Development of a Sign Language Translator Using Simplified Tilt, Flex and Contact Sensor Modules

Quiapo, Carlos Emmanuel A. and Ramos, Katrina Nicole M.
Ateneo de Manila University
School of Science and Engineering
Quezon City, Philippines

***Abstract* – Using simplified tilt, flex and contact sensors, the goal is to develop a basic gesture recognition algorithm and then use it to register the alphabetical symbols of the ASL for the gesture to text translation. A Processing application will be developed for its counterpart, a text to gesture translator.**

Keywords – sign language translator; flex sensor; accelerometer; Arduino; Processing

I. Introduction

Speech-impaired and hearing-impaired people only have the sign language as their principal means of communication. However, this becomes problematic to people who are not familiar with the different gestures of the sign language, thus creating a communication barrier between the impaired and not. [1]

Sign language doesn't use acoustic sounds but visually transmitted sign patterns. By simultaneously combining hand shapes, orientation, movement, and facial expressions, this can be used to express the speaker's thoughts and it carries as much information as any spoken language. [2]

The availability of sign language translators based on the ASL has already been developed and/or researched by various scholars globally. Despite these, there hasn't been a solid and concrete process in which the translation is from text or speech to the ASL gestures thus suggesting a one-sided approach on the matter.

II. Methodology

A. Objectives

1) Enhance the dynamic aspect and generally improve of the 1st Generation Flex Sensor Gloves.

The research paper "User-Oriented Finger-Gesture Glove Controller with Hand Movement Visualization Using Flex Sensors and Digital Accelerometer" developed a control mechanism using registered states depending on how the flex sensors recognize a specific bending angle. The states are based on a set threshold, giving them a limited range. From 64 possible gestures using 5 flex sensors and the Y-axis of a three-axis accelerometer, the current set-up bumped it to 256 states, still not adding the additional contact sensor states.

Table I. Binary Gesture States

LETTER	THUMB	INDEX	MIDDLE	RING	PINKY	X	Y	Z
	128	*64*	*32*	*16*	*8*	*4*	*2*	*1*
A	0	1	1	1	1	0	0	0
B	1	0	0	0	0	0	0	0
C	1	1	1	1	1	1	0	1
D	1	0	1	1	1	0	0	1
E	1	1	1	1	1	0	0	0
F	1	1	0	0	0	0	0	0
G	1	0	1	1	1	1	0	0
H	1	0	0	1	1	1	0	0
I	1	1	1	1	0	0	0	0
J	1	1	1	1	0	1	0	1
K	0	0	0	1	1	0	0	0
L	0	0	1	1	1	0	0	0
M	1	1	1	1	1	0	0	0
N	1	1	1	1	1	0	0	0
O	1	1	1	1	1	1	0	1
P	0	0	0	1	1	1	1	1
Q	0	0	1	1	1	1	1	1
R	1	0	0	1	1	0	0	0
S	1	1	1	1	1	0	0	0
T	0	1	1	1	1	0	0	0
U	1	0	0	1	1	0	0	0
V	1	0	0	1	1	0	0	0
W	1	0	0	0	1	0	0	0
X	1	1	1	1	1	0	0	0
Y	0	1	1	1	0	0	0	0
Z	1	0	1	1	1	0	0	0

The following graph represents the bending pressure for the flex sensors only. This was solely used to filter initial letter states and to know where to attach contact sensors.

978-1-5090-2597-8/16/$31.00

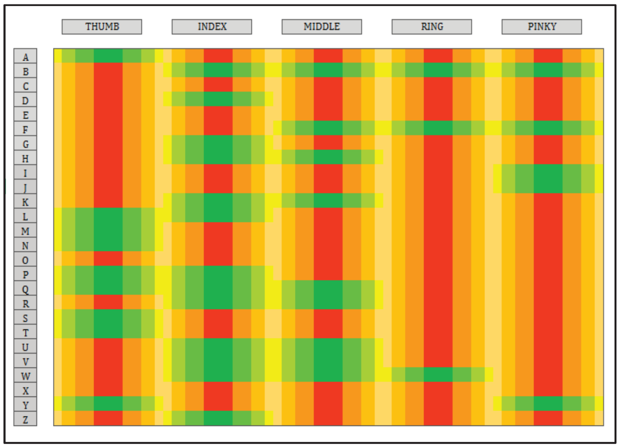

Fig. 1. Pressure Graph for the Flex Sensors

2) Work on existing ideas on sign language translators and develop a two-way translation process.

The availability of translators based on ASL and the likes of it is probably one of the biggest challenges of this project. However, despite the growing innovations on the subject matter, the translating parameters are one-sided, i.e., text to gestures alone. The relevant development for this project is that it aims to cross boundaries with a two-way translation method.

3) Develop a Processing software to personalize a mobile experience for the translator.

Encounters between hearing impaired people and non-hearing impaired people can happen anytime anywhere. The development of a Processing graphic user interface (GUI) is a stepping stone in developing a mobile application for smartphones and the like in the future. The program will allow the user to choose which action to take – whether to translate from sign to text or from text to sign.

4) Maintain the cost of the project to its minimum.

Based on the previous Flex Sensor Gloves thesis, the cost will be maintained to its minimum for more accessibility. The additional contact sensors were made from scrap copper pieces that were attached to wire jacks. Finally, the base of the glove system was maintained using locally available materials that are cheap and accessible.

B. Procedure

1) Materials

a) Arduino UNO Board

A single Arduino UNO board sufficed to make the translator system work. Five analog pins were used for the flex sensors while the accelerometer and the contact sensors had 4 and 6 digital pins respectively.

b) Flex Sensors

Flex sensors are analog resistors that work as variable analog voltage dividers. When the substrate is bent, a resistance output relative to the bent radius is produced by the sensor as shown in the Figure 2.1 The greater the degree to which the flex sensor is bent, the higher the electrical resistance it produces. [1, 9]

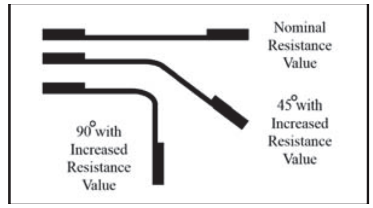

Fig. 2.1. Flex Sensors Offers Variable Resistance Readings

Fig. 2.2. 56cm Spectra Symbol Flex Sensors

For this system, 56cm Spectra Symbol short flex sensors were used.

c) 3-Axis Accelerometer

The accelerometer records the tilting values for the three axes to be inputted as binary states to filter out similar letters like "C" and "E," "A" and "M," and "U" and "V."

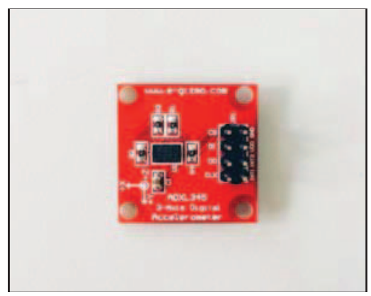

Fig. 2.3. E-Gizmo 3-Axis ADXL345 Accelerometer

d) Contact Sensors

Makeshift copper rings were soldered into wires and then plugged into the digital pins to act as switches. Each pin reads the constant +5V from the Arduino and then turns it off when two contact sensors touch.

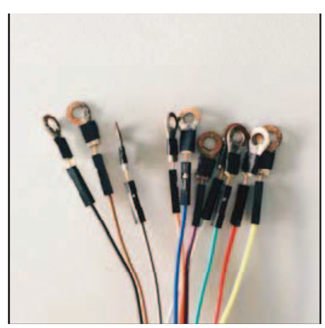

Fig. 2.4. Makeshift Contact Sensors

2) Circuit Design and Glove Assembly

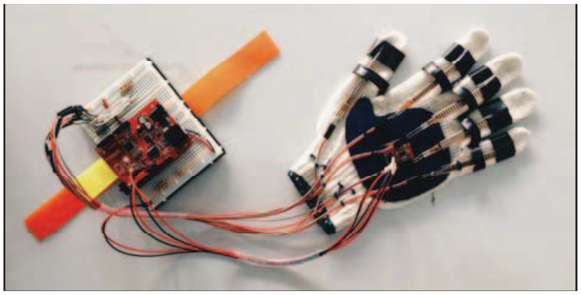

Fig. 3. Glove Circuitry

Each flex sensor was attached using a makeshift plastic casing that was sewn in place to the glove. A black rubber-plastic ring made out of electrical tape was also assembled per finger to hold the sensors in place.

The following circuit design has been observed for all five flex sensors using the analog pins A0 to A4 from the thumb to the pinky respectively. A 20 kΩ resistor per flex sensor was used to control the voltage divider that will input the threshold voltage to be used in determining the flex value in the Arduino program.

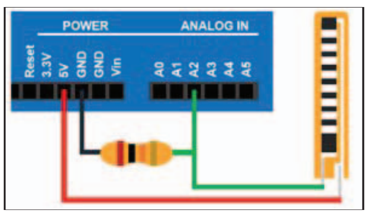

Fig. 4. Flex Sensor Arduino Set-up

For the accelerometer, the digital pins of the Arduino were connected to the CS, D1, D0 and CLK pins of the ADXL 245 based on the specification sheet to fully access the SPI reading capabilities of the accelerometer as shown in the figure below.

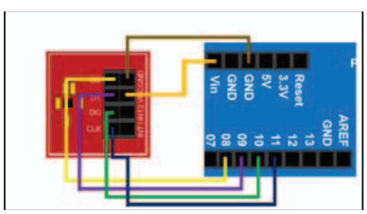

Fig. 5. Contact Sensor Arduino Set-up

The flex sensors provided unique letter states for less than half of the 26 alphabet characters. The five bending states of the fingers made conflicts with letters such as "C" and "E," "A" and "M," and "U" and "V."

Adding the accelerometer using the three axes filtered 18 out of 26 alphabet characters.

To filter the remaining eight characters that are still in conflict with either similar bending angles and/or similar axis position, contact sensors that act as switches were used to trigger and filter more the registration process

The following circuit design was observed for the contact sensors.

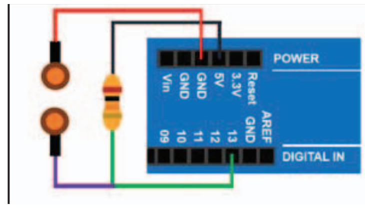

Fig. 6. Contact Sensor Arduino Set-up

III. Results and Data Analysis

A. Initial Problems

The flex sensors, albeit their effect, were not the entire working muscle for the translation system. The five bending states were only able to produce 5 unique ASL characters. The accelerometer on the other hand was able to filter out just 13 more, still behind the 26 over all required character count.

In order to finalize the filtering process, contact sensors were considered. Since the remaining characters, although similar in bending angles, have different touch points — like that of the "U" and "V" — so a sensor acting as a switch was introduced.

Ten contact sensors were used for the entire glove system and the assignments are shown in the following two figures.

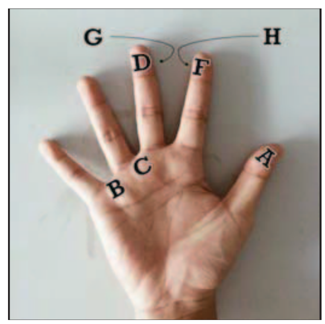

Fig. 7.1. Contact Sensor Arduino Set-up

Fig. 7.2. Contact Sensor Arduino Set-up

The sensor work when a pair of a ground contact sensor and a digital contact sensor touch each other. The assignment of the pairs for the remaining 8 characters of the alphabet is shown in the table below.

Table II. Contact Sensor Pin Assignment

	Ground Contact	Digital Contact	Digital Pin
Letter M	A	B	13
Letter N	A	C	12

Letter O	A	D	7
Letter R	J	D	7
Letter S	A	E	6
Letter T	A	F	5
Letter U	G	H	4
Letter V	K	D	7

B. Stability and Output

All the letters were registered. To test the stability, a separate Arduino code based on the first one was used. The only change in the code is that it outputs the value and not the corresponding bit value to check both the threshold and the frequency.

The letters were categorized into four based on the letters' dependence on specific sensors.

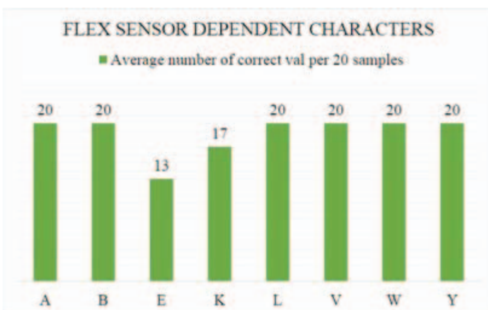

Fig. 8.1. Flex Sensor Dependent Character Frequency Graph

Fig. 8.1 shows the frequency in the Flex only category. This includes letters A, B, E, K, L, V, W, and Y. Most of the letters under this category got a perfect stability score with 20 out of 20 samples matching the expected "val". The average frequency is 18.75 per 20 samples.

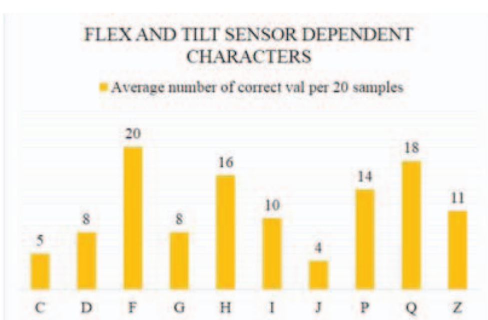

Fig. 8.2. Flex and Tilt Sensor Dependent Character Frequency Graph

Adding the tilt sensors filtered ten more letters: C, D, F, G, H, I, J, P, Q, and Z. Compared to the flex only category, the addition of the tilt sensors were not as successful as the first one. The accelerometer, based on a set threshold, was unstable when shifting from a non-tilted letter to another.

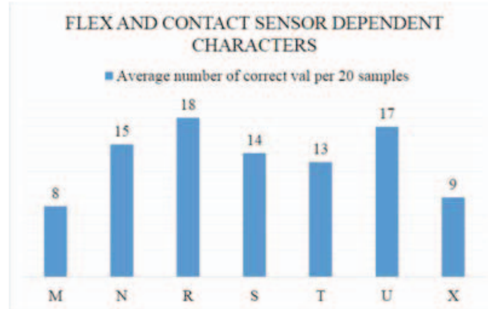

Fig. 8.3. Flex and Contact Sensor Dependent Character Frequency Graph

The third graph shows the result for flex and contact sensor dependent letters: M, N, R, S, T, U, X to filter out seven more characters. There was a stability issue as well but not as much as the flex and tilt only category. 11.40 per 20 samples was the frequency for the flex and tilt only category while 13.43 per 20 samples for the flex and contact only category.

There is only one character that uses all three of the modules: the letter O. Its frequency per 20 samples is 9; a little below the average for both tilt and contact sensor categories.

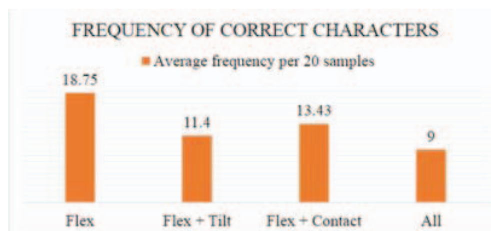

Fig. 8.4. Average Frequency Graph

As early as the first iteration on the previous study, stability comes down to the physical properties of all the sensors. Consistency, although not much of an issue, also reflects as a benchmark for the stability. Even with the precautions such as the rubber rings, plastic casings, and threaded holders for all sensors, the range is limited to turning one sensor on or off.

The above graphs are based on transition. To record per letter for the 20 samples, it starts from rest and then to the gesture representation of a specific character. It takes a little 3 to 4 samples to obtain the final gesture for a character. Some of the letters, especially the tilt dependent ones, take longer to get to the final "val".

C. Processing GUI

The translation system was then imported to a Processing application with fully-accessible GUI. The Processing application uses the "val" read-write communication link with the Arduino code. It converts the initial "val" to a corresponding photo for the gesture to text translation and as for the text to gesture, it collects the input from the keyboard.

The following screen grabs were from the functional Processing GUI.

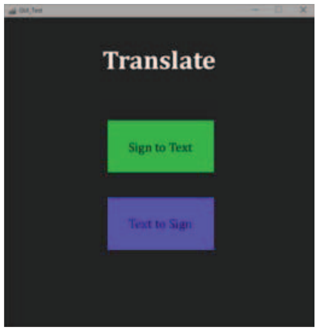

Fig. 9.1. Processing Home Page

Fig. 9.2. Sign to Text Page

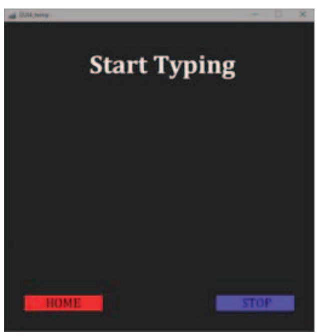

Fig. 9.3.1. Text to Sign Page

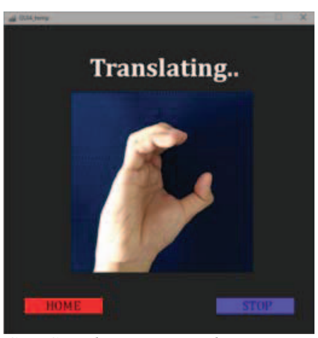

Fig. 9.3.2. Text to Sign Sample Letter Translation

The sign to text function of the Processing GUI outputs the corresponding letter using the text box in the Processing GUI. It recognizes all the 26 characters based on the value triggered by all active sensors.

As for the text to sign, a loop works in the background to display the prepared gesture photos

IV. Conclusion and Recommendations

Using simplified modules for the sensors, the project was able to meet the expectations of a Sign Language Translator. The project was able to maximize the range of the flex sensors while adding new kinds of sensor states for additional filtering.

The addition of the remaining two axes of the accelerometer plus ten additional contact sensors bumped the original 64 states to 256; more than double of the original and thus making the glove controller more dynamic.

Using the Processing software, the translation process was imported to be used as a mobile application. Although one recommendation is to develop an actual app based on mobile phones, the Processing GUI delivered most of the functions that were needed in the two-way translation process. Another recommendation is to use a 3D image processing software to visualize real time translation using blocks of hand shapes. Another proposal is to use Bluetooth shields for wireless communication. As of now, the communication works across Arduino and Processing through serial reading and writing. Since the focus of the project was the registration process of the letters, it is recommended for future researchers to work on the stability of all the active sensors present in the glove system. Letters should be more adaptive to the threshold of the sensors.

Lastly, maintaining the cost was achieved by using locally ready materials like knitted gloves and the makeshift copper rings for the contact sensors. Makeshift flex sensors, like the ones used in the previous research mentioned above, can be also used to perform the same bending angle data collection in the translation system.

References

[1] Haydar, Jamal, et al. "ASL Fingerspelling Translator Glove." International Journal of Computer Science Issues (2012): 254-260.

[2] Frenando, M. "Low Cost Approach for Real Time Sign Language Recognition." IEEE International Conference on Industrial and Information Systems. 2013. 637-642.

[3] Filipino Sign Language Online. 2011. 1 September 2014 <http://filipinosignlanguage.webs.com/>.

[4] Flores, Monique, and Siloy, Charles. "User-Oriented Finger-Gesture Glove Controller with Hand Movement Visualization Using Flex Sensors and Digital Accelerometer." Department of Electronics, Computer and Communications Engineering. Undergraduate Thesis. 2014.

[5] Raut, Ajinkya, et al. "Hand Sign Interpreter." The International Journal of Engineering and Sciences (2012): 19-25.

[6] Gujrati, Ambika, et al. "Hand-talk Gloves with Flex Sensor: A Review." International Journal of Engineering Science Invention (2013). 43-46.

[7] Syed, Abidhusain, et al. "Flex Sensor Based robotic Arm Controller using Microcontroller." Journal of Software Engineering and Applications (2012). 364-366.

[8] Analog Devices. "Digital Accelerometer Data Sheet ADXL345." n.d. Analog Devices. 2 September 2014 <http://www.analog.com/static/imported-files/data_sheets/ADXL345.pdf>.