Sign Language Translator and Gesture Recognition

Conference Paper · June 2015

DOI: 10.1109/GSCIT.2015.7353332

4 authors, including:

Mohammed Elmahgiubi
University of Guelph

Nabil Drawil
University of Tripoli

Mohamed Samir Elbuni
University of Tripoli

Sign Language Translator and Gesture Recognition

Mohammed Elmahgiubi
School of Engineering
University of Guelph, Ontario, Canada

Mohamed Ennajar
National Information Security &
Safety Assurance, Tripoli, Libya

Nabil Drawil, and Mohamed Samir Elbuni
Computer Engineering, Faculty of Engineering
University of Tripoli, Tripoli, Libya

***Abstract*—The mute/deaf individuals have a communication problem dealing with other people. It is hard for such individuals to express what they want to say since sign language is not understandable by everyone. This paper is to develop a Data Acquisition and Control (DAC) system that translates the sign language into text that can be read by anyone. This system is called Sign Language Translator and Gesture Recognition. We developed a smart glove that captures the gesture of the hand and interprets these gestures into readable text. This text can be sent wirelessly to a smart phone or shown in an embedded LCD display. It is evident from the experimental results that gestures can be captured by set of inexpensive sensors, which measure the positions and the orientation of the fingers. The current version of the system is able to interpret 20 out of 26 letters with a recognition accuracy of 96%.**

I. Introduction

Sign-language, or hand-speak, has become a popular method of communicating for those who cannot verbally speak. The sign language is a language that uses the hand movements to express words and alphabets. People who are using the sign language were estimated just in china alone to exceed 80 million and those people always have a problem communicating with others who cannot understand the sign language.

For the purpose of this research work and the use of English language for experimentation within this work, we used the American Sign Language (ASL), due to its popularity and major use globally. However the use of any other finger spelling movements representing other alphabets can be easily adopted. As other sign Languages, ASL represents phonemics using expressions and movements in the face, the upper body and the hands. Word representation in ASL is usually done by loanwords from English.

As a detailed subject, Sign language has various components that can be studied collectively or individually in conjunction with computer engineering especially within the context of. various methods and research that deals with facial recognition and body movements, this is due to the fact that computers and facial recognition can play major role in supporting deaf disabled to communicate with intelligent machines.

However within the context of this research work, we will only be concerned with hand spelling component of the ASL, which is considered as a first step in such translation system.

Hand spelling refers to the use of the various components of a hand (such as fingers, hand position and tilting, etc.) to represent alphabet, numerals and special characters, [1]. As of May 2013 the latest and most updated hand spelling representation of the alphabet by the ASL is shown in Figure 1.

From the Hand spelling representation of the alphabet shown, it is very clear the dependency of the ASL on the fingers' gesture and the hand position and twist. It is also quite clear that multiple fingers are used to represent different characters of the alphabet. Accordingly and to represent all the alphabets, the gesture of each finger has to be captured and the overall all presentation of the fingers and the hand will represent the alphabet.

To accomplish this task,the sign language translator we have developed uses a glove fitted with sensors that can interpret the (ASL) alphabet. The glove uses flex sensors, contact sensors and 6 DOF accelerometer/Gyroscope on a single chip. All these sensors are mounted on the hand to gather the data on every finger's position and the orientation of the hand to differentiate the letters. Sensory data is sent to a computing unit (an arduino microcontroller) to be translated and then displayed.

The remainder part of this paper is organized as follows. Section II explores some pieces of related research work. Section III introduces the proposed sign language translator along with the design constraints. Section IV elaborates on the hardware design, and also explains the software part of the design. Section V shows the results of the experimental work and some evaluating measures. Section VI provides concluding remarks.

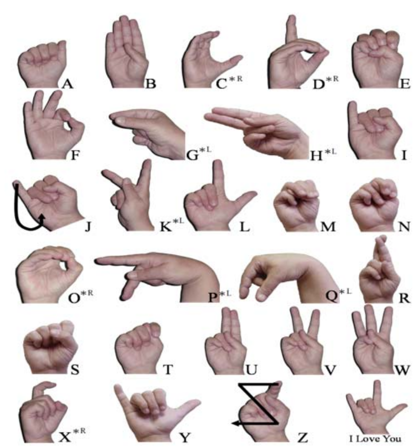

Fig. 1. The American manual alphabet, [1].

978-1-4673-6587-1/15/$31.00 ©2015 IEEE

II. Previous Work

The ability to track a person's movements and determine what gestures they may be performing can be achieved through various tools [2], [3]. There were several attempts to resolve this problem and there were a large amount of research done in image/video based gesture recognition consequently there was some variation within the tools and environments used between implementations.

A. Wired Gloves

These can provide input to the computer about the position and rotation of the hands using magnetic or inertial tracking devices. The first commercially available hand-tracking glove-type device was the Data Glove, a glove-type device which could detect hand position, movement and finger bending. This uses fiber optic cables running down the back of the hand. Light pulses are created and when the fingers are bent, light leaks through small cracks and the loss is registered, giving an approximation of the hand position and they were a little bit expensive costing more than 5000$ apiece [4].

B. Depth-Aware Cameras

Using specialized cameras such as structured light or time-of-flight cameras, one can generate a depth map of what is being seen through the camera at a short range, and use this data to approximate a 3d representation of what is being seen.

C. Stereo Cameras

Using two cameras whose relations to one another are known, a 3d representation can be approximated by the output of the cameras. To get the cameras' relations, one can use a positioning reference such as a lexian-stripe or infrared emitters In combination with direct motion measurement (6D Vision) gestures can directly be detected.

D. Single Camera

A standard 2D camera can be used for gesture recognition where the resources/environment would not be convenient for other forms of image-based recognition. A Software-based gesture recognition technology using a standard 2D camera that can detect robust hand gestures, hand signs, as well as track hands or fingertip.

III. Proposed ASL Translator System

This paper proposes an ASL translation system that is performance and cost effective. In order to achieve this goal, we set a number of constraints on the proposed system. The following list summarize these design constraints:

- System must be simple so anyone can use it.
- System must be compact and portable.
- System must be inexpensive financially.
- System must be able to integrate with today's gadgets such as smart-phones.

To design a system that comply with the aforementioned requirements, a number of main components are proposed. The data acquisition system represented by the glove capturing finger and hand gestures, the microcontroller system, which carries the intelligence to read and interpret the signals and produces the output, the display system which is an LCD directly attached to the microcontroller or a Bluetooth capable device such as smart phone that can dispaly wirelessly received information.

A. Further Details on The Components

The target is to mount most of the components on an elastic glove that gathers all the hand's gestures and translates these gestures (sign language) to be displayed in form of letters. The letters are displayed on an LCD wired display and/or transmitted to a smart phone wirelessly.

1) The Glove: The sign language translator starts with the glove. It is an elastic glove that contains 5 flex sensors, 5 contact sensors, 1 three dimensions accelerometer (a_X, a_Y, a_Z) and 1 three dimensions gyroscope (g_X, g_Y, g_Z).

The flex sensors are attached to the back of each finger. This gives a free movement for each finger and a separate readings. The contact or force sensors determine which fingers are touching and how the fingers are relative to each other.

The Accelerometer/Gyroscope are used for the movement and orientation detection and it is mounted at the upper side of the hand.

2) Microcontroller Unit (MCU/ARDUINO) : The MCU consists of an Arduino mega development board that has an atmega 2560 AVR microcontroller [5]. The Arduino is the part of the design that acts as the brain of the system, to which all other parts are directly connected.

3) The Display: This part function is to display the interpreted letters. The system has two types of display: an embedded LCD screen display, and a wireless display via a Bluetooth powered smart phone. The latter must have a Bluetooth emulator app that is available free of charge at any app store.

4) The Wireless Transmission: To enable wirless transmission between the controller and a smart phone, a Bluetooth module is chosen. The selection of Bluetooth module is based on its capability of being easy to use, and its wide compatibility with today's gadgets. A serial Bluetooth module device is utilized in this work. It is programmed with the AT commands and it uses the commonly used serial 9600 8N1 protocol, i.e., baud rate of 9600bps, a chunk of 8 bits of data sent at a time with no parity bits, and a 1 start/stop bit.

IV. Implementation of ASL Translator

The ASL translator system implementation can not be accomplished without tackling both the sides: the hardware and the software. We first present the hardware part of the design in the following, and then we describe the software part of the design.

A. Hardware Design

Basically the hardware is comprised of a number of sensors, Arduino microcontroller, and the output/display device(s). Next, we demonstrate further details about the features of the used hardware components, their installation, and their physical wire connections.

1) Sensors of Gestures' Capturing: Three types of sensors are used here to capture the hand gesture. The first type of the sensor capture, the part at which all the hand gesture and finger movement are sensed, is consists of five flex sensors along with five force sensors one of each flex sensors. These sensors are fixed on top of an elastic glove that enables the fingers to freely move and bend. Every flex sensor is mounted on top of each finger and each of 4 force sensors is mounted at the bottom of each finger.

These sensors provide measurements about the degree of a bent finger or fingers touching another finger or certain are in the palm. Therefore, all flex sensors are connected to a voltage dividing resistors and powered by a 5 Volt dc, Figure 2. A ratio of the 0 to 5 voltage is then measured then mapped in the range of 0 to 1023 integer levels at the MCU part, through its analog ports.

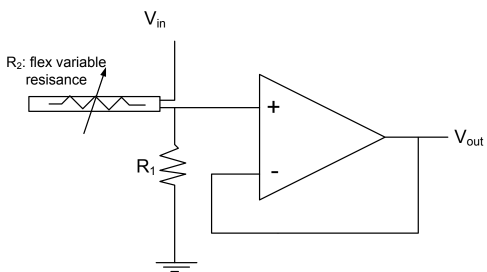

Fig. 2. Voltage divider circuit

The idea hear is to transduce the value of the flex variable resistance to voltage that indicates the bend angle. This voltage can be read by the MCU, Figure 3. The formula below shows how a sensor resistance is transformed to voltage:

$$Vout = V_{in} \times R_1/(R_1 + R_2) \tag{1}$$

$V_{in} = 5$ Volts.

R_1 : the flex sensor resistance in ohms Ω.

$R_2 = 10k\Omega$.

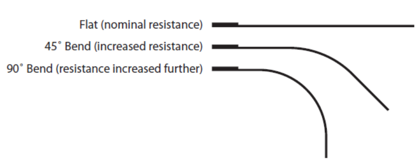

Fig. 3. Flex bend levels

The second type of sensors used in gesture capturing is contact sensors. The contact sensors provide a binary type of measurement, i.e., touching or not. The way we connected those sensors is that we made a series connection between the 5 volt supply and one of the sensor's terminals and then all the way to the MCU's analog input. The contact sensor is a variable resistor too. But the resistance of this sensor is decreased exponentially as more presser is applied. It is thus, we used these sensors to sense any finger touches.

The third type of sensors is Accelerometer. The MPU6050 is a 6 degree of freedom chip from Invensense company, containing a tri axis accelerometer and a tri axis gyroscope. It is operated off a 3.3V supply, and communicates via I2C serial protocol at a maximum speed of 400 kHz. The accelerometer measurement is read by the MCU using 16 ADC.

2) The Processing Part: An MCU board is used as a processing unit, which is a standalone Arduino mega development board that receives the sensor's analog readings through its 15 analog terminals, pass them to its 16 channels 10 bits Analog to digital converter to digitize them and then take those raw readings to be processed. The pin out of the Arduino mega development board can be checked out in [5]. All the voltage divider's circuits are then connected to the analog ports 0 to 4 and the touch sensor's outputs are connected to the rest of the analog pins and the accelerometer is connected to the I2C serial pins SDA and SCL which represents the serial data and serial clock.

3) The Display: This is the most important part to the end user. Its function is to display the interpreted letters. The system has two types of display a wired LCD screen display attached to the MCU part and a wireless display: a Bluetooth powered smart-phone. The latter must have a Bluetooth emulator app that is available free of charge at any app store.

4) The Wireless Transmission: This is a Bluetooth module that is programmed to send the data to the smart phone. The Bluetooth module we used is an HC-05. These modems work as a serial (RX/TX) pipe. Any serial stream from 9600 to 115200bps can be passed seamlessly using two inputs in Port0 of the MCU. it can be powered by a 3.3v power supply, it uses a 2.4GHz frequency ISM band, and can transmit data at speeds of unstable 150Kbps to 2Mbps asynchronously and a stable 1Mbps synchronously. The reason behind choosing these modules is due to their least expensive price in the market while maintaining competing performance. Also, they are suitable for prototyping with very low power consumption due to the EDR technology used in them, in addition, they use a UART interface with a programmable baud rate and have a built in antenna.

B. Software Design

The software part of the system tackles two pivotal aspects in the design. The first aspect revolves around the management of the system operation including hardware initialization and sensory data quantization. The second aspect concerns with the process of gesture artificial learning. The latter is explained below.

1) Gesture Learning Process: At every time the system powers up the user will be asked to open his/her hand for a few seconds then close the hand for few more seconds. In each case the sensory measurements are recorded for the purpose of calibration.

When the user is asked to open his hand the system measures the minimum values of fingers flexes, and then he/she will be asked again to close his hand so the system measures the maximum values of fingers bending. The system then takes those minimum and maximum values and map them in the range of 0 to 200, where 0 is the hand's fingers being all flexed and 200 is the hand's fingers being completely bent. After calibrating the flex sensors the system will be ready to estimate the status of every finger whether it is bent.

The next step in the learning process is to store an average values for every sensor reading when certain letter is signed. Accordingly, when a letter is being signed the system checks for all the 5 flex sensors values. If the values of the sensors matches with the range's values already stored in the system then the matching letter will be displayed. Otherwise, the system does nothing but continue reading the sensors until a valid letter is recognized.

In some cases the system does not define the letter (by only the flex sensors) due to similarity in the fingers flexes between different letters. Hence, additional sensors including 5 contact sensors and an accelerometer are used to extend the measurement space by dimensions where similar letters can be differentiated.

Table I depicts the various threshold levels of the flex sensors for each letter and the different touch values and hand axis orientation that was stored in system to differentiate letters. Due to the lack of some of the hardware components, out of 26 letters, only 6 are not included in the current version of this research work.

TABLE I
RANGES OF SENSORY DATA.

Letter	Flex Sensors					Touch Sensors					Accelerometer		
	$Sensor_1$	$Sensor_2$	$Sensor_3$	$Sensor_4$	$Sensor_5$	$Sensor_1$	$Sensor_2$	$Sensor_3$	$Sensor_4$	$Sensor_5$	X_{axis}	Y_{axis}	Z_{axis}
A	> 100	> 100	> 100	> 100	< 50	-	-	-	-	-			
B	< 50	< 50	< 50	< 50	> 100	-	-	-	-	-			
C	> 100	> 100	> 100	> 100	> 100	No	No	-	-	-			
D	$[100 - 200]$	< 50	> 80	> 80	> 80	-	-	-	-	-			
E	Nil	Nil	Nil	Nil	Nil	Nil	Nil	Nil	Nil	Nil	Nil	Nil	Nil
F	< 50	> 90	< 50	< 50	> 100	-	-	-	-	-			
G	> 100	< 0	> 100	> 100	$[0 - 6-]$	-	-	-	-	-			< 9000
H	> 100	< 50	> 100	< 50	> 100	-	-	-	-	Yes	¿0		
I	< 50	> 50	> 100	> 100	> 100	-	-	-	-	-			> 0
J	< 50	> 50	> 100	> 100	> 100	-	-	-	-	-			< 0
K	> 100	< 50	> 100	< 50	< 50	-	-	-	-	-			< 0
L	> 100	< 50	> 100	> 100	< 50	-	-	-	-	-			
M	Nil	Nil	Nil	Nil	Nil	Nil	Nil	Nil	Nil	Nil	Nil	Nil	Nil
N	Nil	Nil	Nil	Nil	Nil	Nil	Nil	Nil	Nil	Nil	Nil	Nil	Nil
O	$[100 - 150]$	$[100 - 150]$	$[100 - 150]$	$[100 - 150]$	$[100 - 150]$	Yes	Yes	-	-	-			
P	> 100	< 50	> 100	< 50	< 50	-	-	-	-	-			> 0
Q	> 100	< 0	> 100	> 100	$[0 - 60]$	-	-	-	-	-			> 10000
R	> 100	< 50	> 100	< 50	> 100	-	-	Yes	-	-			
S	Nil	Nil	Nil	Nil	Nil	Nil	Nil	Nil	Nil	Nil	Nil	Nil	Nil
T	Nil	Nil	Nil	Nil	Nil	Nil	Nil	Nil	Nil	Nil	Nil	Nil	Nil
U	> 100	< 50	> 100	< 50	> 100	-	-	-	-	Yes	< 0		
V	> 100	< 50	> 100	< 50	> 100	-	-	No	-	No			
W	> 100	< 50	< 50	< 50	> 100	-	-	-	-	-			
X	> 100	$[50 - 100]$	> 100	> 100	> 100	-	-	-	-	-			
Y	< 50	> 100	> 100	> 100	< 50	-	-	-	-	-			
Z	Nil	Nil	Nil	Nil	Nil	Nil	Nil	Nil	Nil	Nil	Nil	Nil	Nil

After determining which letter is gestured the recognized letter is transmitted through the system's serial port to the Bluetooth and from there to the desired smart phone. Another method of displaying the recognized letter is the LCD display, which is wired with the system circuitry.

2) System Operation: This section presents the approaches and algorithms developed to manage the operation of the ASL translator system. Codes written to operate Arduino are known as sketches. They are written in C++. Every sketch needs two void type functions, setup and loop. The setup method runs at the beginning after booting the Arduino, and the loop method runs continuously afterward.

Therefore, hardware and software initialization steps are coded in the setup function. In the loop function, the regular routines of the recognition are coded to run over and over again.

Arduino IDE is the integrated developing environment used in this work. The Arduino development environment contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions, and a series of menus. It connects to the Arduino hardware to upload programs and communicate with them. In the following, further details about the sketch are provided.

The Setup Part: The setup part in our software includes all the initializations in the code, defining the variables and their types, initializing the LCD display, initializing the serial ports of the Arduino, initializing the accelerometer and gyroscope, along with printing the welcoming message when powering up the device. Before the system start working, a calibrations step must be done for the flex sensors to diminish their sensitivity to changes in the environment. The main initializations steps are shown the following pseudocode.

Algorithm 1 Pseudocode for setup function

```
Input: void;
Output: void;
step1: Include libraries;
step2: Define variables;
step3: Define I/O pins;
step4: Initialize serial COM port;
step5: Initialize LCD display;
step6: Initialize accelerometer and gyroscope;
/*Calibrate flex sensors*/
for i = 1 : 5 do
    Calibrate flex(i);
end for
return
```

This part in our software also includes all the libraries that we have used in our sketch and all these libraries are free licensed libraries and open source ones. The libraries used were the software serial library, liquid crystal library, the wire library, I2Cdev library and the MPU6050 library.

The Loop Part: The loop function basically contains two code segments. The first segment deals with reading the sensory/raw data and performing some filtration using parameters extracted from the calibration step. In the second segment, the recognition precess is implemented where the input is the rectified measurements and the output is the gesture recognition decision. This decision is made based on inexpensive and simple criteria that enforces a set of rules obtained from Table I to exclude the dissimilar gestures. After recognizing the gesture, the corresponding letter is printed out either serially through the Bluetooth or in parallel through the LCD's 8 bit bus display. The gesture recognition and display flowcharts are shown in Figure 4.

V. TESTING AND RESULTS

Section III demonstrates the constraints and requirements for the proposed translation system. In this section we show

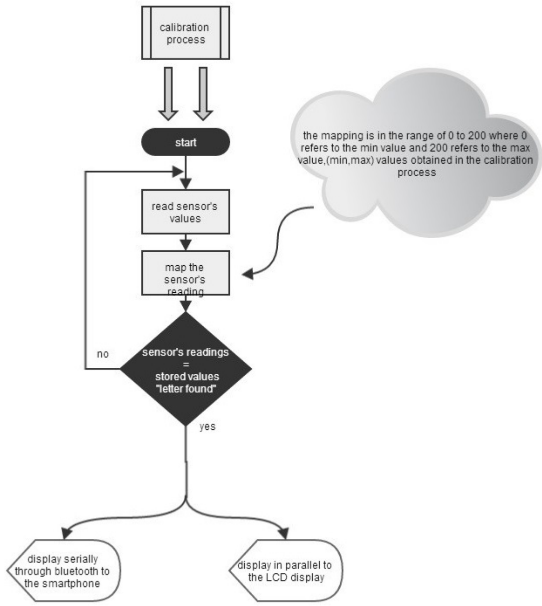

Fig. 4. The detection, recognition and display process (module)

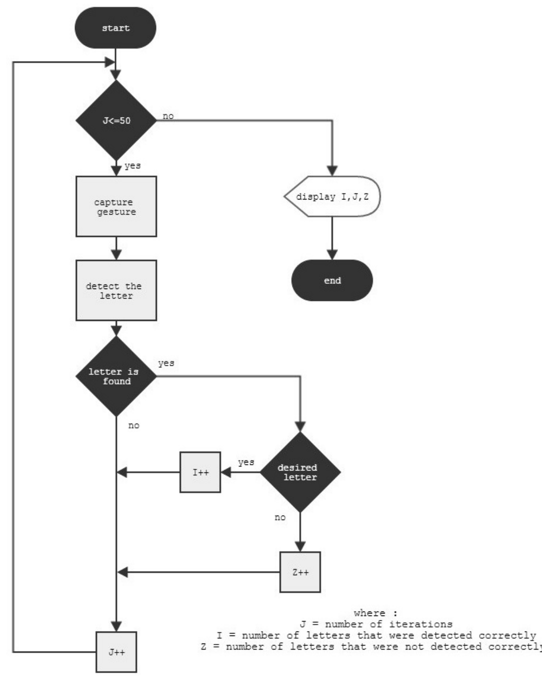

Fig. 5. The testing procedure flowchart

how far we have accomplished in this research work, and how much we have satisfied of the aforementioned requirements. The system evaluation is demonstrated in terms of number of implemented letters and percentage of recognition. The experimental results are provided below in this section.

A. Experimental work

After assembling the hardware and loading the desired code into our chip we have done iterative tests to make sure that we have met the desired requirements. the Performance measures are in terms of reading's error rate at the first stage.

The Flow chart in Figure 5 demonstrates the approach taken to measure the system performance metrics. We tested each letter individually, where 50 iterations are applied on each letter. The frequency of recognizing any letter is stored in the table shown below. However, the recognition might be erroneous in the sense that the letter gestured is not the recognized. Such erroneous result is shown in the table, as well.

Table II depicts two types of recognition: not detected and detected wrong. The second error type exposes the ambiguity issue in the recognition process. This ambiguity happens when two or more gestures have similar finger positions or moves – as a result flex sensors can not do much in such cases. For example, some letters are mistaken for others like the letter "v" and "p" where the values of their flexes are relatively similar to each other. Compare the two gestures pertained to the two letters in Figure 1. All ambiguity cases are mentioned later in this section.

The accuracy and error rates are calculated using the following equations.

$$\text{Accuracy } \% = \frac{\text{detected right}}{\text{Num. of iterations}} * 100 \quad (2)$$

$$\text{Error of wrong readings } \% = \frac{\text{detected wrong}}{\text{Num. of iterations}} * 100 \quad (3)$$

$$\text{Not detected letters} \% = \frac{\text{not detected}}{\text{Num. of iterations}} * 100 \quad (4)$$

Figure 6 shows the accuracy and the error as a percentage of the total number of iterations.

B. Ambiguity Cases

In some cases, similarity between measurements pertained to different letters is very high. This misleading similarity causes wrong detection as seen in the table above, we call it ambiguity. This issue is known as identification problem. Here we present the ambiguity cases along with proposed solutions for each case we had.

Case 1: The letters "C" and "O" have all fingers bent but in the letter "O" the bending is slight more than the "C" letter so we avoided the mistake of one of those letters being misspelled by adding the touch sensor at the index and thumb finger and adding a new filter to the recognition of the letters in the software which is having those two sensors be a zero value meaning they are touching each other for the letter "O".

Case 2: The letter "G" and "L" where both letters have the same three fingers bent and the other two being unbent but in the letter "G" the two unbent fingers are not as stretched as in the letter "L". The proposed solution was adding another rule in the software since the flex sensor can differentiate the flex's levels.

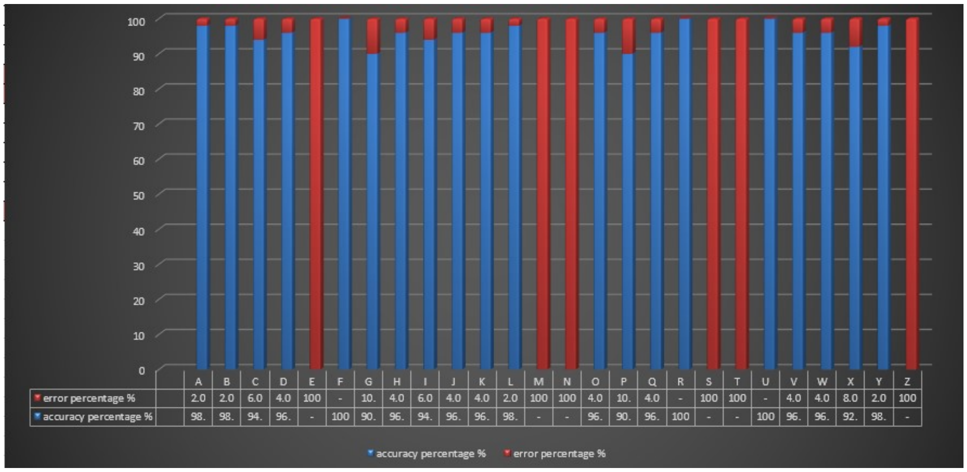

Fig. 6. Accuracy and error percentage

TABLE II
ACCURACY AND ERROR VALUES.

Letter	Detected		Not Detected
	Correct	wrong	
A	49	0	1
B	49	0	1
C	47	1	2
D	48	1	1
E	Nil	Nil	Nil
F	50	0	0
G	45	4	1
H	48	1	1
I	47	2	1
J	48	2	0
K	48	1	1
L	49	0	1
M	Nil	Nil	Nil
N	Nil	Nil	Nil
O	48	0	2
P	45	4	1
Q	48	1	1
R	50	0	0
S	Nil	Nil	Nil
T	Nil	Nil	Nil
U	50	0	0
V	48	2	0
W	48	0	2
X	46	2	2
Y	49	0	1
Z	Nil	Nil	Nil

***Case 3**:* The letters "U" and "V" have both the index and middle fingers completely stretched but the difference is that the distance between the index and middle finger in the letter "V" is more than at the letter "U". In this case the flex sensor provide similar readings for both letters. This issue is solved by adding touch sensors between the two fingers so when the letter "U" is signed the touch sensor should give a reading of zero making it possible to detect the difference.

***Case 4**:* This case is similar to the previous case, but this time the letters to be differentiated are letter "V" and letter "P". Fingers' gesture for both letters are the same where but the middle finger is forwarded a little bit to the front and the thumb is touching the lower area between the middle and index fingers. A proposed solution can be in terms of adding a small touch sensitive sensor to that area.

VI. CONCLUSION

This paper presents an automatic hand-sign language translator – a critical system for mute/deaf individuals. Expected requirements and level of performances of such system are addressed here. The paper list system components and extends the explanations of hardware wire connections and components assembly. Furthermore, the software part of this system is extensively elaborated to include simple system initialization and recognition algorithms. The paper addresses the challenges of identifying ambiguous measurements and proposes respective technical solutions.

It is evident from the experimental results that the system has the potential to help targeted individuals and communities. Especially that the system was able to recognize most of the letters (20 out of 26), and get to an average accuracy of 96%. We look forward to add the remaining letters to the system and better the system performance.

REFERENCES

[1] en.wikipedia.org, http://en.wikipedia.org/wiki/Fingerspelling, accessed: 3/Jul/2012.
[2] N. El-Bendary, H. Zawbaa, M. Daoud, A. ella Hassanien, and K. Nakamatsu, "Arslat: Arabic sign language alphabets translator," in *Computer Information Systems and Industrial Management Applications (CISIM), 2010 International Conference on*, Oct 2010, pp. 590–595.
[3] M. Melnyk, V. Shadrova, and B. Karwatsky, "Article: Towards computer assisted international sign language recognition system: A systematic survey," *International Journal of Computer Applications*, vol. 89, no. 17, pp. 44–51, March 2014, published by Foundation of Computer Science, New York, USA.
[4] K. Ellis and J. C. Barca, "Exploring Sensor Gloves for Teaching Children Sign Language," *Advances in Human-Computer Interaction*, p. 8, 2012.
[5] www.arduino.cc, http://www.arduino.cc/en/Guide/Introduction, accessed: 3/Jul/2012.