), 2 Kbytes of Electrical Erasable and Programmable Read Only Memory (EEPROM) and an analog to digital signal converter (ADC) of eight channels with a resolution of 8 bits.

To capture the data from the ultrasound sensors the MCU should execute an initialization routine and maintain the synchronization with sensors. This requires a great part of the resources of memory and speed of the MCU. Due to this, and also that the MCU used has only eight analog to digital conversion inputs, and there are ten signals that need to be converted (the signals of the ten flex sensors), two MCU's were utilized. The tasks were distributed in both MCU in such a way that they work in parallel. A connection between them allows sending information with each other when needed. Fig. 4 shows the parallelization diagram of the MCU's.

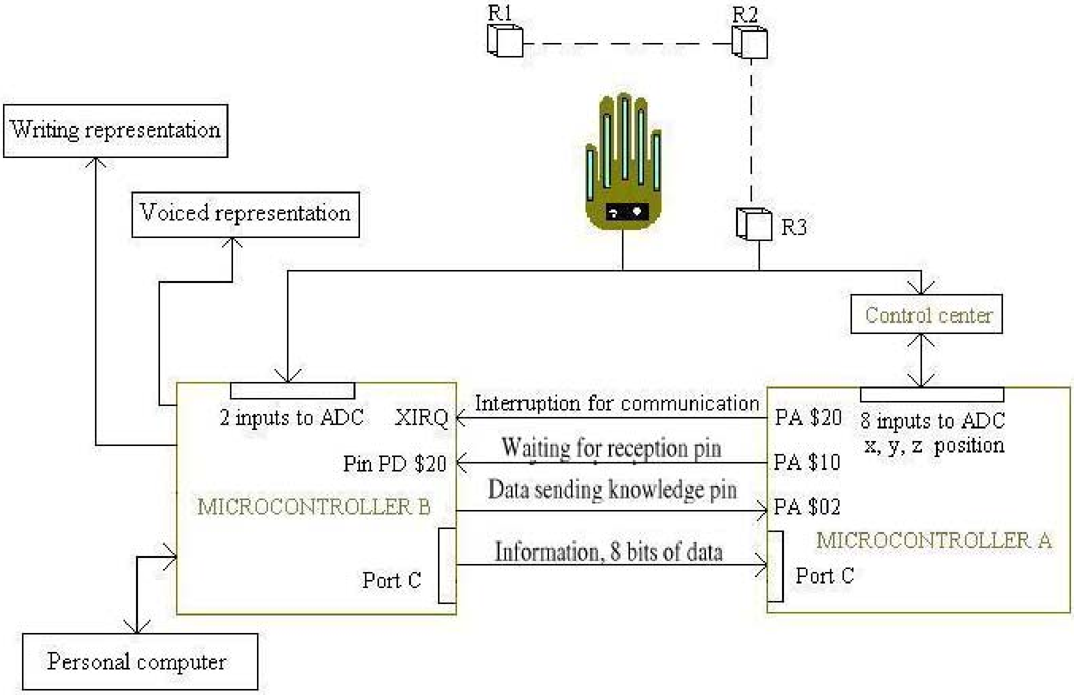


Figure 4. Parallelization diagram of the acquisition module

The main work of microcontroller B is to control the tracking system, capture and convert to digital the values of the signals pertaining to two flex sensors of the data-glove, and transfer this information to microcontroller A when requested. The microcontroller A captures and converts the signals of the other eight flex sensors of the data-glove, controls the written and voice representation system and transfers the information to the personal computer in order to update the complete state of the system. More powerful microcontrollers may also be employed. Regarding mobility, portability and computation density, the use of DSP's or FPGA's might be beneficial.

3. NEUROMAPPING FROM HAND GESTURES TO SPANISH ALPHABET

In order to do the mapping of the information generated by the instruments to the corresponding letters of the Spanish alphabet we make a distinction for two kinds of patterns. Those generated by static gestures and those patterns obtained by gestures that involve some movement of the hand (dynamics). For the static gestures, we only have to detect the posture of the hand. But for the dynamic ones, we have to detect not only the posture but the position changes of the user's hand regarding to his body during the interval of time in which the gesture is executed. From this distinction we have considered two types of data vectors. For the static patterns the data generated by the data-glove and the tracker are constant, while for the dynamic ones (during the time of execution), the data generated by the data-glove remains constant but data generated by the tracker changes (characters J, X and Y of the manual alphabet) or vice versa, the data from the data-glove varies but the position data remains constant (letters K and Ñ of the manual alphabet).

3.1 Data Visualization with a Kohonen network

Before designing the neural classifier to recognize the gestures, a cluster analysis of the pattern distribution was carried out looking for class overlap in the space of information generated by the instruments.

A set of 250 patterns was captured, corresponding to ten patterns for each one of the twenty-five postures that exist in the Spanish Sign Alphabet. This training set was modified later considering the results of the cluster analysis. Visualization of the class distribution offers the advantage of detecting problems of overlapping in data and allows to make adjustments to the gestures (if necessary) before designing the classifier.

The 250 pattern set captured was used to train a Kohonen neural network with the SOM (Self Organizing Map) standard algorithm [16]. A map of topology 10x25 (ten columns and twenty-five rows) was generated. The training was done in two phases, in the first phase of ordering, the reference vectors of the map's units were ordered. The algorithm was run iteratively for 10,000 times with a learning rate of 0.05 and a radius of 10 units. The second phase is adjustment, in which the reference vectors are adjusted to their precise values. To do this, the algorithm was run 20,000 times in an iterative form with a learning rate of 0.02 and a radius of two units. The resulting map is shown in Fig. 5a. From this figure can be noted that the gestures for letters C and O produce similar patterns presenting an overlapping problem. The same problem occurs for letters U and V, and G and L.

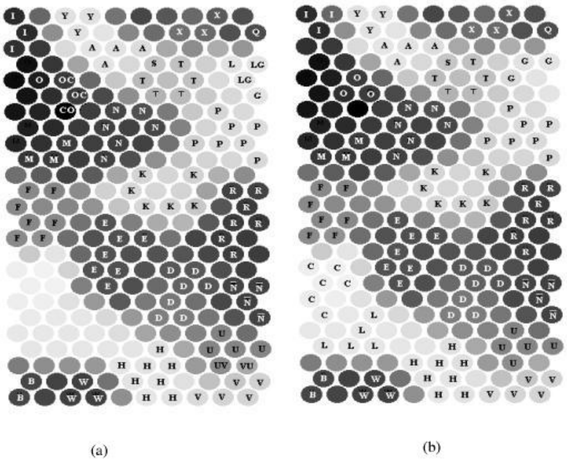


Figure 5. Self-Organizing Maps of the distribution of data generated with a Kohonen neural network
(a) Distribution map using the original set of gestures
(b) Distribution map after modification of C, U, V and L gestures

These problems appear because the data-glove constructed does not measure abduction angles (aperture angles between fingers). Due to this, the gestures that differ only in the degree of the abduction angles between fingers will produce equal or very similar patterns. To solve this problem we propose a modification to the gesture of the letter C (without affecting its basic form) in such a way that the MCP articulation of the Pinky, Ring, Middle and Index fingers do not have any inclination angle, as shown in Fig. 6.

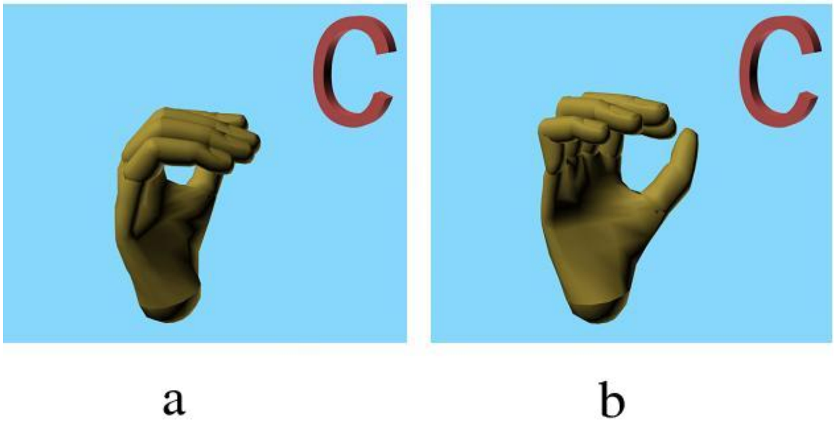


*Figure 6. Modification proposed to the gesture of the letter **C** to avoid the overlapping with the gesture of the letter **O** (a) Original gesture (b) Proposed gesture*

In the case of letters U and V, a modification in the TMC articulation of thumb finger in both gestures is proposed, as shown in Fig. 7.

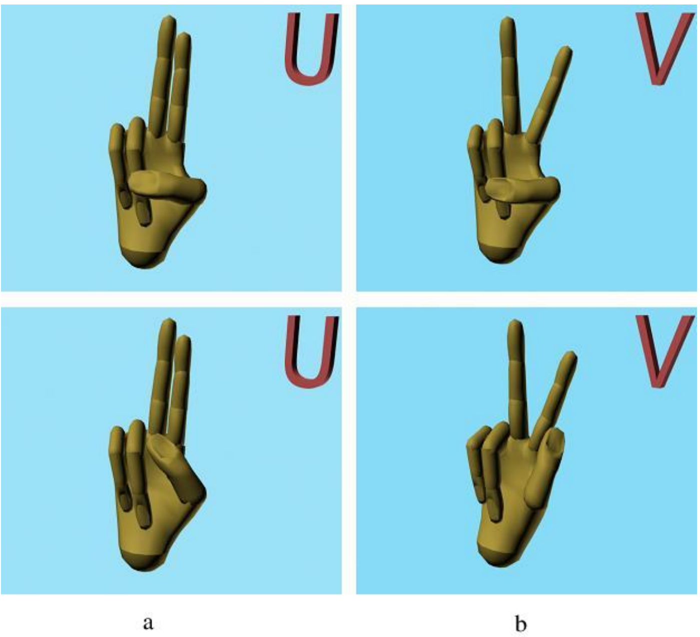


Figure 7. Modifications proposed to gestures of letters U and V to avoid the overlapping problem
(a) At the top the original gesture and at the bottom the proposed gesture for the letter U
(b) At the top the original gesture and at the bottom the proposed gesture for the letter V

For letters G and L, a modification to gesture for letter L (without affecting its basic form) was proposed. Now, the MCP articulation of the Pinky, Ring, Middle and Index fingers do not have any inclination angle, as shown in Fig. 8.

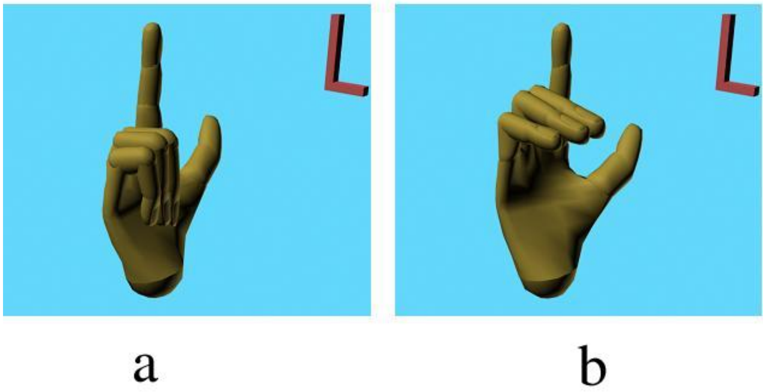


*Figure 8. Modification proposed to the gesture of letter **L** to avoid overlapping with the gesture of letter **G** (a) Original gesture (b) Proposed gesture*

With the new assembly of gestures, we captured another set of 250 patterns with the same structure of the previous one. Training with the new set, using the same parameters as before, a new Self-Organized Map was obtained. The image of the resultant map is shown in Fig. 5b, where it could be observed that the classes are now completely separated, so we have reduced the complexity of patterns.

Another alternative to solve the overlapping class problem is to modify the data-glove including a sensor to measure abduction angles between fingers. This is part of the future work to be explored.

3.2 Classification with Multilayer perceptron

The neural classifier considered was the Multilayer Perceptron trained with the Backpropagation algorithm [17]. The 250 patterns already captured were considered as the training set. Another set of 75 examples was captured and included for testing. This set incorporated three examples of each one of the postures to be recognized. Using this information a multilayer perceptron was trained utilizing the standard back propagation of errors algorithm [17]. The logistic activation function was incorporated for the hidden units and the error criteria considered was the Sum of Squared Errors (SSE) for all of the networks. After an extensive analysis, the topology of the network considered was of 10 input units, one hidden layer with 20 units and 25 output units. In Fig. 9, the graphics of the behavior of the SSE is shown. The black line indicates the error for the training set and the red line indicates the error for the test set. From the figure it is observed that in epoch 7500 the error lines are crossed; while the training error keeps descending the test error begins to stagnate. From the crossing point and ahead the network presents over training. Due to this behavior the network training was stopped at 7500 epochs. This procedure was sufficient to develop excellent classifiers as observed later on evaluations in section 7.

4. SYSTEM INTEGRATION MODEL

When integrating all modules, we should include the detection of the dynamic gestures (J, K, Ñ, X and Z). These gestures require a special treatment because the system should measure space-time characteristics to identify and represent them. In this case, the system does not have to provide a response to the stimulus of a single input pattern in time *t* but to an assembly of patterns that have to be presented in a specific order and in times t, t_1, t_2 etc. The system should analyze the context in which each input pattern is presented. An alternative to solve this problem is to use recurrent neural networks [18] or to implement in the system a module capable of detecting and analyzing the input pattern's context and take a decision about the response to be provided by the system. After some tests with recurrent neural networks we considered the last approach as described in next section.

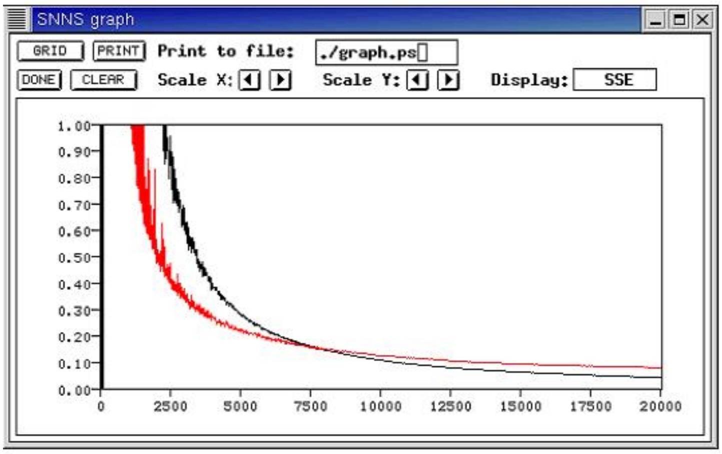


Figure 9. SSE errors behavior during training.
The black line indicates the error for the training set
and the red line indicates the error for the test set

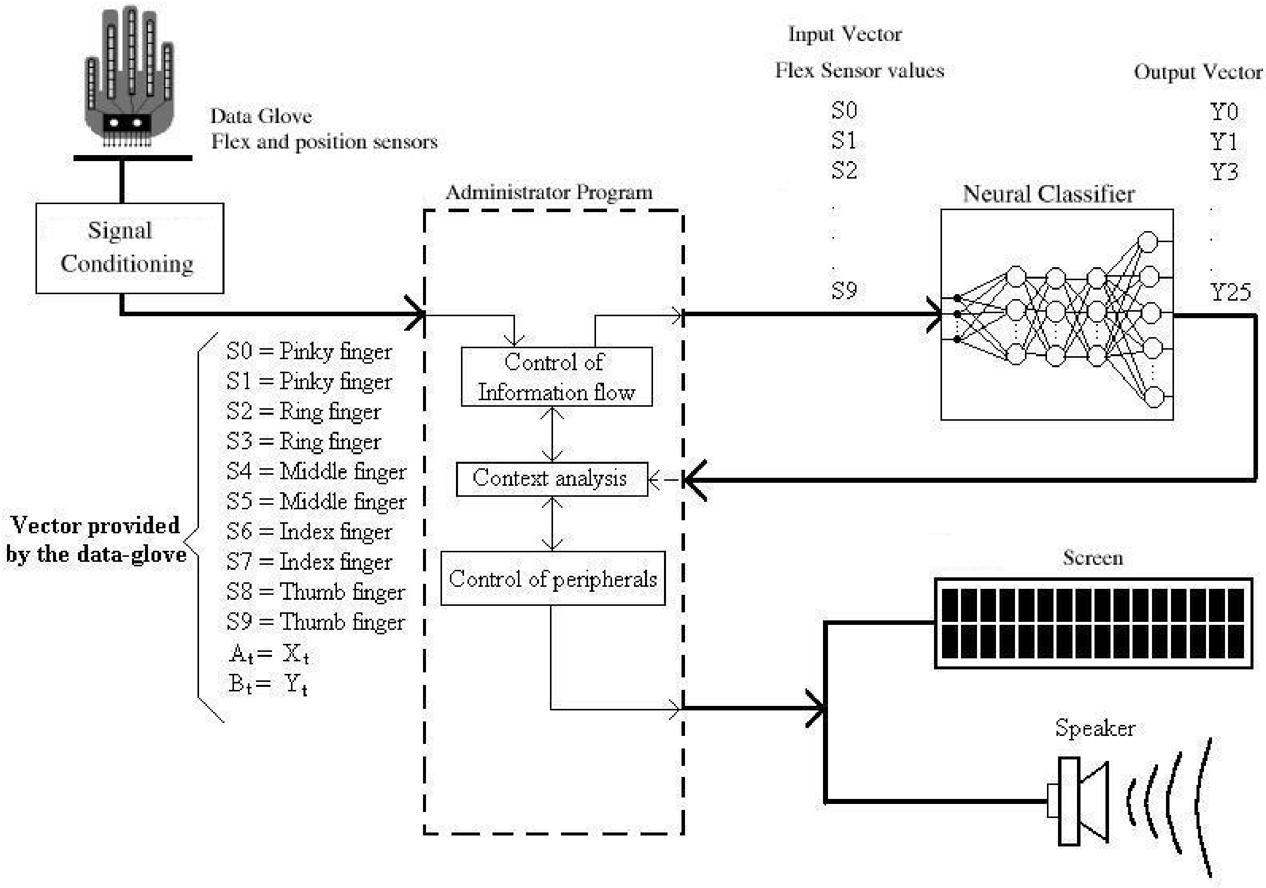


Figure 10. Schematic diagram of the integration model

4.1 The integration model

Fig. 10 shows the integration model. The information generated by the instruments is separated in two vectors. The first one corresponds to the posture of the hand and contains the values measured in the ten flex sensors (S_0, S_1, S_2, ..., S_9). This vector is sent to the multilayer perceptron, providing as a response one of the twenty-five classes that do not involve movement (Y_1, Y_2, Y_3, ... ,Y_{25}). The second one presents the position (x, y) of the user's hand regarding to his body, stored in a concatenated arrangement containing the previous sensed values for these variables. The arrangement corresponds to the sequence of positions (x, y) for the last readings taken (x, y in time [t], x, y in time [t-1], x, y in time [t-2], etc). This arrangement is compared with other ones containing the valid trajectories for dynamic gestures, and, upon finding a similar arrangement, a valid trajectory is found. The result of this operation, together with the posture (or sequence of postures), is processed by an administrator program. This program defines if, as a group, the valid trajectory and the posture(s) correspond to the representation of a static or dynamic gesture.

If a valid gesture is detected, the administrator program indicates at the output peripherals (display and speaker) to represent it in a visual and sonorous form.

4.2 Model implementation

To provide the system with a reference framework, a compatibility adjustment in the trajectories of dynamic gestures was carried out. This adjustment consisted in the following. Consider an imaginary two dimensional plane in front of the user. In this plane we define an origin (x=0, y = 0) and the regions represented by the quadrants (x = positive, y = positive; x = negative, y = positive; x = positive y = negative; x = negative y = negative), as shown in Fig. 11. Each time that a capture is done the hand of the user can be situated in a specific region of the plane. To detect the movements of the user's hand, we only have to detect a region trespassing during the capture period. The record of the region trespassing is stored on the concatenated arrangement that defines the trajectory that a gesture has followed. This record is utilized for context analysis.

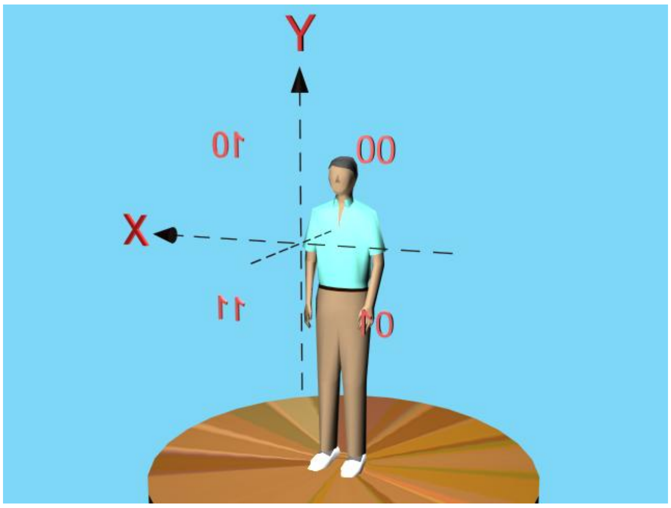


Figure 11. The picture shows a user and the imaginary plane

This compatibility adjustment constrains the quantity of movements that the gestures can execute. A small and finite number of arrangements containing all possible combinations of regions defined for dynamic gestures can then be obtained.

This representation permits the movements to be easily defined by the sequence and number of regions that a gesture should visit when executed. For example, if there are two regions that should be visited when executing a movement and the start position is the region {10}, then, there exist three possible combinations: moving from the region {10} to the region {00}, to the region {01} or to the region {11}. In this way, three different concatenated arrangements can define the complete movement. They are: {10}, {00} for the first gesture; {10}, {11} for the second one and {10}, {01} for the third gesture.

Under previous considerations, the administrator program was implemented. At the beginning, the first duty of the program is to communicate with the processing and acquisition module to obtain a vector of thirteen values indicating the updated state of the data-glove in posture and position. The program takes the information and splits in two distinct vectors, posture vector and position vector. The position vector is stored in memory, while the posture vector is sent to the neural classifier. The classifier provides as output one vector of twenty-five elements for each one of the postures existing in the Spanish Sign Alphabet. In case that the input is not a valid posture the program jumps to the beginning and starts the execution again. In case of being a valid posture the program takes the value of the region where the data-glove is found and stores it as the first value of the arrangement that will define if the gesture is static or dynamic. After this step, returns to transfer the state of the data-glove again. This process is executed five times in order to validate a posture before assuming that it is a valid gesture and represent it in sonorous and written form. If, before the five validations, the posture and the position have been changed, the program assumes that the user has not represented any gesture and the previous readings were occasional movements. Then, the program cleans all variables and returns to the beginning. If the posture is maintained fixed but the position changes during the period of validation, the values that define the position are taken and placed in the concatenated vector. The regions vector is compared with all the vectors of the same size stored in memory that define valid routes of dynamic gestures. If a stored vector of routes is found equal to the vector of regions generated by the movements of the data-glove, the position-movement set is verified to see if it corresponds to a valid gesture. In that case, the information is sent to the representation module to present the translation in written and sonorous form. In the contrary case, the program cleans all variables and returns to start the process again.

5. WRITTEN AND SPOKEN REPRESENTATIONS

This module was implemented with two distinct alternatives to represent the information. The first one uses the resources of the PC in which the administrator program was implemented. In this case, for the written representation, the screen of the PC is utilized and for the verbal representation the PC audio system is used. The second alternative utilizes a portable display to represent letters taking its ASCII code and an analog memory chip capable of storing sound and reproduce it through a speaker when required.

The first alternative is the simplest and the easiest to implement, because to do this it is only necessary to add to the administrator program the control libraries for the screen and speaker. This alternative is attractive when the software is developed for users that require learning and practicing the manual language. However, it has the disadvantage of being tied to a desk top computer. Then, the portability depends on the PC.

In order to make the system mobile, the alternative of the electronic devices was implemented (the portable display and the analog memory). With this, we only need to transfer the code of the administrator program to the language of the microcontrollers, obtaining a portable system.

5.1 Written representation

The written representation is implemented with a device of the DMC series from Optrex Corporation. This device consists of a Liquid Crystal Display (LCD) where the letters are projected. It contains an internal controller (based on a microprocessor), a character generator stored in Read Only Memory (ROM), and a Random Access Memory (RAM). It is controlled through an assembly of instructions that are supplied in an external way by an interface with the microprocessor or microcontroller.

One communication port from the Microcontroller B was used to control this device. For this microcontroller, an initialization and operation program was implemented in order to control the DMC device.

Once the administrator program has identified the sign represented by a user, it projects the corresponding letter in the PC screen and transmits it to microcontroller B to be projected in the LCD.

5.2 Spoken representation

A device of the ISD2500 series of Tandy Company was utilized to implement the verbal representation. It is an integrated circuit with the capacity to record and replay voice signals.

The corresponding sound of each of the letters of the alphabet was recorded in a specific address inside the device in such a way that, in order to reproduce the voice, we only have to indicate the address. This device addresses the sound and plays it in a speaker. It is controlled by microcontroller B, which upon receiving the information of the letter that has been represented by the user takes the corresponding address and sends it to ISD2500 to be reproduced by the speaker.

6. REAL TIME TRANSLATION FROM GESTURE TO VOICE AND TEXT

The complete process, involving the capture of the gesture, the treatment of signals, the processing of the information generated and the interpretation and representation of the corresponding pattern in the Spanish alphabet, should work in real time. This means that the system should work quickly enough to execute all of its tasks and respond immediately before the user represents another gesture.

In signed communication, the execution time of a gesture and the number of gestures that are represented in a specific time depends on the ability of the user. To provide our system with a real time response, we considered the occurrence of an event and established a period of time in which the system should provide an answer. The administrator program requests the updated information from the instruments each time that the established period of time is finished. The vector generated by the data-glove is sent to the multilayer perceptron, and if it corresponds to a valid posture the information is stored, otherwise the information is eliminated.

In the implementation, the system takes five readings to validate a gesture. If in the five readings, the position-movement arrangement corresponds to a valid gesture then the interpretation and representation of the gesture in the Signed alphabet is executed.

For the beginning users, the period of time necessary to request the information starts with 200 milliseconds; a gesture is recognized and represented in a second approximately. As the user is getting practice, this period of time can be reduced in such a form that, for a trained user, the time could be of 50 milliseconds approximately, which means that the user could execute around four gestures per second.

7. SISTEM EVALUATIONS

Four different topologies of the multilayer perceptron were implemented in order to evaluate the system. The adequate topology was selected to solve the classification problem. Subsequently, the integration model was evaluated. For this evaluation, the model was tested using the four classifiers designed, and the identification percentages for the twenty-seven gestures of the Spanish Sign Alphabet were obtained.

The topologies of the four multilayer perceptron implemented were: 10:5:25, 10:10:25, 10:20:25 and 10:30:25, where the first number indicates the units in the input layer, the second number corresponds to the units in the hidden layer and the third one to the units in the output layer. To evaluate the precision of classification and the percentage of samples not classified, 625 representations were executed (25 executions of each one of the postures) for each one of the designed classifiers.

The Table II shows in a concentrated form the results obtained during the evaluation process. The rows show the different precisions measured for both, the classifier and the complete integration model. The columns show the four different classifier topologies evaluated.

The last row of the table corresponds to the percentage of error due to other factors, it refers to the classification errors that were not persistent, present whether in the classifier or in the integration model but not in both. Those errors may be due to gestures that are not always represented in the same form by a user. For those gestures the patterns generated are not recognized by the system or are incorrectly recognized. Another possible cause can be that the data utilized to design and train the neural network is not sufficiently representative of the classes.

The errors induced by the model implementation can be assigned to specific details of the administrator program implementation. It integrates the parts of the system, capture the information and provide answers in real time. The fact, that in a unique process, patterns with static and dynamic features are analyzed can generate synchronization problems in the internal flow of information. These errors can cause that erroneous data are sent to the neural classifier. A possible solution for these problems is to implement the detection of postures and dynamic gestures in separated processes. In that way, the administrator program may be able to detect, at the beginning of each representation, whether a gesture is defined only by a posture or by a posture plus movement, and execute the adequate process for that type of patterns.

Table II
Percentages obtained during the system evaluation

	Topology 10:5:25	Topology 10:10:25	Topology 10:20:25	Topology 10:30:25
Classifier precision	92%	97.60%	98.88%	98.24%
Integration model precision	89.71%	92.18%	97.39%	95.88%
Average precision per class in the classifier	95.55%	99.11%	99.55%	99.23%
Average precision per class in the integration model	92.28%	98.23%	98.68%	97.79%
Samples not classified by the classifier	7.52%	1.44%	0.64%	0.96%
Samples not classified by the integration model	5.62%	2.60%	1.23%	1.23%
Classifier total error	8.0%	2.40%	1.12%	1.76%
Integration model total error	10.29%	7.82%	2.61%	4.12%
Consistent errors in the classifier	5.12%	1.60%	0.80%	0.96%
Consistent errors in the integration model	4.82%	1.52%	0.54%	1.77%
Percentage of error induced by the model	5.47%	6.30%	2.07%	2.35%
Percentage of error caused by other factors	4.28%	1.52%	0.54%	1.77%

8. SYSTEM WEARABILITY

Three important aspects were considered in the analysis of resources demanded by the system: memory space, capture and processing time, and energy consumption. The space required in memory by the system includes ROM and RAM memory that correspond to the space required for storage and running the program. The time required for data capture and processing should consider that the system will work in real time, but it is affected by the number of floating point operations to be executed per second. The energy consumption corresponds to the power consumed by the system in normal operation.

Table III
Resources required by system

RAM space	ROM space	Floating point operations	Required Power
58 Kb	2 Kb	14,900	6 watts

Table III shows in a summarized form the values found for the system working with the proposed integration model and using the 10:20:25 classifier. The 58 kb of ROM memory corresponds to the space required by the administrator, including the neural network, and the context analysis code that detect the movements. The 2 kb of RAM memory correspond to the space required by the program to be executed. The number of floating point operations (14,900) is the number of operations executed by the administrator program with a period of 200 milliseconds (an expert user).

The required resources in time and space are covered by any portable computer existing in the market. The implemented system consumes 500 milliamps with a power supply of 12 volts, which corresponds to a power of 6 watts, not considering the energy consumption of the PC. Nowadays there are in the market small batteries that comply with these requirement. Now, considering that all portable computers include their own battery, we can say that the system resources can be provided easily by commercial devices. Therefore, the developed system can become a wearable system.

9. FINAL REMARKS

The HCI developed system was evaluated utilizing four different topologies of multilayer perceptron neural networks. The best results were obtained by a neural network with an input layer of ten units, a hidden layer with twenty units and an output layer with twenty five units. The total classification precision obtained with this version was 97.39%.

The data-glove was developed with commercial flex sensors, reducing construction and maintenance costs of the system notably (around 300 dollars not including the computer). For a commercial glove, the costs easily rise to more than 3,000 dollars and maintenance may also be expensive. The incorporation of the ultrasound tracker for dynamic gestures detection allowed associating the twenty-five postures that do not involve movement to thirty-nine different trajectories. The system's vocabulary could then be expanded to more than 970 distinct gestures.

Programmable automatons that perform the capture and treatment of signals allow transferring the administrator program to the automaton's language, omitting the personal computer to perform the information processing.

Kohonen networks were used to visualize class distribution in the space of the instruments, allowing the detection of problems with six letters of the alphabet. Slight modifications of four gestures were proposed (without affecting their basic form) to separate pattern classes. Kohonen networks may also be used to visually verify the consistency in the sensors' response, by periodically generating a SOM to observe the pattern classes.

At the present time, the system has been adapted to detect the twenty seven letters of the Spanish alphabet, requiring only the use of one hand of the user. For future work, we consider to improve it by integrating the use of both hands in order to be able to recognize the ideograms of the Spanish Sign Language.

An important aspect of this system is the broad range of applications that may be considered. The PC implementation converts it in a valuable tool for teaching and practicing the signed alphabet. Also, due to simplicity and compactness, it can be a wearable system for impaired persons, serving them as a support tool for communicating with persons that do not understand the signed language.

10. ACKNOWLEDGMENTS

The first author was supported by CONACYT.

11. REFERENCES

[1] Pavlovic V. I., Rajeev S., Thomas S. H., Visual Interpretation of Hand Gestures for Human-computer Interaction: A Review, IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol. 19, No 7, pp. 677-695, 1997.

[2] Starner, T. & Pentland A. Visual Recognition of American Sign Language Using Hidden Markov Models. In: International Workshop on Automatic Face and Gesture Recognition (IWAFGR), Zurich, Switzerland, pp. 189-194, 1995.

[3] Fels S. S. and Hinton, G.E., Glove-Talk: A Neural Network Interface Between a Data-Glove and a Speech Synthesizer, IEEE Transactions on Neural Networks, Vol. 4, No. 1, pp. 2-8, 1993.

[4] Fels S. S. and Hinton, G.E., Glove–Talk II: A Neural-Network Interface which Maps Gestures to Parallel formant Speech Synthesizer Controls, IEEE Transactions on Neural Networks, Vol. 9, No. 1., pp. 205-212, 1998.

[5] Sujan V.A & Meggiolaro, M. A., Sign Language Recognition Using Competitive Learning in the HAVNET Neural Network, In Applications of Artificial Neural Networks in Imaging V (Electronic and Imaging 2000), N.M. Nasrabadi and A. K. Katsaggelos, editors, San Jose, CA., volume 3962 of Proc. SPIE, pp. 2-12, 2000.

[6] Kadous, M. W., Temporal Classification: Extending the Classification Paradigm to Multivariate Time Series, Ph.D. thesis, The University of New South Wales, School of Computer Science and Engineering, 2002.

[7] Barthelmess, P., Ensemble-based Human Communication Recognition, University of Colorado at Boulder Technical Report CU-CS-935-02, Department of Computer Science, 2002.

[8] Johnston T., Auslan: The Sign Language of the Australian Deaf Community, Ph.D. thesis, Department of Linguistics, University of Sydney, 1989.

[9] Moghaddam B. and Alex, P., Probabilistic Visual Learning for Objects Representation, IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol. 19, No. 7, pp. 696-710, 1997.

[10] Bobick A. F. & Wilson A. D., A State-Based Approach to the Representation and Recognition of Gestures, IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol. 19, No. 12, pp. 884-900, 1997.

[11] Zhao M., Francis K., Queck H., Wu X., RIEVL: Recursive Induction Learning Hand Gesture Recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 20, No. 11, pp. 1174-1185, 1998.

[12] Wilson A. D. & Aaron F. B., Parametric Hidden Markov Models for Gesture Recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol. 21, No. 9, pp. 884-900, 1999.

[13] Kramer J. & Leifer L., The Talking Glove: A Speaking Aid for Nonvocal Deaf and Deaf-Blind Individuals, in RESNA 12th Ann. Conf., pp. 471-472, New Orleans, Louisiana, 1989.

[14] Quam D. L., Gesture Recognition with a DataGlove, in IEEE National Aerospace and Electronics Conf., Vol. 2. pp. 755-760, 1990.

[15] Serafin-de-Fleischmann M. E., Lenguaje Manual, Aprendizaje de Español Signado para Personas Sordas, Editorial Trillas, 1996.

[16] Kohonen T., Self–Organizing Maps, Springer Series in Information Science, Vol. 30, Third Edition, 2001.

[17] Bishop C. M., Neural Networks for Pattern Recognition, Clarendon Press., Oxford, 1995.

[18] Lawrence S., Giles C. L. and Fong, S., Natural Language Grammatical Inference with Recurrent Neural Networks, IEEE Transactions on Knowledge and Data Engineering, Vol. 12, No. 1, pp. 126-140, 2000.

Authors' Biographies

Rafael Villa Angulo is a Computer Engineer from the UABC, Mexicali, Mexico, and obtained the M.Sc. degree in Computer Science from CICESE. He is a researcher at Instituto de Ingeniería, UABC in Mexicali, Mexico.

Hugo Hidalgo-Silva obtained the Electronics Engineer degree from the Instituto Tecnológico de Chihuahua in 1980, the M.Sc. degree in Electronics and Telecommunications from CICESE and a Dr. degree in Computer Science from CIMAT, Guanajuato, México. He is a researcher at Computer Science department in CICESE, where he works on Pattern Recognition and Neural Networks.