Robotic Hands to Teach Sign Language

Raheem Maliki, Dalal Alhaidar, Khalid Attallah, Saleh Alsalem, Melissa Morris, Sabri Tosunoglu

Florida International University
Department of Mechanical Engineering
Miami, Florida 33174

rmali005@fiu.edu, dalhi001@fiu.edu, salsa029@fiu.edu, katta005@fiu.edu, mmorr009@fiu.edu, tosun@fiu.edu

ABSTRACT

This paper develops a robotic platform intended for the task of reproducing and teaching sign language. Specifically, realistic arm and finger motions were developed and then produced by the prototype constructed. The device presented can be used as an interactive way to teach sign language, one that is particularly engaging to children, or be a critical component of an automated system to translate spoken language into visual signs in real-time.

1. INTRODUCTION

Human beings are innately social and the need to communicate is essential in order to pass on ideas and information. This is done primarily by speech and listening, but unfortunately some people live with disabilities that hinder verbal communication. Individuals who are deaf cannot hear speech, while physical or mental impairments inhibit the production of speech for others. Therefore, sign language was developed as a means of non-verbal real-time communication.

This project looked into the concept of teaching children sign language with the aid of robotic hands. Generally, children have short attention spans and can usually become overstimulated during learning of a subject as broad as sign language.

Furthermore, we live in an age were robotics are becoming a regular part of our daily lives, hence using robotic hands to teach the children will not only improve their enthusiasm towards learning the subject, but will also reduce over-stimulation in classrooms.

American sign language (ASL), the primary form of sign language in the United States traces its origins back to the early 1800s. A Yale graduate by the name of Dr. Thomas Hopkins Gallaudet wished to communicate with his neighbor's deaf daughter. He traveled to France where signing had already been developed and met Laurent Clerc. Clerc later traveled back to the US with Gallaudet and started the first school for deaf in Hartford, Connecticut. In 1864 congress passed a legislation that created the National Deaf-Mute College, the first college of signing, to issue degrees. This university was later renamed Gallaudet University [1].

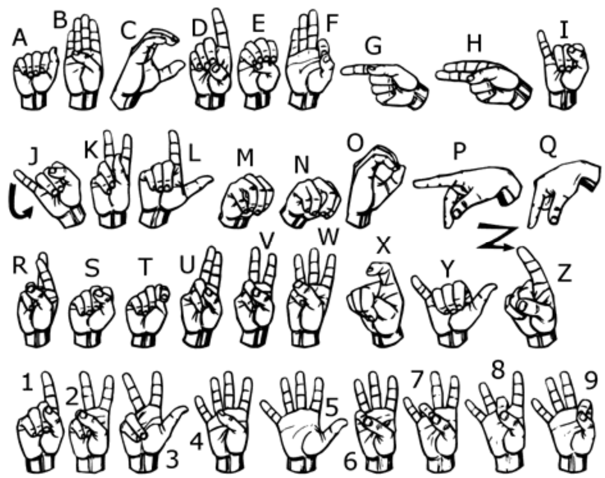

Figure 1. Sign Language Letters [2]

Much like sign language, robotics has a long history. The term robot was coined from "robota", which means servitude. There are different forms of robots available today. These include robots in animal form to full humanoid robots that complete tasks with or without instructions from people. Robots are now an everyday item seen in homes around the world. In general, their purpose is to make leading a life easier for people whether it is an iRobot [3] that aides cleaning the house to the more humanoid robots like the Softbank companion robot shown in Figure 2.

Figure 2. Softbank Robot Companion [4]

2. Conceptual design

The primary goal of this project was to find a cost-effective solution to the proposed topic of developing a robot to teach sign language. Since learning sign language is worldwide endeavor, an open source design for the robot was sought as one objective. In addition, the robot was minimally required to have have sufficient degrees of freedom and range of motion to enable the performance of sign language.

To help accomplish the project goals with the limited time and resources available during the duration of this project, currently available platforms were investigated. The InMoov robot, an open source design, was used as the base for this project because it met the primary objectives noted above. A complete InMoov robot is shown in Figure 3. Figure 4 shows a closer view of only arm and hand portion.

Figure 3. InMoov Robot [5]

The primary form of production for this project was 3D printing. ABS plastic was used as it was found to the stronger that its PLA counterpart. The arms of the InMoov robot are intended to be printed and mounted on a specially-designed platform for programming.

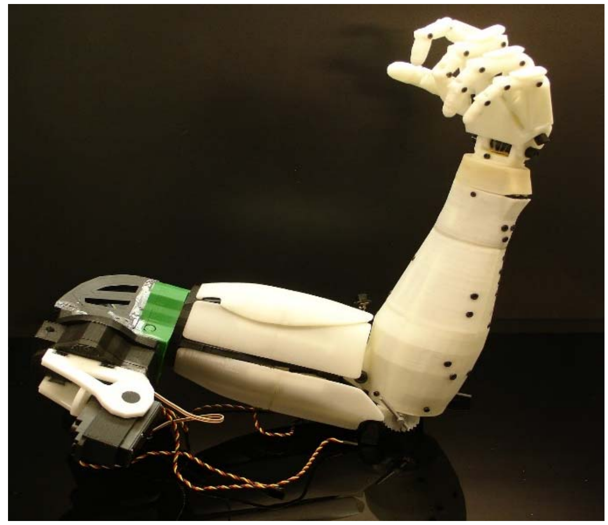

Figure 4. InMoov Robotic Arm [5]

2.1 Servo Motors

Three different specifications of servos were used in the robotic arm assemblies. The servos had to be carefully selected to perform the different angles of motion required for different parts of the arms and hands. There was especially differences in the torque requirements to move different parts of the robotic hand that required proper sizing and consideration of actuators. Table 1 summarizes the servos and specifications selected.

Table 1. Servos and Specifications

	HS 805BB	MG996R	HK15298B
Max Torque	24.7 Kg-cm	11 Kg-cm	18 Kg-cm
Speed	0.19 sec/60°	0.19 sec/60°	0.18 sec/60°
Mass	152.0 g	55.0 g	66.0 g

2.2 Assembly

My Robot Lab (MRL) was used in the initial assembling and programming of the robot. Several of the robot components had to be put to certain angles before assembling. The instructions for assembling can be found on the InMoov website. [6]

The Raspberry Pi was used to do the initial programming of the robotic arm. The MRL Graphical User Interface (GUI) was installed on the Raspberry Pi and ran using the sequence shown in Figure 5.

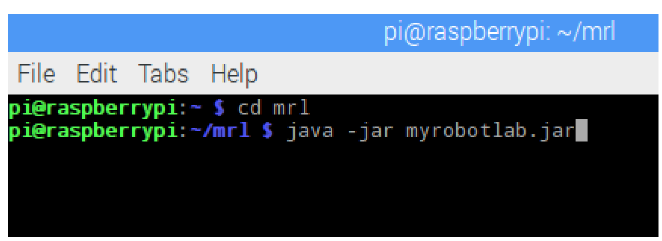

Figure 5. Sequence to Launch MRL

Due to the limited Pulse Width Modulation (PWM) pins available on the Raspberry Pi, an Adafruit PCA9685 with 16 channels was used to interface the Raspberry Pi with all the servos via i2c communication. This servo driver is shown in Figure 6.

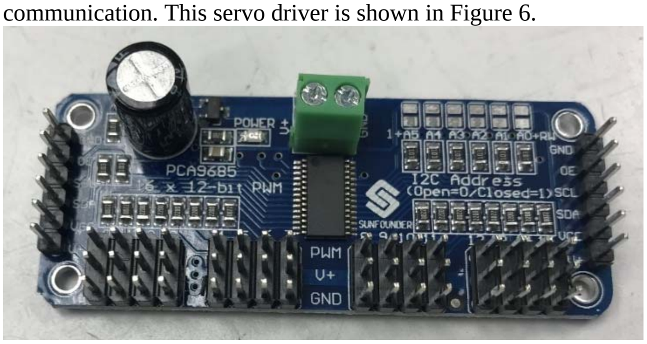

Figure 6. Adafruit PCA9685

Once the MRL is launched in the Raspberry Pi, a window such as the one in Figure 7 is displayed.

Figure 7. MRL GUI

The programming language used for the MRL GUI is Python. Once all the necessary services necessary for servo control are installed on the MRL GUI, the following code is entered in the python window to initiate the servo control GUI:

```
.adaFruit16c =
Runtime.createAndStart("AdaFruit16C","Adafru
it16CServoDriver")
2.raspi =
Runtime.createAndStart("RasPi","RasPi")
3.adaFruit16c.setController("RasPi","1","0x4
0")
4.lpinky = Runtime.createAndStart("LPinky",
"Servo")
5.lring= Runtime.createAndStart("LRing",
"Servo")
6.lmid = Runtime.createAndStart("LMid",
"Servo")
7.lindex = Runtime.createAndStart("LIndex",
"Servo")
8.lthumb = Runtime.createAndStart("LThumb",
"Servo")
9.lwrist = Runtime.createAndStart("LWrist",
"Servo")
10.llowerbicep =
Runtime.createAndStart("LLowerBicep",
"Servo")
11.lupperbicep =
Runtime.createAndStart("LUpperBicep",
"Servo")
12.lshoulderside =
Runtime.createAndStart("lshoulderside",
"Servo")
13.lshoulderforward =
Runtime.createAndStart("lshoulderforward",
"Servo")
14.lpinky.attach(adaFruit16c,0)
15.lring.attach(adaFruit16c,1)
16.lmid.attach(adaFruit16c,2)
17.lindex.attach(adaFruit16c,3)
18.lthumb.attach(adaFruit16c,4)
19.lwrist.attach(adaFruit16c,5)
20.llowerbicep.attach(adaFruit16c,6)
21.lupperbicep.attach(adaFruit16c,7)
22.lshoulderside.attach(adaFruit16c,9)
23.lshoulderforward.attach(adaFruit16c,8)
```

Each servo on the robotic hand is assigned a pin on the Adafruit PCA9685. Once the servos are plugged into the correct pins, they can be controlled by the GUI.

Once the code is run, a window such as that shown in Figure 8 appears. The servos can be activated by moving the sliders shown in the upper portion of the window.

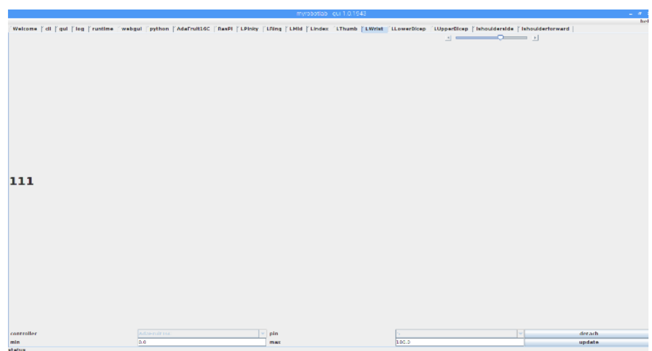

Figure 8. Servo Control GUI on MRL

As noted previously, three different servos were used to control different areas of the arms and hands. The HK15298B servos were used to drive the fingers, the MG996R servos were used to drive the wrist rotations, and the HS805BB servos were used to drive the shoulders and the biceps. Due to the limited angle of motion of the HS805B servos, they were modified to allow continuous rotation.

This was accomplished by opening the servos as shown in Figure 9 and cutting the stopper on the gears which limited their range of rotation. Furthermore, the feedback potentiometer in each servo was removed and soldered to an extension cable. Each potentiometer was attached to different joints of the robotic arm where required to obtain proper feedback and control of the servo positions.

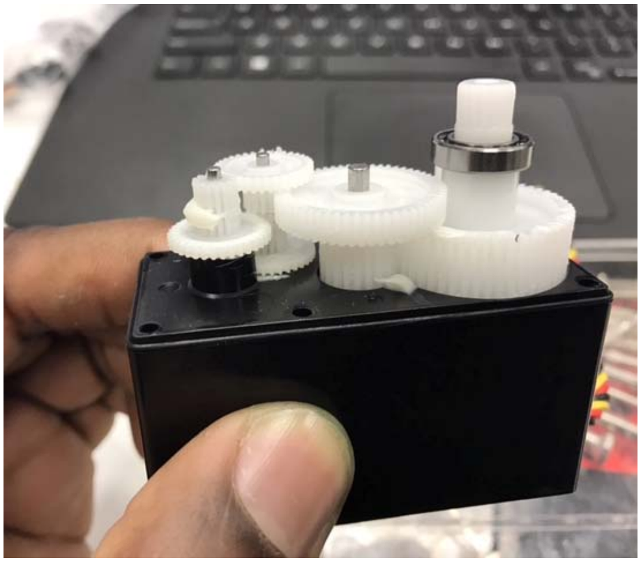

Figure 9. Inside the HS805BB Servo

The servos were implemented according to guidelines provided by the InMoov open-source documentation. However, modifications were required to accommodate the specific servos used as well as to accommodate other variances in printing, design and the materials used.

Figures 10 – 14 shown depict various parts as they were assembled into the final prototype. Specifically, finger servos, assembly of fingers, forearm attachments to the forearm and the wrist as well as the torso of the robot are displayed in these figures.

Figure 15 shows the completed left arm attached to a moveable base at a proper height proportional to the size of the arm.

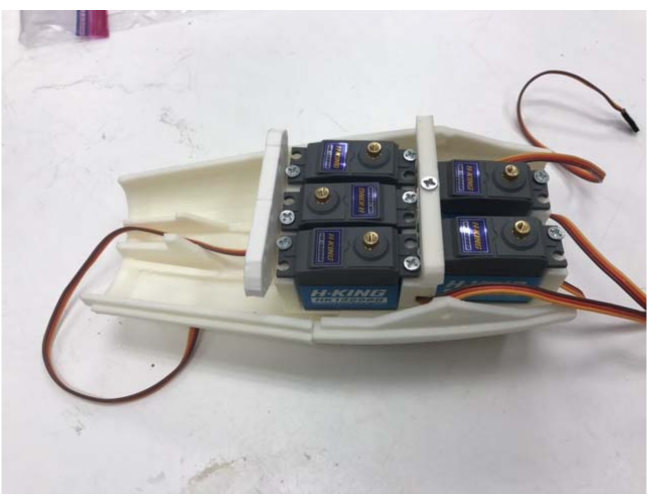

Figure 10. Finger Servos in the Forearm

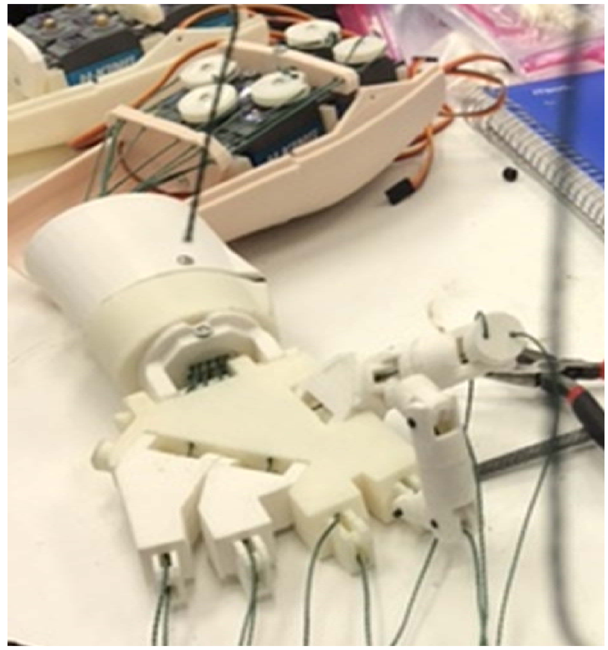

Figure 11. Assembly of Fingers unto the Wrist

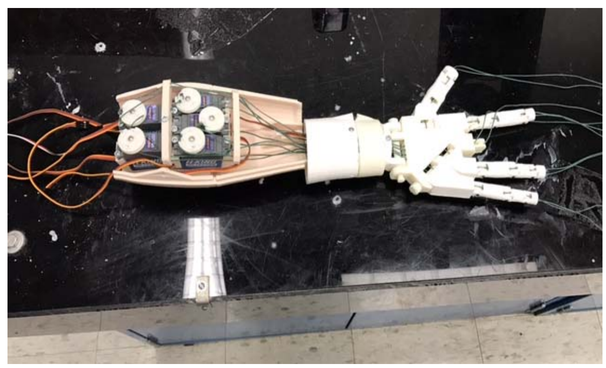

Figure 12. The Forearm Attached to the Wrist

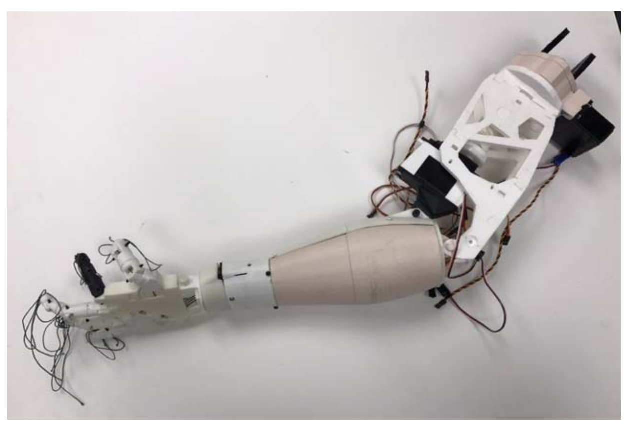

Figure 13. Forearm Attached to the Bicep

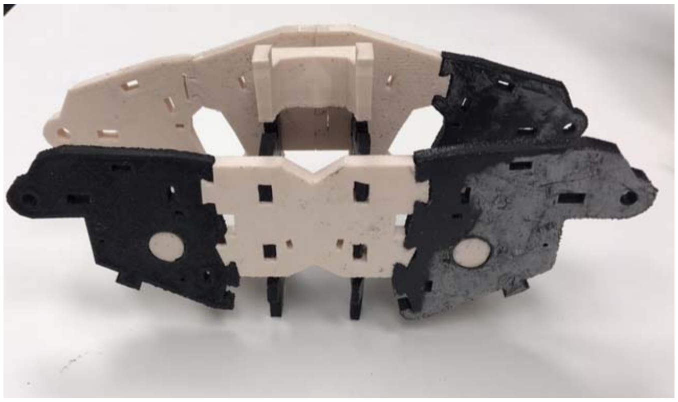

Figure 14. The Torso of the Robot

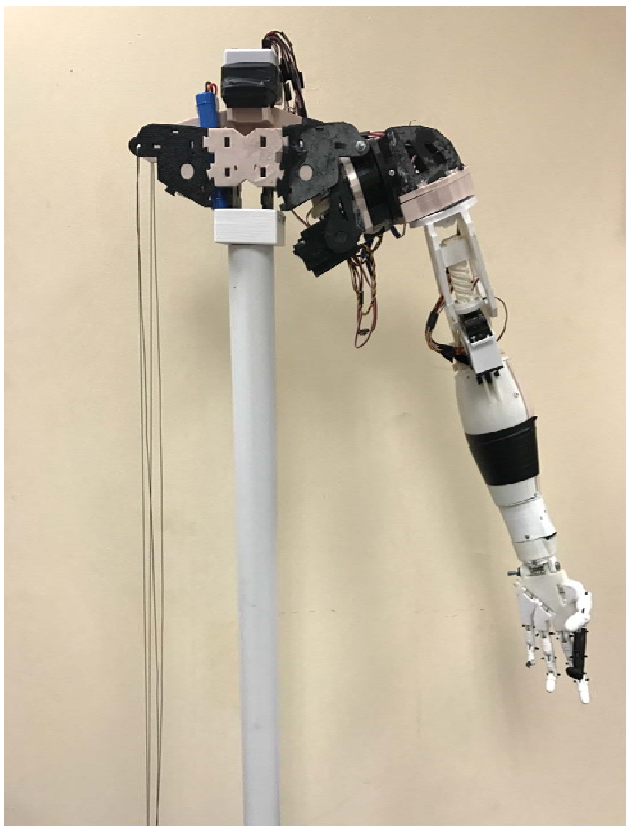

Figure 15. The Left Arm of the Robot

3. Programming

Though a Raspberry Pi with MRL was used in the preliminary stages of assembly and testing, a different controller was used in the final system to ease the amount of programming and speed up development and testing.

The EZ-B V4, shown in Figure 16, was used for the final programming of the robotic arm. It is a robotic servo controller that has 24 digital pins in addition to eight analog pins. It features built-in voice command options, speaker and Wi-Fi capabilities allow for integrated web-based remote control.

Figure 16. EZ-B V4 Robotic Controller

The EZ-B runs its own operating system and software that makes programming servos an easy task. The on-board Wi-Fi capabilities makes it possible to control the robot from different devices including PCs, laptops, tablets, and smartphones.

For this project, two specific software functions of the EZ-B are used, namely, the frame creating options and the action creating option. An example of the GUI for these tools is shown in Figures 17 and 18.

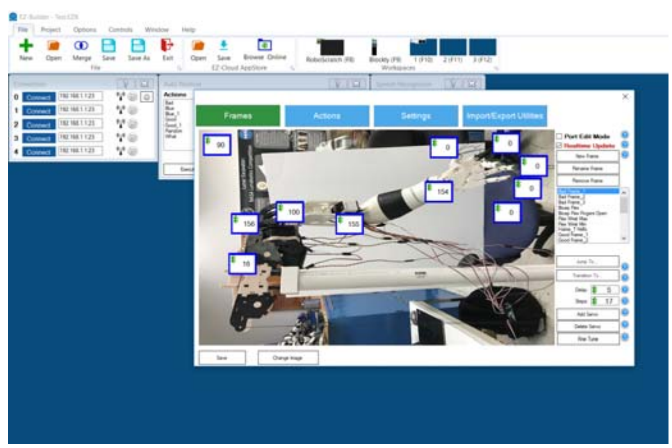

Figure 17. EZ-B Frame Creating Window

In the frame creating window of the EZ-B software, all the attached servos are given specific frames. This then allows intuitive actuation of each servo to obtain the desired joint angles.

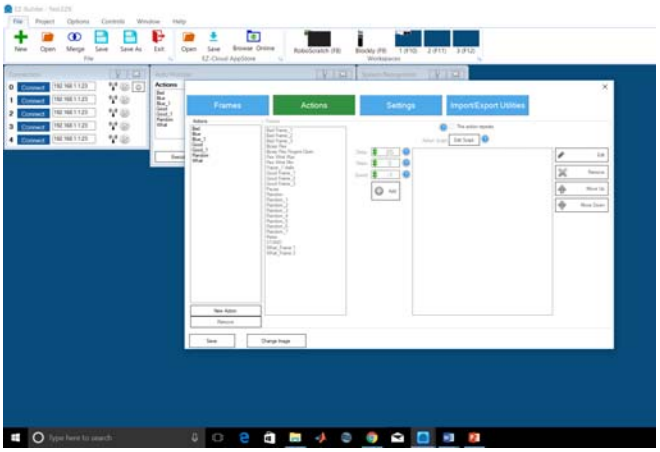

Figure 18. EZ-B action Creating Window

In the action creating window of the program, several frames can be put together to create specific gestures. This feature is especially helpful as sign language depends upon the use of gestures. This greatly reduced the amount of time spent coding.

4. Design Considerations

4.1 Health and Safety

Caution must be taken when operating the robotic arms as there are moving parts that form pinch points or collision hazards. To reduce the risk of harming someone, adequate space should be given for the workspace of the robotic arms during operation. Children should be properly supervised when learning sign language from the robotic arms and be kept away at a safe distance. Also, users must make sure that the wheels are properly secured. The wheels of the current prototype do not have brakes and the robot might move slightly while operating, but brakes can be installed and utilized in future versions. The structural components of the robotic arms were printed and assembled in accordance to the regulations for health and safety to reduce problems with this aspect.

4.2 Manufacturability

The 3-D parts of the robotic arms used are found on the InMoov project website as mentioned previously. The parts were printed on a 3-D printer. The choice of material used in the printing is something that must be considered in the manufacturability. The major difference during assembling with either PLA or ABS plastic is the method in which the plastic parts and joined together. Acetone, if used properly, can be used to weld ABS together. Acetone melts the ABS plastic and gives the impression of the parts haven been printed as one solid piece. If, however, PLA is used, a strong glue is to be used to bind the parts together. ABS plastic was used in the printing as it had a higher strength than its PLA counterpart and due to the ability to combine parts together to form larger components.

A PVC pipe was used as a stand for the robot in conjunction with the mounting platform that was designed for the robotic arms. The other end of the PVC pipe was attached to steel wheel base to ensure ease of transporting the robotic arm while being a sturdy foundation.

5. Design Experience

5.1 Overview

The objective of the project was to design humanoid robotic arms that could teach sign language to children. The design experience included contemporary issues, standards used, the impact of design in global markets, ethical responsibilities, and life-long learning.

5.2 Contemporary Issues

Several issues were encountered when assembling the robot. The primary issue encountered was the accuracy of the parts generated by the 3-D printer that was used. The accuracy of the printer varied from print to print which resulted in difficulties mating the parts. This is evident in the Figures 19 and 20 below.

Figure 19. Forearm Assembly Misalignment

The misaligned parts had to be reprinted several times until suitable alignment was achieved. In addition, several of the parts had to be modified to fit the required design criteria.

One of these areas was the wrist. The wrist was modified due to the change in size of the servo utilized from that which was suggested by the InMoov project. The space provided had to be modified to allow for a bigger servo to fit.

Figure 20. Forearm Assembly Misalignment

6. Future Work

The platform used as the basis of this project was developed from an open source and is available for download. The code generated for making the arms perform sign language were made available to the public. Users man print the parts for the robotic arm, purchase and install the other components, and install the code developed to gain benefit of the platform presented in this paper. It is expected that the robot will encounter some marketing and implementation issues since it is a new idea and can benefit from further programming and development. However, it is expected that eventually people around the globe will get used to the idea of learning sign language by using humanoid robotic arms.

Several improvements can be made in the future to further improve this project. The focus of this phase of the project was assembly of the left arm and hand and making sure they moved properly. In the future, some of the parts can be modified to improve signing. For example, the wrist only rotates in the presented implementation and it does not bend back and forth. Adding another servo for the wrist will greatly improve the degrees of freedom of the arms, which in turn will make signing more accurate and versatile.

7. Conclusions

Children with certain disabilities and conditions face communication obstacles. Studies have noted that children can be overstimulated when taught sign language by traditional means. This can be detrimental to their ability to learn sign language. The robot presented in this work was designed and programmed to teach children sign language while overcoming the common obstacle of overstimulation.

The robot' signs need to be correct and precise to avoid teaching a wrong sign or incorrect gestures. The InMoov robot has never been used in such a way before nor has it been fashioned towards such a goal. As such, improvements and future work are needed to make the robot even more versatile.

8. Acknowledgments

The authors extend their thanks to the Robotics and Automation Laboratory, Department of Mechanical and Materials Engineering, and Florida International University for providing the opportunity to work on this extraordinary project. The unlimited access provided to the 3D printer in the Robotics and Automation Laboratory made it possible for the authors to build the prototype literally after printing hundreds of parts.

9. References

[1] History of American Sign Language - Start ASL. Retrieved October 11, 2016, from https://www.start-american-sign-language.com/history-of-american-sign-language_html

[2] Sign Language Letters. (2016, March 1). Retrieved October 16, 2016, from http://www.mnhuntingcalendar.com/sign-language-letters/

[3] Roomba Robot Vacuum | iRobot. (n.d.). Retrieved October 16, 2016, from http://www.irobot.com/For-the-Home/Vacuuming/Roomba.aspx

[4] Do you remember Pepper? The Robot in Japan with Emotions. Newdawn. (2016). Retrieved October 17, 2016, from http://newdawnblog.com/2016/08/11/do-you-remember-pepper/

[5] Build yours. (n.d.). Retrieved October 16, 2016, from http://inmoov.fr/build-yours/

[6] InMoov Project, https://www.inmoov.fr/.