HUMAN ROBOT INTERACTION WITH CLOUD ASSISTED VOICE CONTROL AND VISION SYSTEM

by

RAVI PATEL

Presented to the Faculty of the Graduate School of
The University of Texas at Arlington in Partial Fulfillment
of the Requirements
for the Degree of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

THE UNIVERSITY OF TEXAS AT ARLINGTON
May 2018

Acknowledgments

I would first like to thank my research advisor Dr. Panos Shiakolas for all his guidance, support and motivation throughout my research work. I could not have imagined having a better advisor and mentor for my masters thesis research. I would also like to thank my thesis committee member Dr. Subbarao and Dr. Kumar for their valuable time and inputs on my research.

I would also like to thank to my lab colleagues at Micro Manufacturing Medical Automation and Robotic Systems (MARS) laboratory including Christopher Abrego, Dr. Prashanth Ravi, Tushar Saini, Kashish Dhal, Samson Adejokun, Sudip Hazra, Parimal Patel, Henry Nguyen and Abdul Hafiz.

I would like to give special thank to staff member of MAE department for cooperating with my research work and helping me in my speech recognition evaluation as a test subject.

I would like to make a special thanks to my family: my parents Kirti Patel and Bhanu Patel and brother Yash Patel for holding down the line while I am doing my graduation and believing in me.

April 12, 2018

Abstract

HUMAN ROBOT INTERACTION WITH CLOUD ASSISTED VOICE CONTROL AND VISION SYSTEM

Ravi Patel, MS

The University of Texas at Arlington, 2018

Supervising Professor: Panos S. Shiakolas

The objective of this research is to investigate a way of interaction between humans and robots, which is through voice or speech commands. A Biomimetic Artificial Hand (BAH) is used as a platform to perform grasping tasks using human voice as an interacting and instructing medium between humans and robots. It is a hands-free approach of issuing commands to the BAH since it does not require the user to wear any specialized equipment. Previous research has shown difficulties in recognizing more than one word, database management for stored voice, and requirement of sufficient computing power. National Instruments software LabVIEW and hardware myRIO are used as the interface between the user and BAH. The concept of using cloud application services is applied, which is based on using the speech recognition Application Program Interface (API) by Microsoft which accepts a verbal command, transfers to the cloud for further processing and returns the command in string (text) form. This approach reduces the use of local computing power requirements and yields fast and accurate speech recognition (SR). A vision system is also incorporated as a a safety feature to verify the presence of the correct object in the workspace. The string results returned from the API is further locally processed to identify the action to perform, object, object identifiers (number, color, size) and grasping pattern of object from the existing database. Voice command evaluation performed on the hardware platform with a biomimetic artificial hand indicates that the proposed interaction modality could be advantageously employed for successfully instructing or interacting with a robotic device.

TABLE OF CONTENTS

Appendix

LIST OF FIGURES

LIST OF TABLES

Nomenclature

API	Application Programming Interface
BAH	Biomimetic Artificial Hand
CAS	Cloud Application Service
HRI	Human Robot Interaction
JSON	Java Script Object Notation
MCV	Microsoft Computer Vision
SR	Speech Recognition

CHAPTER 1

Introduction

Voice control of a machine using natural human voice has been a research goal for decades given that verbal communication is the most natural way to communicate amongst humans. This advantage of natural communication among humans has led to a desire to have this same form of communication between robots and humans. A robotic system can be controlled using various different modes of interaction such as glove control [8], remote control using graphic user interface and voice control robot by issuing voice commands [9] [10]. Various studies have been conducted in the field of robotics, in order to control a robot with voice commands [5] [6] [7] (which is the focus of this research). Voice commands allows the user to control the robotic system without being physically present in the robot workspace. Although the voice control approach presented in this research could be applied to other research platforms and applications, a biomimetic artificial hand (BAH), developed at the MicroManufacturing Medical Automation and Robotics Systems (MARS) Lab at the University of Texas at Arlington is used as an experimental platform to evaluate the performance of the research performed for voice control. Moreover, a vision system is also incorporated as a feature to verify appropriate grasping of objects considered for this research.

It is a well known fact that disabled people face different challenges and difficulties regarding their physical impairments. The BAH can potentially provide assistance to people with disabilities by improving their capabilities of movement [11]. On this platform, different ways of interacting with the BAH are investigated such as glove control and predicting grasping using machine learning, manual control of the BAH fingers, and voice control of the BAH. There are also some efforts of Human Robot Interaction (HRI) by monitoring muscle movement and brainwaves using devices such as MindWave in order to actuate an artificial hand and grasp objects [12]. Furthermore, the voice control modality can also be used in a hazardous environment, which will eliminate the risk of human injury or casualty. Despite having an affordable biomimetic artificial hand, it is important to have an easy way to “interact” with it and accomplish tasks such as grasping objects.

1.1 Human Robot Interaction

Human-Robot Interaction (HRI) is a field of study dedicated to understanding, designing, and evaluating robotic systems for use by or with humans [13]. HRI makes it possible to operate in an alien environment where humans cannot access physically. Sensors on the robot can then replicate human senses and provide data for further investigations. Humans possess the ability to understand the environment consisting of semantic entities and higher order relationships which include knowledge about previous interactions and events [14]. Meanwhile, robots can estimate a metric picture of the world from sensor information and then determine which sequence of actions to take [14]. In order to send/receive data and commands, different methods and approaches are used to interact with a robot.

Remote controlling a robot is still the most common way of interacting with a robot, but it limits the user commands and range of operation. Gesture control is also a modality in development [15], such as the Lego NXT robot controlled using a hand gesture. In a MindWave control module, the system reads and parses the data coming from MindWave to a valid data values for further programming [10]. Research at the University of Tokyo, developed a new master hand to control a slave robot for telexistence, where the user wears a master hand to control the slave robot [16]. The National Aeronautics and Space Administrations (NASA) and Defense Advanced Research Projects Agency (DARPA) has investigated telemanipulation and developed a robotic hand that is controlled by a user wearing a glove, in order to work in a hazardous environment of low earth orbit and planetary exploration [17]. Voice control modality for robot can also be applicable in reinforced learning of robot behavior as feedback [18]. Voice assisted robots can also assist in surgery, for instance when a surgeon needs help with equipment movement. If, during the surgery, the surgeon finds that the orientation of a camera or telescope is not appropriate, then instead of using their hands to correct the orientation, they could issue a voice command such as “move right”, “move left”, etc. to set the orientation of the camera [1].

1.1.1 Biomimetic Artificial Hand and NI LabVIEW

For this research, a BAH was used as a platform for evaluating voice commands. A 3D model of the right hand was downloaded from the InMoov project, developed by Gael Lengavin and manufactured and assembled in house [19]. The final working assembly of the BAH is presented in Figure 1.1. Each finger consists of three joints, whereas the thumb consists two joints [19] [20]. All the fingers are underactuated and tendon driven with the help of five servo motors (one servo motor for each finger). The BAH has a span of 195mm (from outstretched pinky to outstretched thumb) and a palm height of 212mm (from wrist to tip of middle finger) [21].

National Instruments LabVIEW is used as the programming language for the platform. LabVIEW is a graphical programming language that allows the user to have a pictorial view of virtual mechanical and electrical controls and indicators. Furthermore, the software allows the user to perform mathematical computations in real-time while interacting with control hardware. It is helpful in applications that

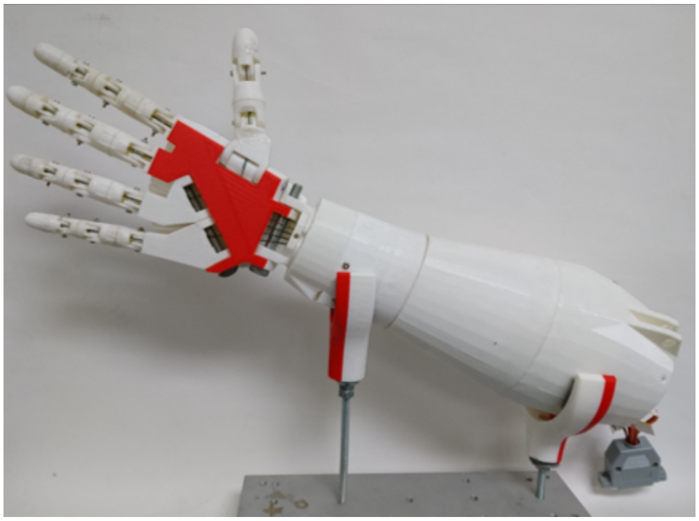


Figure 1.1. Assembled InMoov prosthetic hand at MARS Lab.

require testing, measuring, and controlling with rapid access to hardware and data [22]. Among one of the hardware platforms provided by National Instruments is the myRIO, which is a re-configurable Input/Output micro controller [23]. The myRIO has an onboard LinuxRT operating system with high clock speed and has relatively more I/O and GPIO pins than the Raspberry Pi and Arduino. It includes analog input, analog output, digital input, digital output lines, audio input/output and wireless connectivity to a computer. The programming and processing is performed on the host computer (with LabVIEW installed) and then the processed parameters are transferred to the myRIO. Figure 1.2 represents the relation between hardware and software setup developed as a part of the previous research on BAH.

1.2 Speech Recognition System

Digital processing of a speech signal algorithm is one of the widely used speech recognition (SR) approaches. Speech signal identification consists of the process of converting a speech waveform into a text format that is useful for further processing.

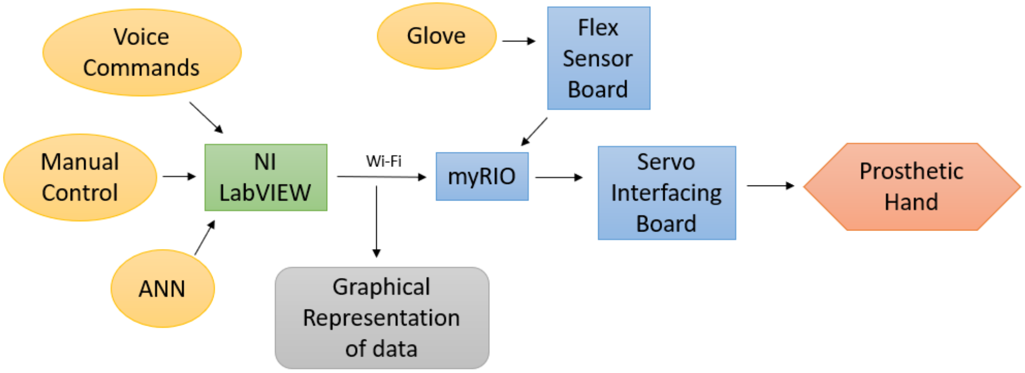


Figure 1.2. Relationship between current hardware and software with all the control modalities.

SR is done after capturing voice inputs from the user through a microphone. An approach to convert speech to text (string) was investigated in order have a useful format for programming and algorithm. Textual format allows the developed system to program different desired voice commands in text. A study of different SR systems was performed before arriving at a final usable approach of cloud service application. Figure 1.3 presents the concept of converting speech to text on a cloud server. The textual format returned from the server is analyzed locally to execute desired actions based on the voice commands.

Voice Command
Speech Analysis
Text Format

Figure 1.3. Concept of Speech to Text conversion.

Considering the applications of the BAH in real world applications, the goal was to develop a system with the least amount of hardware and software complications, to prevent tedious debugging in the future in case of any system error. The current system consists of a hardware chassis motor control and data collection from the embedded sensors. All modalities that control the BAH utilize some processing power of the real-time target system (NI myRIO). Moreover, other interacting modalities are currently being investigated to include into the system. Therefore the developed environment in LabVIEW uses less hardware and software resources provided by the RT target.

Different methods proposed by other researchers for SR were studied and considered for developing a voice control environment in order to provide useful conversion of the voice commands [1] [2] [5] [24] [25]. Along with the objective of developing a voice control environment in LabVIEW for BAH, it was desired to have a system able to recognize a good range of voice commands and also recognize full sentences, in addition to having a flexible and expandable platform.

1.2.1 Literature Review

Research published in The American Journal of Surgery, voice activated surgical robotics model voice commands in different statistical methods, which is modeling a single unit of spoken word and modeling a word as small sequence of sounds called phones [1]. The modeling and storing of data obtained from the audio signals requires a large amount of computer memory. This voice data varies with every person's language background, voice levels, and speech accent [2]. Two similar voice commands, but in different languages, need to be modeled differently in order to be recognized. The advantage of modeling phones is that the total number of phones are relatively less compared to model of single unit word. Also, the number of phones remain constant regardless of the words required to model. During SR, the input audio signal is converted into frames and each frame is then compared to phone models, and a match score for each comparison is saved in memory. The match score for each word is calculated by combining the scores from the phones that comprise the word [1]. Figure 1.4 represents the flow diagram for speech recognition using the above procedure.

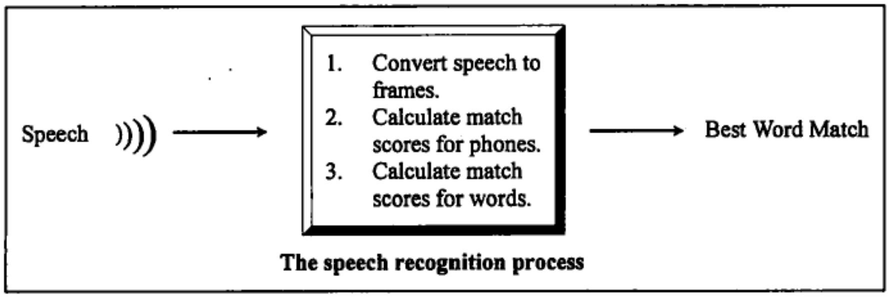

Speech
1. Convert speech to frames.
2. Calculate match scores for phones.
3. Calculate match scores for words.
Best Word Match
The speech recognition process

Figure 1.4. Speech Recognition Process using methodology described in [1].

One can take advantage of this system of modeling words and phones, if there are a limited number of voice commands to be recognized. The application of this SR system in the development of desired voice control environment, requires a large database of modeled voice commands and phones to be developed, which will increase if one wants to support different languages [1]. Also, comparing the input voice commands with each single modeled word in the database requires more computing power from the local hardware host, which will deviate from the objective of a system that uses less computing power of the RT target.

Another approach investigated for developing a SR system was using the concept of Mel-Frequency Cepstral Coefficient (MFCC) and Dynamic Time Wrapping (DTW) for producing voice features and comparing them with the voice features of an input audio signal [2]. MFCC consists of seven steps which are Pre-emphasis, Framing, Hamming Windowing, Fast Fourier Transform, Mel Filter Bank Processing, Discrete Cosine Transform, Delta Energy and Delta Spectrum as presented in Figure 1.5. Every step monitors voice features of the input audio signal at different frequencies. These voice features are then saved to a database as a coefficient matrix.

The coefficient matrix of input audio signal is then compared with all the matrices stored in the database in order to recognize speech. DTW is used to compare the coefficient matrices. The DTW algorithm is based on dynamic programming and measures the similarity between two time series. The DTW compares two dynamic patterns and measures their similarity by monitoring the standard deviation between them [2]. Figure A.2 represents a flow process of the algorithm.

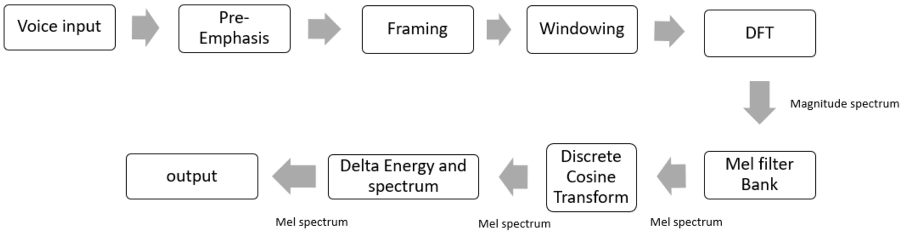


Figure 1.5. Process flow of MFCC algorithm for getting coefficient matrix for an audio signal [2].

Research from Hebei University of Technology in Tianjin China, a speed control robot system was presented using MFCC and DTW for SR. SR was performed using MFCC and DTW techniques where, voice features were extracted using MFCC algorithm and a neural network model is trained based on voice features extracted. The issued voice command is then compared to an existing patterns in a database using DTW technique [27]. The SR accuracy test was performed in both normal and noisy environments. According to their test results, there was 100% accuracy in normal environments, whereas the noisy environment did affected the SR accuracy distinctly [27].

In both approaches discussed above [1] [2] requires the creation of a database of modeled words for recognition, and manage that database as it expands with more words. Once all of the data for SR of desired words are modeled, a good amount of processing power on the local machine is required for search for matching of the model, phones or coefficient matrices in the database. As the size of database increases, more memory power is required or else the system will slow down. A large amount of data is required to create an artificially intelligent system that can perform SR for any voice command by any person with different speech accents in different languages.

1.3 Cloud Application Services

One of the possible solutions to the difficulties encountered in using the methodologies discussed in section 1.2.1 could be alleviated using cloud applications [28]. Rapidly expanding internet resources and wireless networking have the potential to liberate robots and automation systems from being limited to on-board computation, memory and software limitation [4]. CAS allow researchers to take advantage of network applications that provide high level processing with minimal management effort [29]. This allows researchers to leverage high processing power of services and received the already performed analysis. An example of a cloud application is the online word processing capabilities offered by Google Docs. One can easily access a Microsoft Word document from anywhere in the world without having the Microsoft Office package installed on a local machine [28]. This is helpful because one does not have to worry about installing software packages to run documents on their local machine [30].

Considering as well the scenario of a self driving car which requires an inordinate amount of sensors in order to sense surrounding environment and then analyze data for decision making [28]. The Google self driving car exemplifies this idea. It indexes the maps and images collected and updated by the satellite, street view, and crowd sourcing from the cloud to facilitate accurate localization [28]. All these network applications allows the user to perform parallel computing on a machine and on the network at same time for faster processing of data. This approach of using cloud applications in robotics was used for developing an environment in LabVIEW to use SR services provided by different providers. This is performed by taking advantage of highly trained advance deep learning neural network models. By using CAS, the use of local processing power is reduced by performing speech analysis by using the processing power of the cloud server.

In this research, SR services provided by IBM, Microsoft, and Google were investigated by developing an environment in LabVIEW. There is no need to develop a database for SR from this server. This would be inevitable if a SR system was developed using methodologies discussed in section 1.2.1. As there is no database, there is no comparison of input audio signal with database. Therefore, the only processing power required is to compare the input voice to database for SR. SR is performed using the Microsoft Bing Speech server. The results from Microsoft Bing Speech were reliable considering the domain of this research, which will be discussed in later sections.

1.3.1 Literature Review

The first industrial robot that was connected to a web server was in 1994, where users were allowed to teleoperate a robot via an internet browser [32]. They successfully connected IBM SR5427 SCARA arm robot to the web, where the robot accepts XYZ coordinates and angles in IEEE format and checksums [32]. Robotics and Automation systems using artificial intelligence have seen a rapid amount of growth in short time. In the near future, robots using this technology will be serving society, for families and individuals [3]. It will require a large amount of data in order to create advance artificial intelligence systems which can generate perception close to that generated by a human. Considering a service robot, very good amount of semantic knowledge in order to work independently between humans without supervision is required. The more data about the surrounding provided to the robot, the more "smarter" it will respond. Performing computation of all this data on local robotic hardware might require more computing speed and storage space. But in the case of a robot with limited dimensions and specifications, allocating more memory for software computation is not an option. Doing so might increase the cost of maintaining the system and managing the data. The concept of "Cloud Computing" will play an important role in the future for internet computing [3] [28].

Cloud Computing can be viewed as system to compute software and data over the internet [3] [4] [28]. The computational data and software is stored at data centers which support highly efficient hardware required for large computing loads. The concept of Cloud Robotics is based on an idea of Cloud Computing where large amounts of data and remote software is provided by different open source service providers for research in robotics and automation. The robots can remotely connect to the server and collect data whenever needed [3]. For example, in order to get knowledge of the surrounding, the robot can take pictures of the environment and upload them to the server, which provides recognition of the pictures sent. In the scenario of robots working collectively, this approach can help share data centrally via servers and learn from each other [3]. Figure 1.6 represents the concept of cloud robotics proposed by Fuji Ren from the University of Tokushima in Japan in 2011, where different robots share knowledge and feedback developed by each robot to save computing time used for generating the same results [3]. Cloud Computing gives robots access to a vast resource of data and information that are beyond the processing capacity of local database systems, such data is referred to as "Big Data" [28].

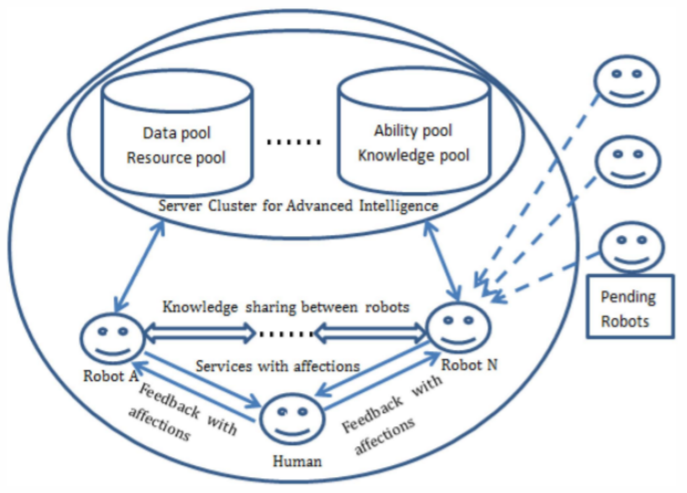


Figure 1.6. Concept of Cloud Robotics [3].

Cloud applications in robotics benefit robot programming in four ways as mentioned by Kehoe [28]:

- Big Data: access of using the global library for images, maps and object data [34] [35]
- Cloud Computing: parallel computing on demand for statistical analysis, learning, and motion planning [28] [36].
- Open Source/ Open Access: Sharing information such as codes and data obtained from testing, algorithm and hardware design [4] [37] [38].
- Collective Robot Learning: robot sharing trajectories, control policies and outcomes [4] [28].

Figure A.3 and A.4 represent the system architecture of object recognition performed during offline and online phases. In the offline phase, object recognition is performed by having each object recorded to train the object recognition model. Whereas in the online phase, object recognition is performed by sending a digital image to the Google object recognition engine to get the object's description from the server [4].

1.4 Object Recognition System

A vision system for recognizing object was not part of the objective of this research, but it was included as a feature after an environment was developed in LabVIEW to incorporated different cloud services. The vision system in the proposed system serves as a safety feature and also provides information of the object to the voice control system. This vision system uses Computer Vision API from Microsoft Azure [39]. Considering the development of robots that can effectively perform tasks such as cleaning tables, re-shelving books and grasping objects; the ability of robots to recognize and clearly distinguish between different objects is one of the most important factors for the robot in order to work safely and independently [4]. Errors in object recognition also raise safety concerns for the user. A vision system provides an "eye" to a robot in order to understand the surrounding and act appropriately.

Millions of people upload digital images everyday and there are ongoing projects to automatically classify images to specific categories as per the information in the image. An artificial neural network based model could be trained using sets of images that are required to be recognized by the system. Once again considering application of BAH in the real world as described in section 1.2, microprocessors are limited in processing power for applying real-time deep learning neural network models. So, the same concept that was applied to get SR from cloud services is also used to get object recognition from cloud services in LabVIEW.

1.4.1 Literature Review

Current approach of object recognition makes use of machine learning and artificial neural networks to categorize image content. A large number of image data is modeled and categories using different classifiers [40]. To get knowledge about an object model with a large learning capacity is required for millions of images trained.

There is method provided for image recognition, where digital images are recognized based on trained classifiers [41]. Training an image classifier required a collection of small regions of the image that corresponds to the face of the image and determine feature vectors for each image classifier. A subset of images is retrieved from the image classifier based on the distance between feature vectors [41].

1.5 Scope of Study

This research focuses on the development of a Human-Robot Interaction modality which uses voice as an interacting medium between humans and robots. The development of a vision system using CAS is also discussed which adds a safety feature to the proposed system. A brief explanation on the advantages and applications on using CAS for speech recognition over using traditional methods is described. The concept of using cloud applications is demonstrated in this study by using cloud computing services on a robotics platform. The Following is a detailed description of the content in each chapter.

Chapter 1 provides an introduction to HRI, speech recognition, CAS, and object recognition, along with examples and applications. Development of a BAH, software and required hardware are also introduced. A brief literature and possible methodologies for developing a speech recognition system is also described.

Chapter 2 discusses how an environment was developed in NI LabVIEW to integrate the cloud applications for speech and object recognition using Microsoft's and Google's Application Programming Interface (API). This chapter also discusses formating of the JSON result obtained from the API for further programming purpose and also different language support that this API offers. Two modes of controlling BAH named object grasping and individual finger control are discussed. It also demonstrates the algorithm developed to identify the action and object information in voice commands and also access grasping patterns of the object in a CSV file for object grasping task.

Chapter 3 discusses the methodology used to incorporate a vision system for safety and used in parallel with the voice control system in detail using the CAS. It also discusses the concept of using voice and vision system in parallel to each other for a safe operating environment. Detail description of features offered by Microsoft Vision API is also presented.

Chapter 4 discusses how the speech recognition system responded to voice commands evaluated with the help of people speaking different English accents. It also discusses the performance and assumptions of a object recognition system using a set of different objects in the grasping workspace. Difficulties observed in speech recognition and approach to avoid difficulties are also discussed. In this chapter results of operating voice and vision system in parallel are also discussed in brief.

Chapter 5 summarizes the research work performed on the development of voice and vision system using CAS on a BAH. This chapter also discusses some of the recommendations for future research.

CHAPTER 2

Development of Speech Recognition System Using Cloud Application Services

Parallel computing for processing voice commands in real-time is currently very popular. Some commercial sources such as Amazon's Elastic Compute Cloud [42], Google's Compute Engine [43], IBM Watson [44], and Microsoft Azure [39] provide parallel computing in very affordable and efficient way. As discussed in section 1.3, an approach of using CAS is considered for SR of voice commands. Therefore, an environment was developed in LabVIEW in order to interact with the cloud application service APIs.

In this approach, the speech processing is performed using Microsoft Bing Speech service on the Microsoft Azure server. On successfully analyzing the speech, server returns a JSON (Java Script Object Notation) file with speech to text results. This result is then deciphered in LabVIEW in order to use the text format of the result provided by Microsoft Bing Speech API. Microsoft Bing Speech is a service provided by Microsoft Azure, which is a cloud based intelligent service to get text recognition of an audio file. This provides an easy way to create voice control based applications. Figure 2.1 represents the flow process of developed voice control system. There are two modes of interaction with BAH developed using voice commands, namely object grasping and individual finger control. Figure A.1 represents the graphic user interface of SR system developed in LabVIEW using CAS.

The objective of object grasping is to grasp objects with known grasping pattern which are stored on a local database. This module recognizes which object to grasp

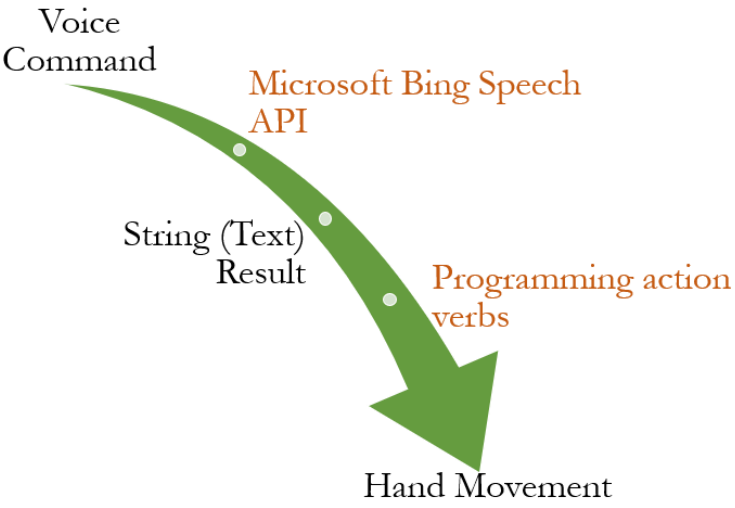


Figure 2.1. Proposed system architecture for voice control of BAH Using CAS.

from an issued voice command and identifies the grasping pattern from the database. Figure 2.2 represents different spherical and cylindrical objects which where used for grasping analysis and creating grasping pattern database as part of research in the MARS Lab. Certain action verbs such as *pick, place, grab*, and *drop* are used to engage with the objects such as cylinders, spheres, bottles, apples, etc. A voice command such as "*pick up an apple*" will identify the action to perform i.e. "*pick*" and object to operate on i.e. "*apple*".

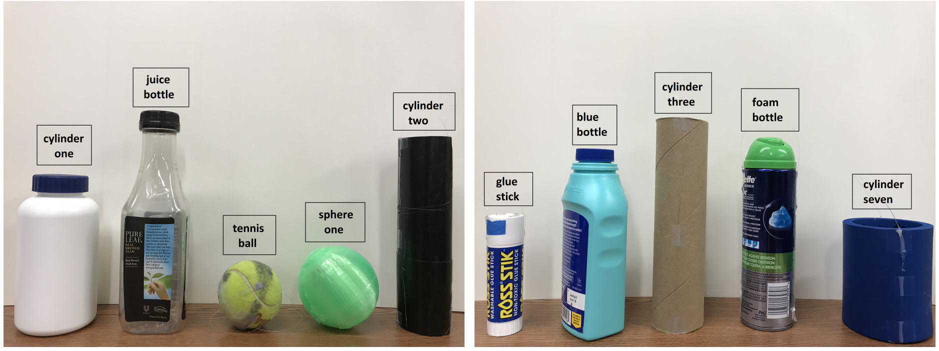
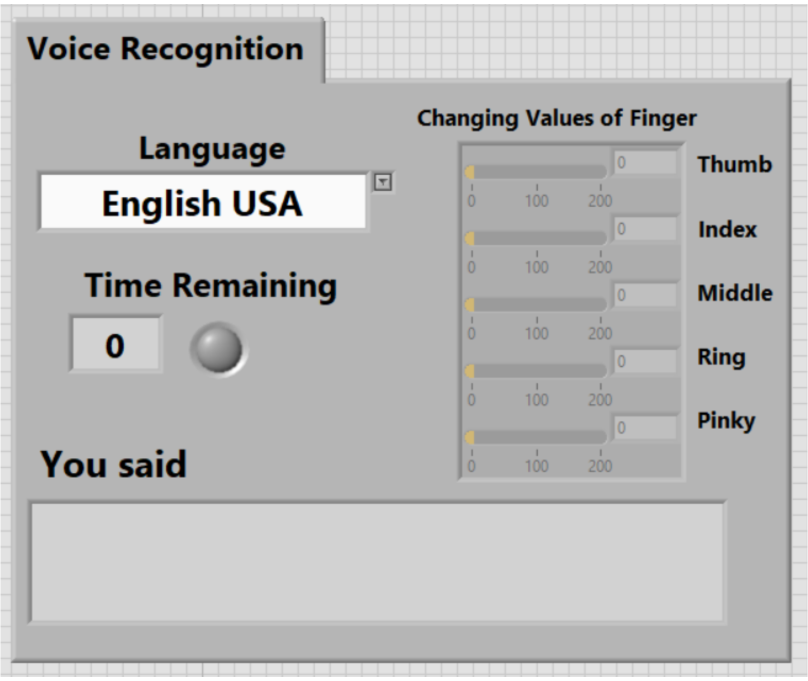


Figure 2.2. Set of objects used for creating grasping pattern database.

Whereas the objective of finger control modality is to control the position of each finger individually using voice commands. An example of voice command for this mode will be, "*close index 27 degree*", which closes the index finger motor by 27 degrees. Figure 2.3 represents the graphical user interface of the finger control mode, which shows changing values of the finger, issued command in tab "YOU SAID" at the bottom, countdown timer for recording voice command and "Language" selection in which the voice command is issued.



Voice Recognition

Language: English USA

Time Remaining: 0

You said

Figure 2.3. GUI for the finger control mode.

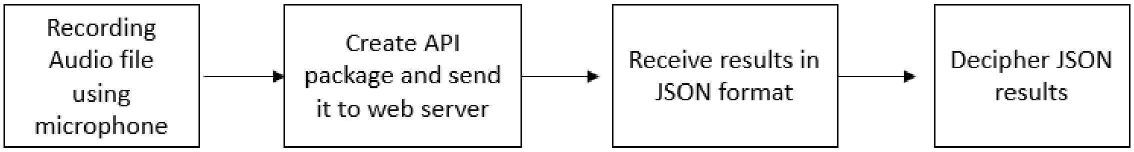

Recording Audio file using microphone
Create API package and send it to web server
Receive results in JSON format
Decipher JSON results

Figure 2.4. Process flow of getting SR using CAS.

Figure 2.4 represents the flow process for obtaining SR from the Microsoft Bing Speech API. Here, a voice command is recorded using a microphone within a specific time period defined by the user and then saved to the local memory as a ".wav" file format [45]. The audio is recorded in LabVIEW at a sampling rate of 22050 Hz with 16 bits per sample. After recording, the program saves the recorded audio as a physical file at the specified location on the local memory. The maximum length of audio transcription to text allowed by the Microsoft Bing Speech service is 15 seconds. The transcription to text of audio with content of more than 15 seconds is a paid service from Bing Speech, so the audio recording time is restricted to 7 seconds (this can be changed as required). For this research, a maximum of 15 seconds of audio content is sufficient for recording voice commands.

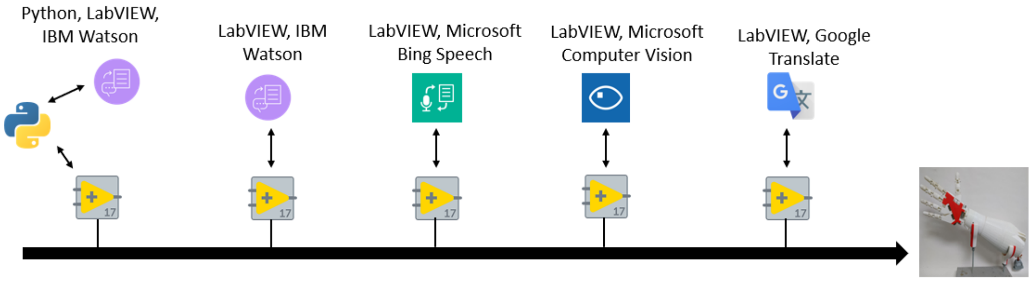

Python, LabVIEW, IBM Watson
LabVIEW, IBM Watson
LabVIEW, Microsoft Bing Speech
LabVIEW, Microsoft Computer Vision
LabVIEW, Google Translate
17
17
17
17
17

Figure 2.5. Path followed to developed an environment in LabVIEW to incorporate different APIs.

2.1 Microsoft Bing Speech API (Application Program Interface)

In order to develop a voice control system using SR services, a Python program was developed to get SR of voice commands in LabVIEW using an add-on named LabPython for integrating LabVIEW and python programming language [46]. This program used IBM Watson speech-to-text service to receive SR of voice commands

[44]. But as the control module for BAH was developed on NI LabVIEW, a single software system was desired in order to avoid software conflicts in future. Figure 2.5 represents the path followed to develop an environment in LabVIEW that allows the user to integrate different APIs with LabVIEW.

As presented in Figure 2.5, IBM Watson was used as SR engine before using Microsoft Bing Speech as it requires a lower number of headers to make a successful API call. But, IBM Watson is a paid service so using the same concept and more headers, an environment was developed in LabVIEW to get SR from Microsoft Bing Speech. To get access to Bing Speech API, one needs to create a Microsoft Azure account and get an API key which is essential for the the package developed sent to the API. An API key is most essential in order to use the API services. A POST request method is used to send a web requests to the server, that accepts the data encoded in the package of request message [47]. LabVIEW provides a helpful feature of HTTP client in data communication toolbox, which allows the user to send web requests to servers using functions such as GET, PUT, POST, and POST Multiple. Figure 2.7 gives a description on POST and add Header SubVI used to make an API call within LabVIEW.

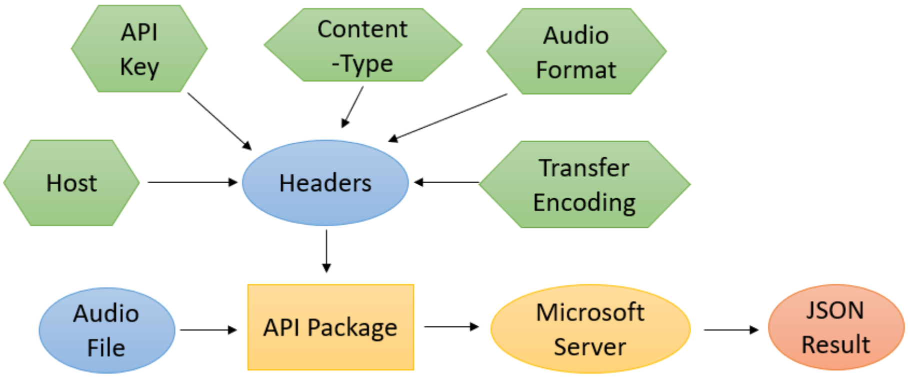
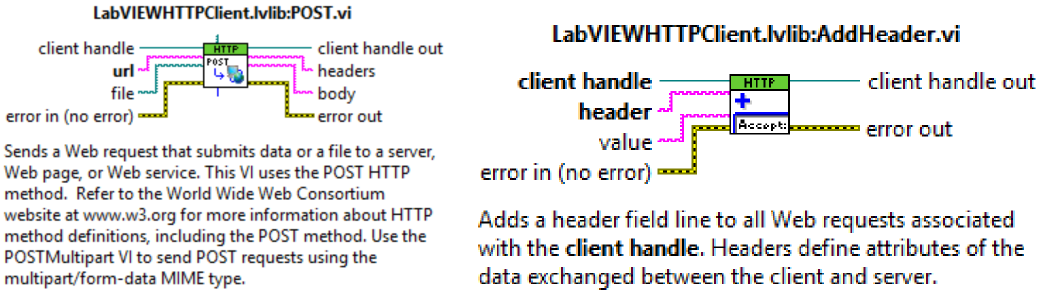


Figure 2.6. Formating an API package for SR using headers and audio file.

The Header SubVI is used to add appropriate request headers to the API package since they are required by the server, as shown in Figure 2.6. All the headers such as the server host, content type encoded, content length, authorization key, transfer encoding and content type for audio are required with the audio file in order to get SR [45]. Values of certain headers remains constant for many users and only the API key changes. The file location, where the audio file is physically present in the memory, is added to POST Sub-VI with the URL of API server. Bing Speech Recognition service allows SR in 29 different languages [48].



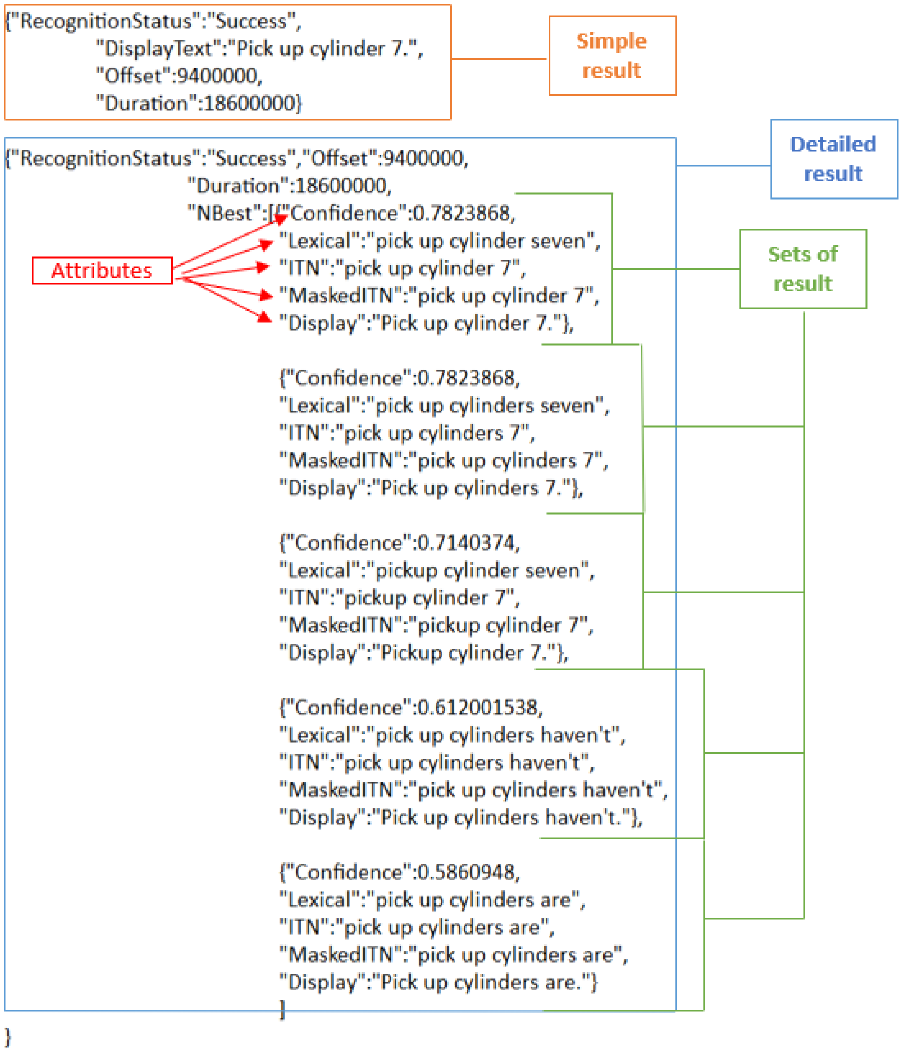
Sends a Web request that submits data or a file to a server, Web page, or Web service. This VI uses the POST HTTP method. Refer to the World Wide Web Consortium website at www.w3.org for more information about HTTP method definitions, including the POST method. Use the POSTMultipart VI to send POST requests using the multipart/form-data MIME type.

Adds a header field line to all Web requests associated with the **client handle**. Headers define attributes of the data exchanged between the client and server.

Figure 2.7. POST and add header SubVI in the LabVIEW.

2.2 Result in JSON format

The SR API returns a JSON (Java Script Object Notation) file as a result. JSON (Java Script Object Notation) is an object notation format used for transmitting data over the web that contains different data types such as numerics, strings, arrays, etc. The JSON result has a stepped structure which contains different attributes and values associated to the attribute. Bing Speech offers two formats of results for SR based on the information provided in the web request. One format is “Detailed” and another is “Simple” as shown in Figure 2.8. The *simple* format contains recognition status and recognized text in the result, whereas the *detailed* format contains an attribute N-best, which incorporates an array of results based on the confidence score as represented in Figure 2.8.



```
{"RecognitionStatus":"Success",
          "DisplayText":"Pick up cylinder 7.",
          "Offset":9400000,
          "Duration":18600000}
```

```
{"RecognitionStatus":"Success","Offset":9400000,
                    "Duration":18600000,
                    "NBest":[{"Confidence":0.7823868,
                              "Lexical":"pick up cylinder seven",
                              "ITN":"pick up cylinder 7",
                              "MaskedITN":"pick up cylinder 7",
                              "Display":"Pick up cylinder 7."},

                             {"Confidence":0.7823868,
                              "Lexical":"pick up cylinders seven",
                              "ITN":"pick up cylinders 7",
                              "MaskedITN":"pick up cylinders 7",
                              "Display":"Pick up cylinders 7."},

                             {"Confidence":0.7140374,
                              "Lexical":"pickup cylinder seven",
                              "ITN":"pickup cylinder 7",
                              "MaskedITN":"pickup cylinder 7",
                              "Display":"Pickup cylinder 7."},

                             {"Confidence":0.612001538,
                              "Lexical":"pick up cylinders haven't",
                              "ITN":"pick up cylinders haven't",
                              "MaskedITN":"pick up cylinders haven't",
                              "Display":"Pick up cylinders haven't."},

                             {"Confidence":0.5860948,
                              "Lexical":"pick up cylinders are",
                              "ITN":"pick up cylinders are",
                              "MaskedITN":"pick up cylinders are",
                              "Display":"Pick up cylinders are."}
                             ]
}
```

Figure 2.8. Types of result format offered by Bing Speech API, Simple(orange) and Detailed(blue).

Considering the results from the *simple* format, the recognition status of audio and conversion to text is show in the “RecognitionStatus” and “DisplayText” fields, as presented in Figure 2.8. There is a sets of results expected in the “RecognitionStatus”

Table 2.1. Various types of Recognition Status from the API results

Status	**Description**
Success	Recognition is Successful and Text is present
NoMatch	Speech detected but no word from the target language
InitialSilenceTimeout	Silence in audio and no content

field, as shown in Table 2.1 [49]. The "Offset" field shows the "offset" time in an audio file and "duration" shows the length of an audio file.

As shown in Figure 2.8, *detailed* result format contains different sets of information available in one recognition, such as confidence, N-best values, Lexical form, Inverse Text Normalization (ITN), Masked ITN, and Display Text. Each of these attributes in the JSON result are discussed in detail as follows.

The results of detailed format JSON file contains various elements like confidence, which shows the confidence score calculated by *confidence classifier*, which are the trained set of features developed by Microsoft. This confidence classifier discriminates maximally between the correct and incorrect recognition [49]. In essence, confidence score shows the accuracy of speech to text.

Lexical form shows the conversion of speech-to-text in lexical format if there is any numeric quantity present in voice commands. For example, if a voice command "pick up cylinder 7" is issued, it will return "pick up cylinder seven" in the lexical form (if recognized correctly). It is useful if the user wants a different format of results for further programming based on the text. One has to be cautious while using lexical form of representation as profanity is not masked in this type.

Another type of result is ITN, which helps convert text from the canonical form to numeric form. If a voice command "close middle finger thirty nine degrees" is issued, ITN will return "close middle finger 39 degrees". It is an inverse of text normalization. Similar to the lexical form, ITN does not hide profanity in the result text. But detailed format provides “Masked ITN”, which helps mask profanity in the result text. At last the “Display” attribute in the JSON result adds a grammatical sense to the sentence structure, such as punctuation and capitalizing first alphabet, which helps in reading and understanding text correctly.

As mentioned in the discussion before, LabVIEW i3 JSON tool kit is used to parse or decipher JSON result and get the content of results. Figure 2.9 gives information on the JSON sub-VIs and the process to obtain the results from the JSON file. The “Get Array” Sub-VI in LabVIEW will parse out any array in the JSON file whereas sub-VI “Get Object Parameter” will parse out the values in the object attribute. The object parameter can be a string, numeric, integer, etc. The final output from the JSON VI is shown in string form and used for further programming.

Before parsing the JSON result, the program will check value for attribute “RecognitionStatus”. If the value of “RecognitionStatus” is “Success” then only the program will go further and parse the JSON result, otherwise it will return “SAY COMMAND AGAIN”.

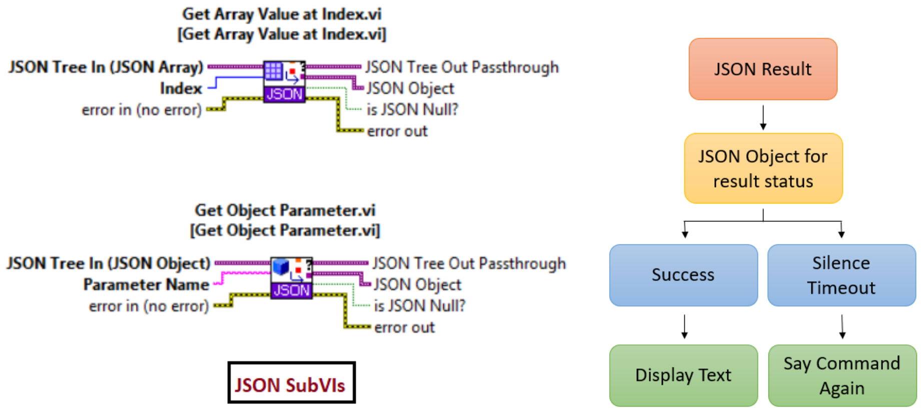


Figure 2.9. JSON Sub-VIs(left) and process flow to get display text(right) out from the JSON file.

2.3 String Search Algorithm for Action Verbs and Object Names in the Results

In order to successfully execute voice commands on the BAH, different action verbs and object names were programmed to the system. A string search algorithm was developed to identify any action verb or object name in the voice commands. For example, if a voice command “Pick up cylinder 27” is issued this algorithm will search for the action verb “Pick” and object name “cylinder 27” in order to search for the grasping pattern of “cylinder 27” in the database.

Once the string is deciphered from the JSON result, all the alphabets of the string are formated to lower case in order to make search string operation easy. The string indicator on the front panel of VI it will display the original results. The “match regular expression” function from the LabVIEW was used to perform a search operation on result string.

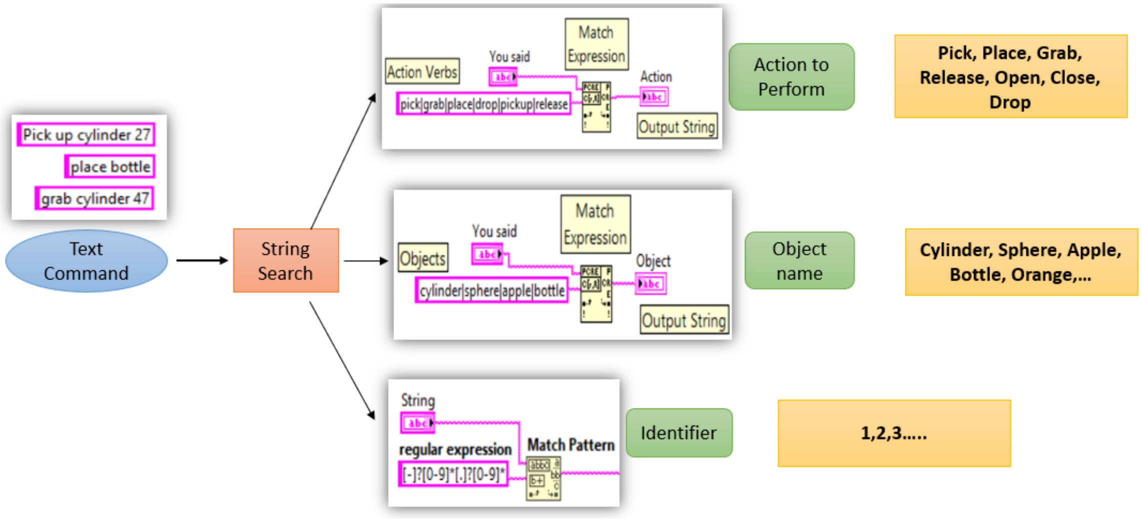


Figure 2.10. Process flow to search for the action verbs and object names in the string result.

As shown in Figure 2.10 seven action verbs are used for the search operation and programming, namely pick, place, drop, grab, release, open, and close. There are also four object names used for searching, namely cylinder, sphere, apple, bottle. Note that the user can add more objects in the database for recognition. After the action verbs and object names are acquired from the search operation, a string is concatenated to search for any object identifier such as a number in string result. Identifiers depend on the object database and can vary according to the algorithm objective.

2.4 Action Verbs Programming and Database Search for Grasping Pattern of Object

After acquiring action verbs from the search operation, a case structure is programmed in LabVIEW which activates different case based on action verbs present in the voice command, as shown in Figure 2.11. For example if action verb "release" is identified, it will select the case structure "release", as shown in Figure 2.11. The same procedure is followed for all the other action verbs such as "pick", "grab", "drop", etc. The action verbs "release" and "drop" should release the object, action verbs such as "grab" and "pick" should grasp the object. An array of 0 is passed for action verbs "release" and "drop" since 0 is the motor angle that opens the fingers in the hand. For action verbs such as "grab" and "pick", a database search operation for the object grasp pattern is performed, as presented in figure 2.11.

Figure 2.12 shows a part of the database (in CSV file format). Each column has a specific name. The first column (Objects) is pulled out and a search operation is performed for the object name in the voice command. Once the object name is located, the rows following to that cell are acquired as an array and displayed as an array. The grasping pattern values are then send to the myRIO to grasp the object [21].

The database Sub-VI contains grasping patterns for different test objects used in this research. Objects were named according to closest shape they resemble such

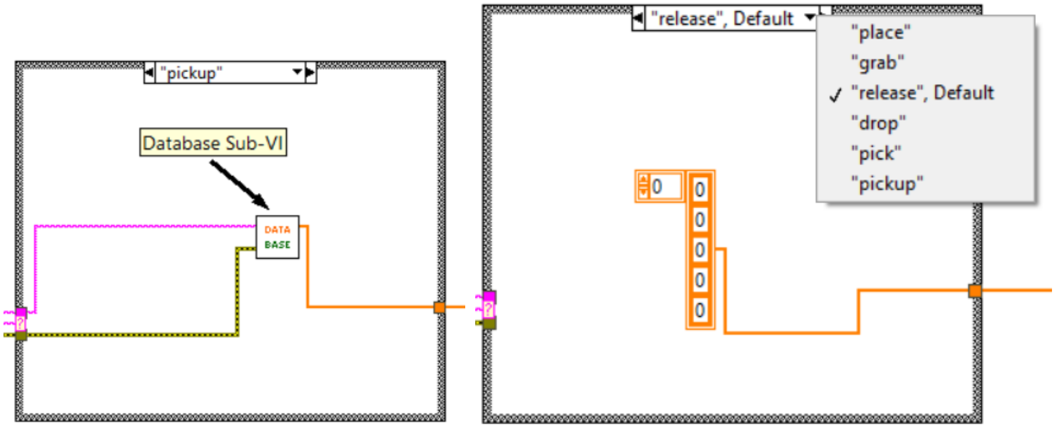
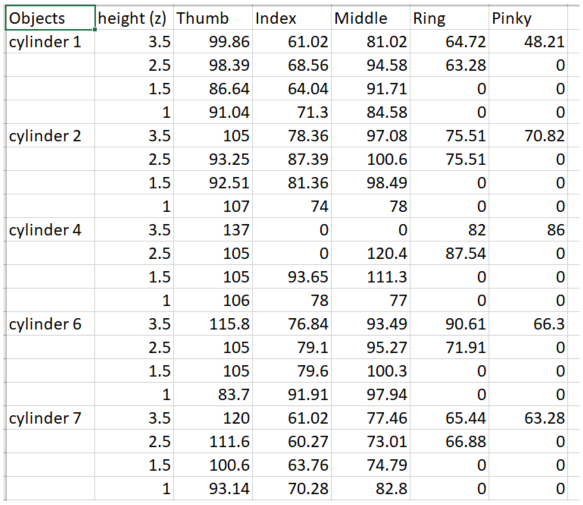


Figure 2.11. Programming a case structure for all the action verbs.

as spheres and cylinders. For example cylinder 1, cylinder 2, sphere 1, sphere 2 and so on [21]. Figure 2.13 represents the search operation performed to identify grasping patterns from the CSV file.



Objects	height (z)	Thumb	Index	Middle	Ring	Pinky
cylinder 1	3.5	99.86	61.02	81.02	64.72	48.21
	2.5	98.39	68.56	94.58	63.28	0
	1.5	86.64	64.04	91.71	0	0
	1	91.04	71.3	84.58	0	0
cylinder 2	3.5	105	78.36	97.08	75.51	70.82
	2.5	93.25	87.39	100.6	75.51	0
	1.5	92.51	81.36	98.49	0	0
	1	107	74	78	0	0
cylinder 4	3.5	137	0	0	82	86
	2.5	105	0	120.4	87.54	0
	1.5	105	93.65	111.3	0	0
	1	106	78	77	0	0
cylinder 6	3.5	115.8	76.84	93.49	90.61	66.3
	2.5	105	79.1	95.27	71.91	0
	1.5	105	79.6	100.3	0	0
	1	83.7	91.91	97.94	0	0
cylinder 7	3.5	120	61.02	77.46	65.44	63.28
	2.5	111.6	60.27	73.01	66.88	0
	1.5	100.6	63.76	74.79	0	0
	1	93.14	70.28	82.8	0	0

Figure 2.12. Small portion of the CSV database.

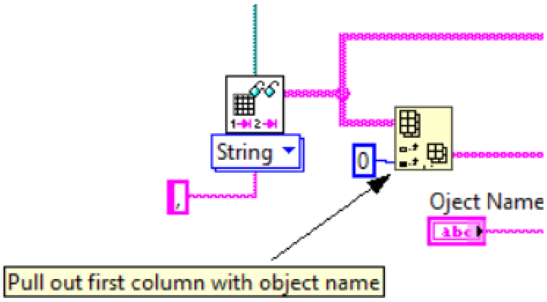


Figure 2.13. Indexing the values to perform a database search.

2.5 Individual Finger Control Module

As discussed in section 2 this mode of interaction with BAH controls the motion of each finger individually. This mode follows the same SR algorithm described in Figure 2.6. The SR results are obtained from the Microsoft Bing Speech API, but processing varies from object grasping mode as the format of voice commands change. Unlike commands for object grasping mode which include action verbs such as pick, place, drop, and release, in this mode action verbs are open, close, and reset finger. The output of this mode focuses on moving specific finger a specific amount in degrees. An example of voice command for finger control would be "open middle finger 27 degrees" or "close ring finger 65 degrees".

After acquiring the speech recognition result, a similar string search operation is performed, as described in section 2.3, to acquire information the action to be perform (finger to control and amount to operate). A number search in the text results will identify any numeric quantity in the voice command. Figure 2.14 outline of process used to search for specific string in the voice command.

A cascade case structure is programmed in a while loop using the action verbs as a case control element. After the completion of each loop, the values for finger angle are retained for the next occurring loop until a "stop" voice command is issued.

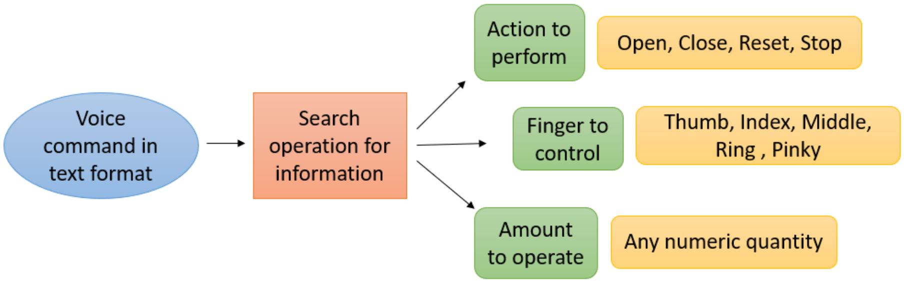
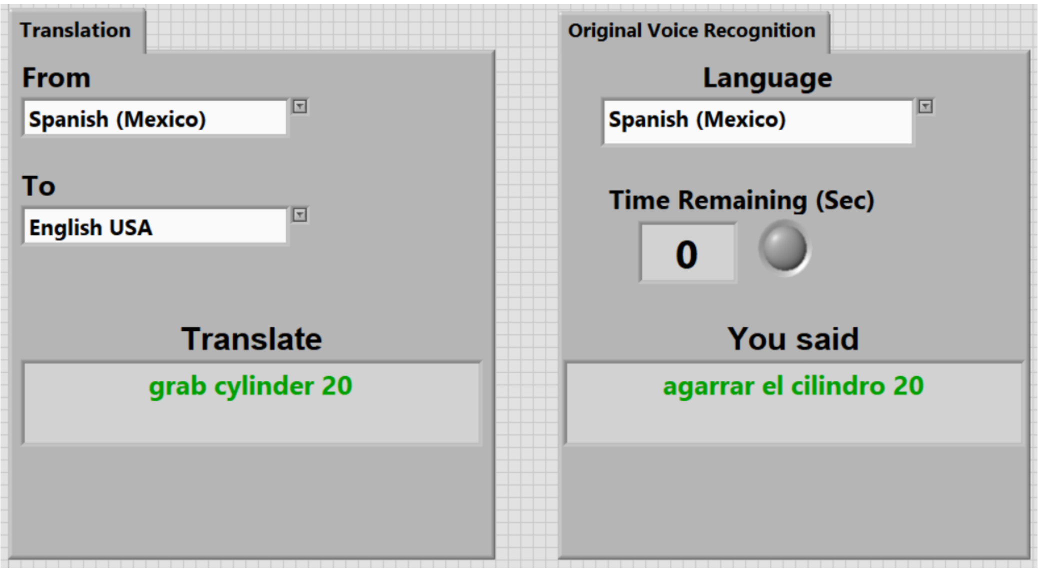


Figure 2.14. Search process for the finger control mode.

There are two parameters that control two different case structures namely action verb (open, close, reset) and finger identifier (thumb, index, middle, ring, pinky). Action "open" executes the case structure to open the finger with an amount specified in the voice command. It returns with voice instructions "Cannot open more" if the finger is fully open. Similarly action "close" causes the finger to close a specified amount. Whereas the action "reset", all the motor angles for each finger are set to zero.

2.6 Additional Language Support and Language Translation

In this research, additional languages are evaluated for SR. The Bing Speech API provides 29 additional language for expanding the workspace of the developed system. It was chosen to evaluate the system using Spanish language. A translation process from Spanish text into English was performed using another CAS, Google Translate [50]. Figure 2.15 represents the process algorithm to get spoken language translation to English language. In Figure 2.15 translation of a voice command issued in Spanish "agarrar el cilindro veinte" to English language "grab cylinder veinte" is shown. Once the voice command is successfully translated to English, the string (text) follows the same search process as described in section 2.3. An approach of using Spanish text and not translating to English can also be applied by performing search of action verbs and object identifiers in Spanish. A database of objects grasping pattern in Spanish would be required in order to search grasping pattern of the object in Spanish text.



From
Spanish (Mexico)

To
English USA

Translate
grab cylinder 20

Language
Spanish (Mexico)

Time Remaining (Sec)
0

You said
agarrar el cilindro 20

Figure 2.15. LabVIEW GUI using the Google Translate API to translate from Spanish(left) to English(right).

Figure 2.16 is a block diagram representation of using Google Cloud Translate API in the LabVIEW. This follows the same process as it required the use of Microsoft Bing Speech in LabVIEW. Changes were made in adding headers to API package as header requirement of Translate API differs from Speech API. The JSON result of Google Translate API (presented in Figure 2.17) is then deciphered in LabVIEW. Parsing the JSON result depends on the format in which the attributes are arranged in one data structure.

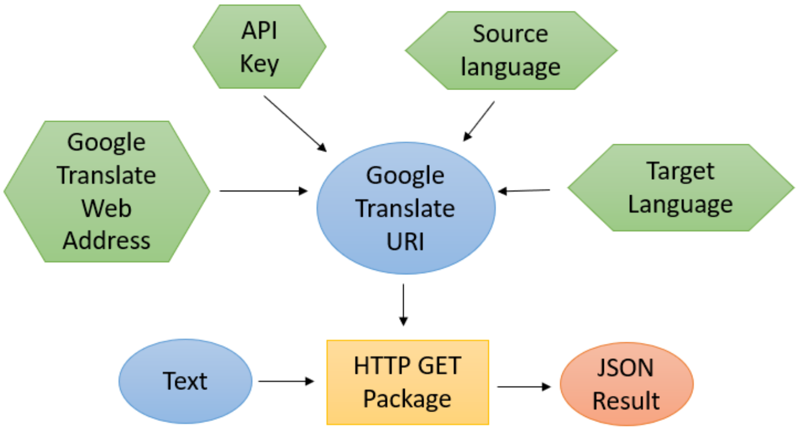
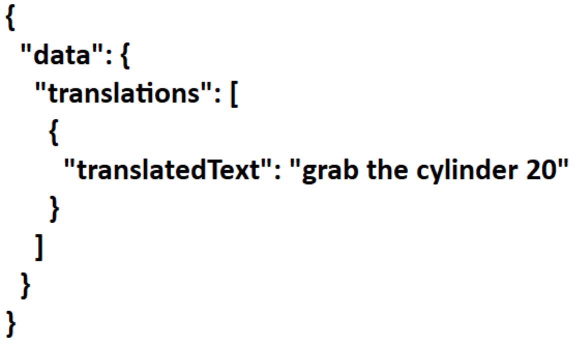


Figure 2.16. Formating an API package for Google translate using all required headers.



```
{
 "data": {
  "translations": [
   {
    "translatedText": "grab the cylinder 20"
   }
  ]
 }
}
```

Figure 2.17. JSON result of a Google Translate API call.

CHAPTER 3

Object Recognition as a Smart Safety Feature to the Voice Control System

Once an environment was developed to incorporate different APIs in LabVIEW and parsing the results obtained from APIs. An additional API from Microsoft i.e. Computer Vision was incorporated in LabVIEW by using NI IMAQ. The MCV service provides information on image that is sent to the server [51]. This proved as a helpful feature once incorporated along with the voice control system. MCV allows us to get information on object to be grasped that is present in the workspace. An assumption was made while using this vision system, that all the images that are sent to the server are taken in appropriate lighting on the object and object is appropriately distinguishable in the image. The vision API performs analysis on the visual content in the images and returns with information on the image in various forms such as tag, description, adult face, categories, etc. The API offers recognition in various different types and Table 3.1 shows types used for this research.

The vision system is used to verify whether the voice command is applicable on BAH or not. Considering a scenario where a voice command is issued "pick

Table 3.1. Different Recognition type offered by MCV API

Type	**Description**
Tag	tag based on scenery and action
Categorize	categorize based on visual features
Description	generate description based on image content

up an apple", and there is a bottle in front of the BAH, then executing the voice command and applying the grasping pattern of an apple to grasp the bottle, might apply inappropriate force to the bottle. This situation can result in under grasping of the object or apply more force then required and may result in damage to either the BAH or bottle. So, it was important to have a system that can prevent happening of this kind of disasters.

3.1 Capture Image to Local Memory

NI LabVIEW's motion and vision toolbox is used to capture image within LabVIEW and store it in the local memory [52]. Figure A.5 represents the user interface for vision system with different type of recognitions that can be requested from the MCV. While the program is running, preview of image is displayed in image display box shown in figure A.5. Finally the result obtained from MCV API is shown in Image Recognition box at the bottom of figure A.5.

Logitech HD 720p camera is used to take pictures of object in workspace to be grasped by the BAH. NI IMAQ is used to interface LabVIEW with external camera to take pictures of the objects. A while loop helps the program running until a boolean is pressed which simultaneously take picture and stop the while loop after saving the image to the local memory. The image stored in the local memory is then send to the Microsoft Vision API server for recognition. The challenge here was to take the picture of the object and save it to the local memory with different names every time, because in some cases LabVIEW might not have permission to overwrite the new picture file with the previous one. So, a time stamp is used with the file name that changes file name to that present time and it can be saved automatically.

3.2 Creating API Package for Microsoft Computer Vision API

There are some sets of prerequisites, for the images that are sent to the server, specified by Microsoft which are to be fulfilled before requesting the object recognition service. The API supports image in four formats namely JPEG, PNG, GIF and BMP with size less than 4 MB (Megabyte). The image dimensions should not exceed 50 x 50 pixels. The HTTP POST call is used in LabVIEW using web toolbox to send web requests to the Microsoft server as a package that includes image file and required headers for image recognition. The file location of captured image is attached to the web request to sent it to the API. The results of this POST web requests sent to the server are received in a JSON format. Requesting results from the Microsoft Vision API service using HTTP POST call follows the same structure as shown in Figure 2.6. An API subscription key issued from Microsoft Azure portal is required in order to complete this web requests. Getting the results out of the JSON format follows same programming structure as described in SR system development section 2.2.

All the recognition types offered by the Microsoft Vision API has different attributes in the JSON data structure. Figure 3.2 represents the JSON result requested to the server for recognition type “Description”. Figure 3.2 shows the recognition results that includes “description”, “captions”, and “metadata” for the image 3.1. An image caption of “a man in glasses looking at camera” is returned as a result for the image 3.1 as shown in figure 3.2.

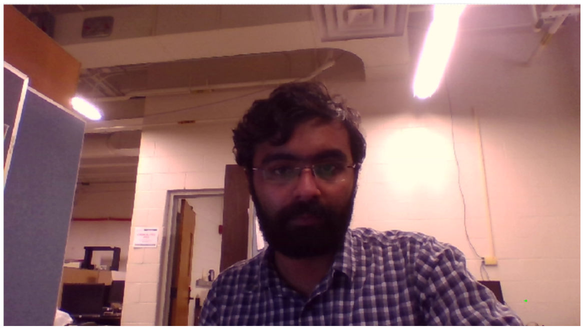


Figure 3.1. Image sent for recognition to the API with obtained result as 3.2.

3.3 Voice and Vision System in Parallel

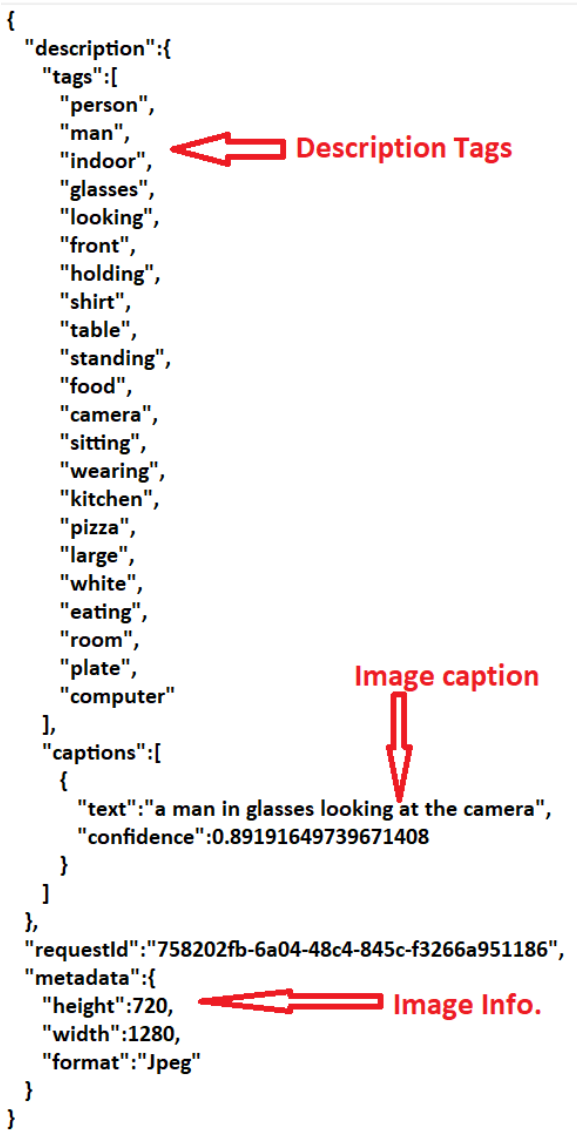
The speech and object recognition system developed using Microsoft CAS can be used in parallel with the vision system, to verify that the voice command issued for grasping the objects are appropriate and applicable. So, an environment was developed in LabVIEW that uses both the speech and object recognition service from Microsoft and compare the results obtained from both the services in order to further execute the voice commands. Once the recognitions obtained is deciphered in LabVIEW, the system identifies the object name present in both the speech and object recognition results. After getting the object information from both the recognitions, this system will compare the results to check whether the object information has a match or not. If there is a conflict in the information of object from both the recognition results, then the system will not execute the voice command. This feature of object recognition allow the user to avoid damage to either the objects or BAH because of in appropriate grasping.

Figure 3.3 represents the graphic user interface of voice and vision system running in parallel. The SR is performed on the left part of figure 3.3 in preferred language and the object recognition of the image is performed on the right. The "Object Result" tab on the top left of figure 3.3 represents the object information after comparing the results obtained from both the systems, that whether there is a conflict or not. Whereas the “Output” tab on the top right of figure 3.3 provides description on the final result. If there is no conflict in the object information obtained from speech and object recognition, then the system will proceed further and identify grasping pattern for the object from the database.

Figure 3.4 represents the result of a match in the object information from speech and object recognition. The “Object Result” tab gives information that the object is an “Apple” and the “Output” tab describes that “This is apple” and at the extreme right of the figure 3.4 the grasping pattern for apple is retrieved from the database. But, In case of conflict in object information between speech and object recognition as shown in figure 3.5 the system will show “conflict” in “Object Result” tab and description of actual object present in the workspace in “Output” tab. In the case of conflict, the voice command “pick up orange” was issued even though there was an apple present in the workspace. But the vision system identified the object correctly and restricted the system to get grasping pattern of orange as represented in the figure 3.5.

The vision system also adds one more feature of providing information of the object to the voice control system. In this scenario, if an voice command “pick up object” is issued, with no information on object as shown in figure 3.6. The vision system will provide the information by identifying the object and will also search for the grasping pattern of that object in the database. Figure 3.6 exemplifies the concept where vision system identifies an “apple” in the workspace and then the search algorithm identifies the grasping pattern of “apple” from the database.

In combined, the voice and vision system will first search for object information in the voice command, if it identifies the object information in the voice command then it will compare this information with the information provided by the vision system and generate a result. But, if the system does not identify the object information in the voice commands, then the vision system will provide the information and execute the task. In this way the vision system serves two purposes, that are safety and providing information whenever its needed. This makes the combined system (voice and vision) more safer to operate and smarter to react.



```
{
  "description":{
    "tags":[
      "person",
      "man",
      "indoor",
      "glasses",
      "looking",
      "front",
      "holding",
      "shirt",
      "table",
      "standing",
      "food",
      "camera",
      "sitting",
      "wearing",
      "kitchen",
      "pizza",
      "large",
      "white",
      "eating",
      "room",
      "plate",
      "computer"
    ],
    "captions":[
      {
        "text":"a man in glasses looking at the camera",
        "confidence":0.89191649739671408
      }
    ]
  },
  "requestId":"758202fb-6a04-48c4-845c-f3266a951186",
  "metadata":{
    "height":720,
    "width":1280,
    "format":"Jpeg"
  }
}
```

Figure 3.2. JSON result for recognition type “Description” and for image 3.1.

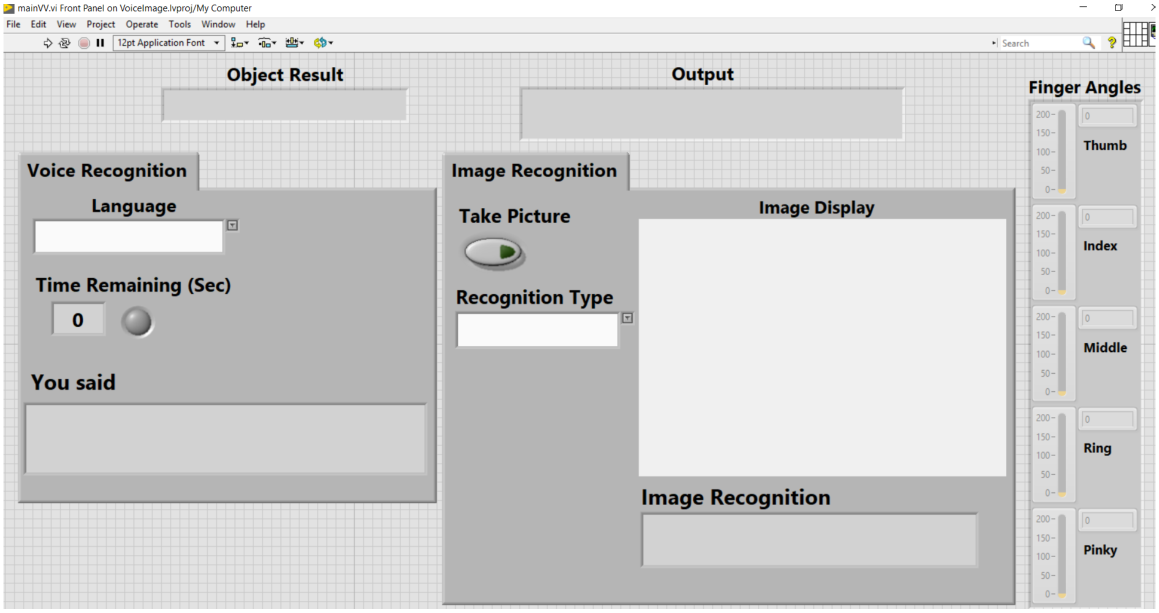
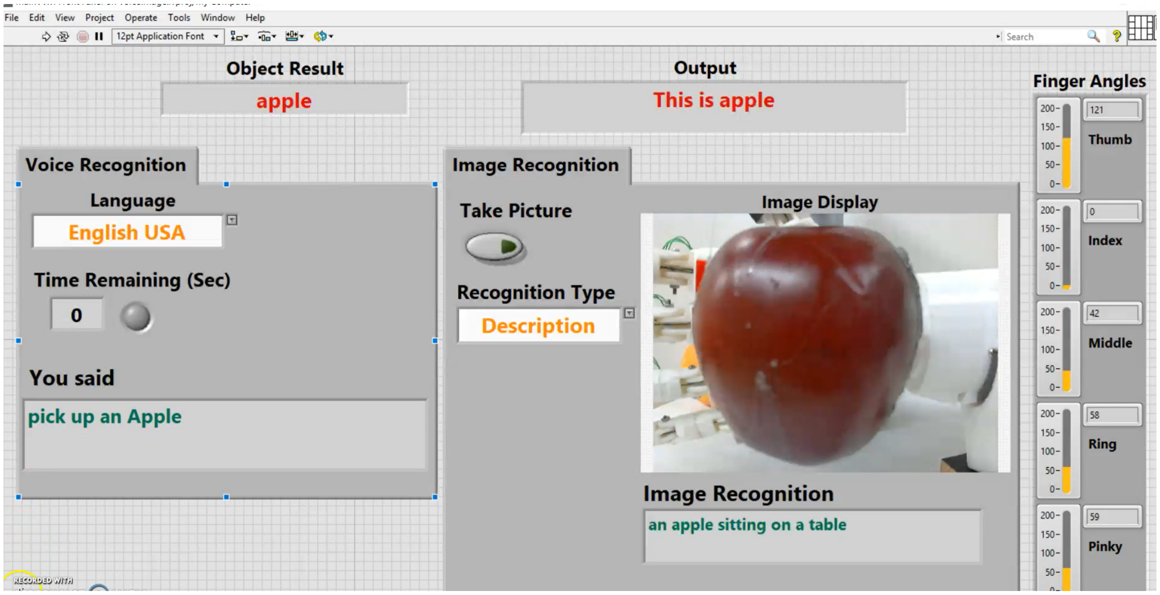


Figure 3.3. GUI of Voice and Vision System in parallel.



Object Result: apple

Output: This is apple

Voice Recognition

Language: English USA

Time Remaining (Sec): 0

You said: pick up an Apple

Image Recognition

Take Picture

Recognition Type: Description

Image Display

Image Recognition: an apple sitting on a table

Finger Angles

Thumb: 121

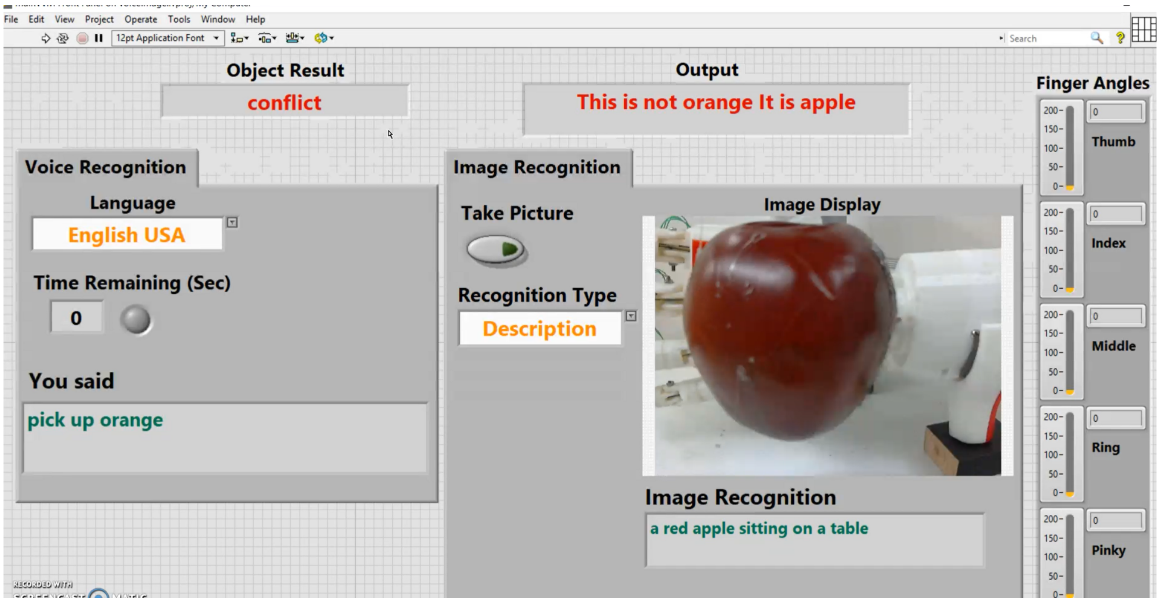
Index: 0

Middle: 42

Ring: 58

Pinky: 59

Figure 3.4. Match in the results of speech and object recognition.


File Edit View Project Operate Tools Window Help
12pt Application Font
Search
200
150
100
50
0
RECORDED WITH SCREENCAST-O-MATIC

Object Result
conflict

Output
This is not orange It is apple

Voice Recognition

Language
English USA

Time Remaining (Sec)
0

You said
pick up orange

Image Recognition

Take Picture

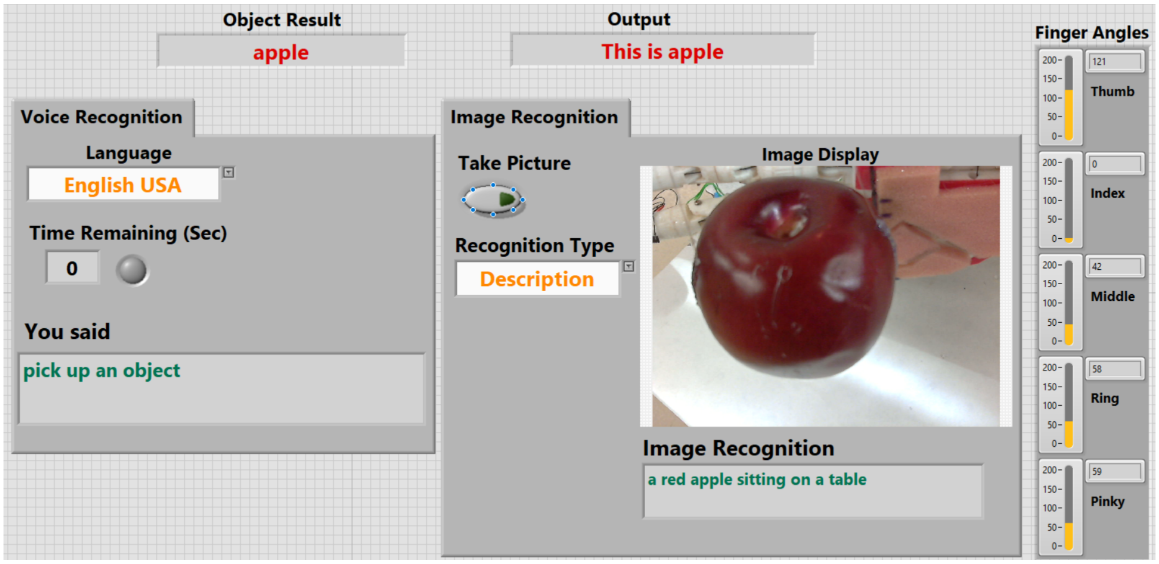
Recognition Type
Description

Image Display

Image Recognition
a red apple sitting on a table

Finger Angles
Thumb 0
Index 0
Middle 0
Ring 0
Pinky 0

Figure 3.5. Conflict in the results of speech and object recognition.



Object Result
apple

Output
This is apple

Voice Recognition

Language
English USA

Time Remaining (Sec)
0

You said
pick up an object

Image Recognition

Take Picture

Recognition Type
Description

Image Display

Image Recognition
a red apple sitting on a table

Finger Angles
Thumb 121
Index 0
Middle 42
Ring 58
Pinky 59

Figure 3.6. Results of the object information provided from the vision system.

CHAPTER 4

Results and Discussion

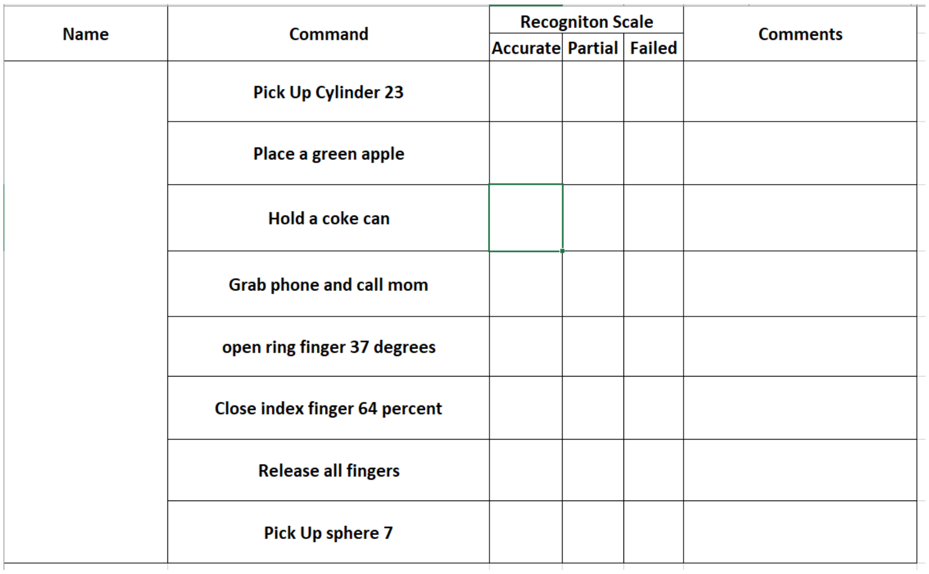
In this chapter, the results from the proposed experiments of the developed system will be discussed.

4.1 Speech Recognition Evaluation Results

4.1.1 Speech Recognition Evaluation in English

In this research, the SR of the voice commands was performed using CAS (Microsoft Bing Speech from Microsoft Azure). As mentioned earlier in section 2.1 the Bing Speech API allows the user to remotely perform SR using Microsoft Servers and acquire the result in textual format. In order to verify that the system recognizes voice commands within the workspace of this research, the performance was evaluated with 8 commands that included a combination of different action verbs, object names and object identifiers. Figure 4.1 represents the format of test performed with the help ten individuals that have different English speaking accents. All the individuals were asked to issue each voice command twice, which results in set of 16 commands issued per person. An accuracy out of 100 for each command was tabulated as shown in Table 4.1. For recording an audio in LabVIEW, a Snowball microphone from Blue was used for recording audio.

In the SR evaluation likert scale, “Accurate (100)” was chosen if the recognized the spoken word correctly. If the SR recognizes even a single spoken word (out of the entire command) incorrectly, then the test would be scored as “Partial (50)”. The



Name	Command	Recogniton Scale			Comments
		Accurate	Partial	Failed	
	Pick Up Cylinder 23				
	Place a green apple				
	Hold a coke can				
	Grab phone and call mom				
	open ring finger 37 degrees				
	Close index finger 64 percent				
	Release all fingers				
	Pick Up sphere 7				

Figure 4.1. Likert scale based sheet developed for SR evaluation.

score of "Failed (0)" was chosen if the system did not recognizes the command.

From the results in Table 4.1, it was concluded that the system responds to the defined action verbs and object names expected for recognition. However, some difficulties were encountered in recognition of words such as "sphere" and "thumb". After further evaluating, it was assumed that this difficulty was because of the rhyming na-

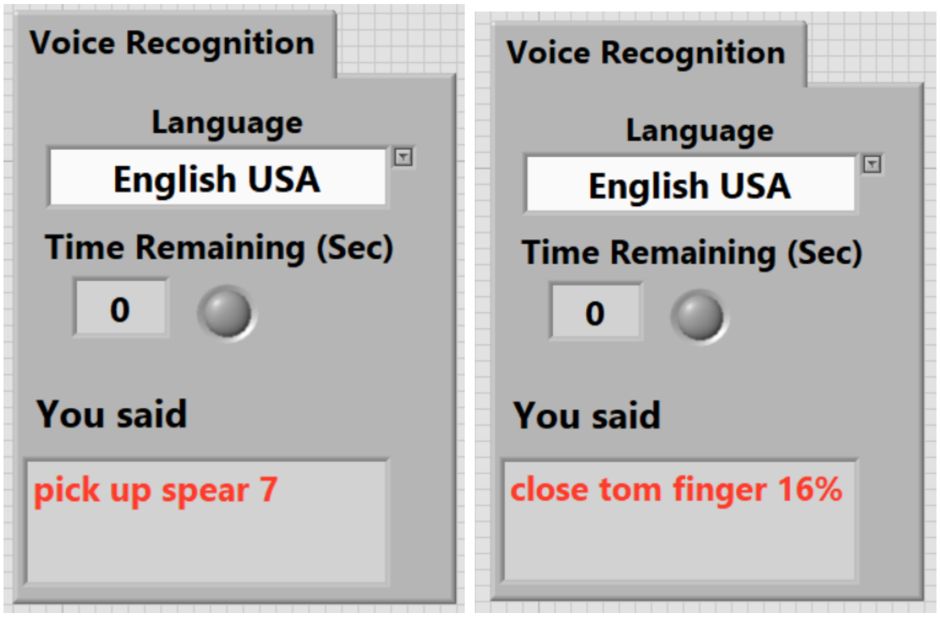
Table 4.1. SR evaluation results performed in English

Sr No.	**Command**	**Accuracy (of 100)**
1	Pick up cylinder 23	87.5
2	Place a green apple	95
3	Hold a coke can	97.5
4	Grab phone and call mom	90
5	Open ring finger 37 degrees	95.8
6	Close index finger 64 percent	90
7	Release all fingers	90
8	Pick up sphere 7	57.5

ture of the words, but this difficulty in recognition was not persistent. There were occasions where these rhyming words were recognized correctly. Because of this difficulty, the accuracy of voice command 8 in Table 4.1 is less compared to the other voice commands since it contains the word “sphere”. During SR evaluation, in many occasions the word “sphere” lead to incorrect recognition of other words such as “spear”, “fear”, and “spare” as presented in Figure 4.2. In order to address this issue of incorrect recognition an algorithm was developed that converts these 3 incorrect recognitions into the correct word “sphere”. This same procedure was developed for the word “thumb” as well. But, this would create complications in future if someone wants to explicitly use the words “spare”, “spear”, and “fear”. To further address this issue, it was found that the Microsoft Bing Speech API uses its own Language Understanding Intelligent Service (LUIS) to identify the intent of the voice command issued [53]. The LUIS helps focus the context of the spoken command, which addresses any words with rhyming nature.

In the case of recognizing the word “sphere”, if a command “Pick up round sphere 7” is issued, LUIS would interpret the voice command correctly. The word “round” helps recognize the word “sphere” since the context of both words belong to a same group (say mathematics). This same approach of using a word with context helped in recognizing word “thumb” correctly. By issuing the voice command as “Open thumb finger 54 percent”, it was observed that the extra word “finger” gave an appropriate context to the sentence, hand helped recognize “thumb” correctly.

The SR system also encountered difficulties while recognizing voice command that was issued with a fast pace. Some incorrect recognition were obtained at some occasions when the voice command was issued with fast pace and inappropriate pace between two different words. Table 4.2 displays the results of incorrect SR due to improper pause between two distinct words. Expected recognition of the voice com-



Voice Recognition

Language

English USA

Time Remaining (Sec)

0

You said

pick up spear 7

Voice Recognition

Language

English USA

Time Remaining (Sec)

0

You said

close tom finger 16%

Figure 4.2. Results of incorrect Recognition of words with rhyming nature.

Table 4.2. Incorrect SR due to improper pace between two distinct words in the voice command

Command	**Recognition**
Grab phone	gramophone
Coke can	call ken/cocaine
Pen cup	pankop

mands were obtained when the voice commands were issued while providing an appropriate pause between two distinct words.

Although the SR system was capable of recognizing words that were used in workspace, the system was evaluated by issuing some ubiquitous words. An evaluation, consisting in 12 ubiquitous words that are used to grasp different objects in daily life, was performed on the system. Eight individuals were asked to issue these commands, 3 times each. This resulted in a set of 24 tests for each word and the accuracy was calculated out of 100 as shown in Figure 4.3.

Table 4.3. SR for ubiquitous words from daily life grasping task

Sr No.	**Words**	**Accuracy (of 100)**
1	apple	100
2	orange	100
3	pen	75
4	cup	96
5	glass	92
6	mobile	100
7	cellphone	100
8	laptop	100
9	notebook	100
10	pencil	100
11	earphones	71
12	keys	96

4.1.2 Speech Recognition Evaluation in Spanish

As mentioned earlier, the SR service from Microsoft provides language support for 29 different languages. To ensure that the developed system in LabVIEW is capable of expanding in different languages, a evaluation was developed with Spanish voice commands. Three individuals with Spanish speaking background were asked to issue this 8 commands, each person twice, as presented in Table 4.4.

The results of SR evaluation in Spanish came out as it was expected with good recognition accuracy. This ensured that this system is capable of incorporating different languages provided by Microsoft Bing Speech. All 3 SR evaluations performed ensured that the developed system in LabVIEW using CAS responds well to the voice commands that could be used in the workspace of controlling the BAH.

Table 4.4. SR evaluation results performed in Spanish

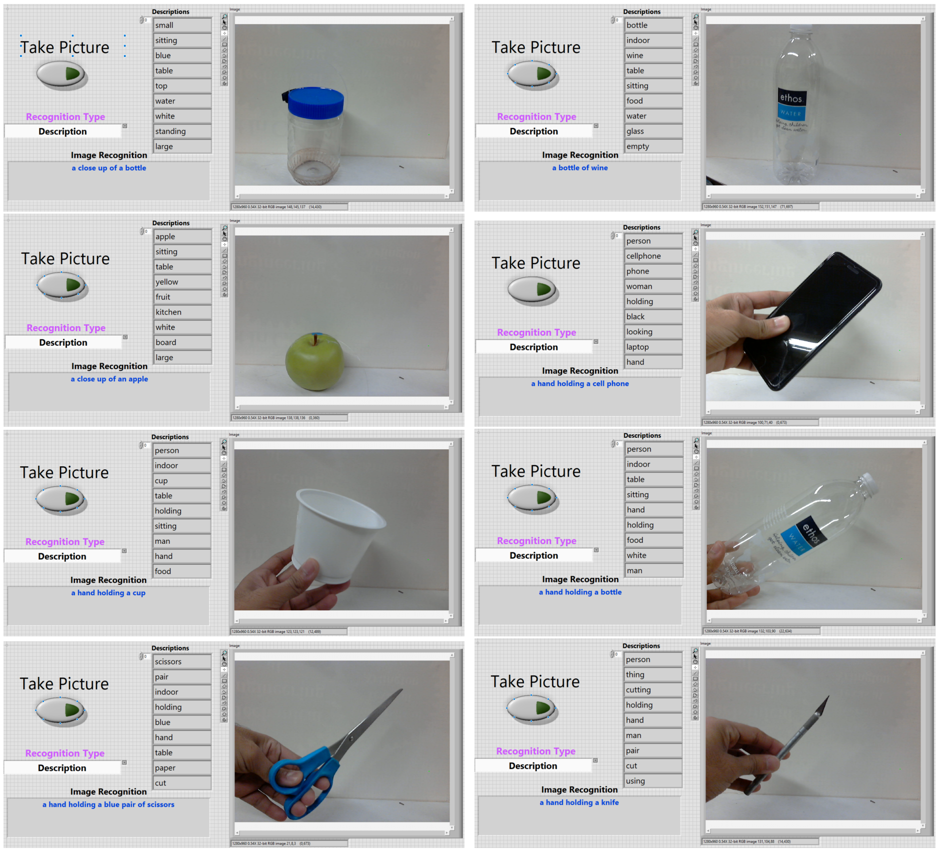
Sr No.	Command	Accuracy (of 100)
1	recoger el cilindro veinte	100
2	lanzamiento del cilindro veinte	100
3	agarrar el cilindro veinte	100
4	colocar el cilindro veinte	100
5	dedo anular abierto veintisiete grados	100
6	cerrar el dedo indice veinticuatro por ciento	100
7	suelta todos los dedos	100
8	recoger la esfera siete	83.3

4.2 Object Recognition Evaluation Results

The object recognition feature was evaluated to verify that the MCV provides information on various objects. In this object recognition evaluation, it was assumed that there is appropriate lighting while taking picture of the workspace and that the camera is focusing on the object only. Some of the object recognitions are shown in Figures 4.3 and 4.4. In all the recognition that the MCV system performed, it recognized the objects in the workspace as shown in Figures 4.3 and 4.4.

4.3 Evaluation of Voice and Vision System in Parallel

A test was also performed to check the credibility of the dual algorithm developed to compare the results of voice and vision system in order to avoid damage because of inappropriate grasping. Figure 4.5 represents the results of the voice command "pick up a water bottle" issued with the intention to grasp an apple in the workspace. The system did not executed the grasping pattern of the bottle to grasp an apple as it was expected. The system which was developed to use voice and vision in parallel with improved safety, responded consistently when more tests were per-



Descriptions: small, sitting, blue, table, top, water, white, standing, large
Image Recognition: a close up of a bottle

Descriptions: bottle, indoor, wine, table, sitting, food, water, glass, empty
Image Recognition: a bottle of wine

Descriptions: apple, sitting, table, yellow, fruit, kitchen, white, board, large
Image Recognition: a close up of an apple

Descriptions: person, cellphone, phone, woman, holding, black, looking, laptop, hand
Image Recognition: a hand holding a cell phone

Descriptions: person, indoor, cup, table, holding, sitting, man, hand, food
Image Recognition: a hand holding a cup

Descriptions: person, indoor, table, sitting, hand, holding, food, white, man
Image Recognition: a hand holding a bottle

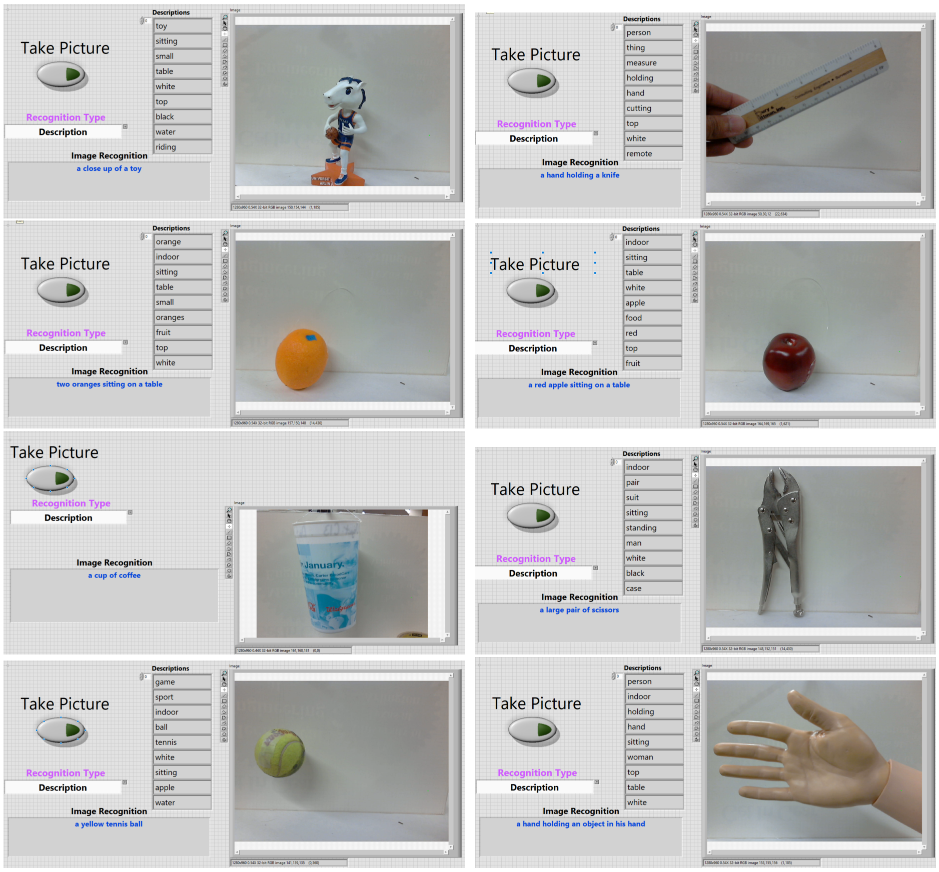
Descriptions: scissors, pair, indoor, holding, blue, hand, table, paper, cut
Image Recognition: a hand holding a blue pair of scissors

Descriptions: person, thing, cutting, holding, hand, man, pair, cut, using
Image Recognition: a hand holding a knife

Figure 4.3. Object Recognition results using objects that grasped daily.

form with different objects such as an orange, apple, and bottle. This system allows the user to expand the database in term of including more objects.

As mentioned earlier, a vision system also serves the purpose of providing object identification if there is no object information present in the voice command. This was also verified by issuing a voice command “pick up an object” as shown in Figure 3.6. Although there is no object information in the voice command “pick up n object”, the algorithm developed then provided the object information and also retrieved the grasping pattern of object. The developed algorithm responded consistently for different objects also such as orange and bottle.



Recognition Type
Description

Descriptions
toy
sitting
small
table
white
top
black
water
riding

Image Recognition
a close up of a toy

Recognition Type
Description

Descriptions
person
thing
measure
holding
hand
cutting
top
white
remote

Image Recognition
a hand holding a knife

Recognition Type
Description

Descriptions
orange
indoor
sitting
table
small
oranges
fruit
top
white

Image Recognition
two oranges sitting on a table

Recognition Type
Description

Descriptions
indoor
sitting
table
white
apple
food
red
top
fruit

Image Recognition
a red apple sitting on a table

Recognition Type
Description

Image Recognition
a cup of coffee

Recognition Type
Description

Descriptions
indoor
pair
suit
sitting
standing
man
white
black
case

Image Recognition
a large pair of scissors

Recognition Type
Description

Descriptions
game
sport
indoor
ball
tennis
white
sitting
apple
water

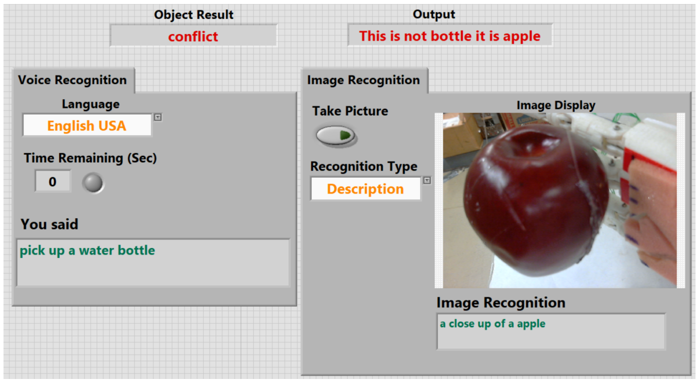
Image Recognition
a yellow tennis ball

Recognition Type
Description

Descriptions
person
indoor
holding
hand
sitting
woman
top
table
white

Image Recognition
a hand holding an object in his hand

Figure 4.4. Object Recognition results using objects that grasped daily.



Object Result
conflict

Output
This is not bottle it is apple

Language
English USA

Time Remaining (Sec)
0

You said
pick up a water bottle

Recognition Type
Description

Image Recognition
a close up of a apple

Figure 4.5. Results of conflict while using voice and vision system in parallel.

CHAPTER 5

Conclusions and Recommendations for Future Research

5.1 Conclusions

A research platform was successfully developed to investigate HRI on a BAH using voice commands. The system developed in LabVIEW using cloud services, currently has the ability to recognize 29 different languages and translate more than 100 different languages. This developed environment in LabVIEW is capable of incorporating different APIs, which makes the system expandable. The goal of this research was to develop a voice control environment with the assistance of vision system for improving safety in HRI.

An additional feature of object recognition is also investigated along with the SR module, in order to provide a safe operating environment. The programming software LabVIEW, handles all the processing including creating an API package, making a HTTP POST call to the API, displaying the results from the JSON file, and performing string (text) search operations. Both speech and object recognition is performed on the Microsoft Azure server.

A string search algorithm was also developed which uses the returned string to identify the desired action to perform, object information and identifier for objects (number, color) in the voice commands after the speech recognition is performed. Two modes of interacting with the BAH were developed, namely issuing voice commands for grasping task and controlling individual finger position. As voice commands for both the modes of interaction differ in terms of sentence structure, so the search algorithm for both modes was developed separately.

The results from speech and object recognition are returned from the Microsoft server in a JSON format. Therefore, a JSON deciphering algorithm was developed to acquire results from the JSON file for further programming. The JSON result includes a collection of name/value pairs and an array of values, so an algorithm was developed so that the user can opt between different recognitions provided by the cloud services.

Grasping of different objects was successfully performed, first using only the voice control system and then with both the voice control and vision system in parallel. A test was perform where the issued voice command conflicted with the workspace environment. In this case, the vision system was capable of returning a warning to the user and more importantly, not following through with the mechanical activation of the fingers of the BAH.

5.2 Recommendations for Future Research

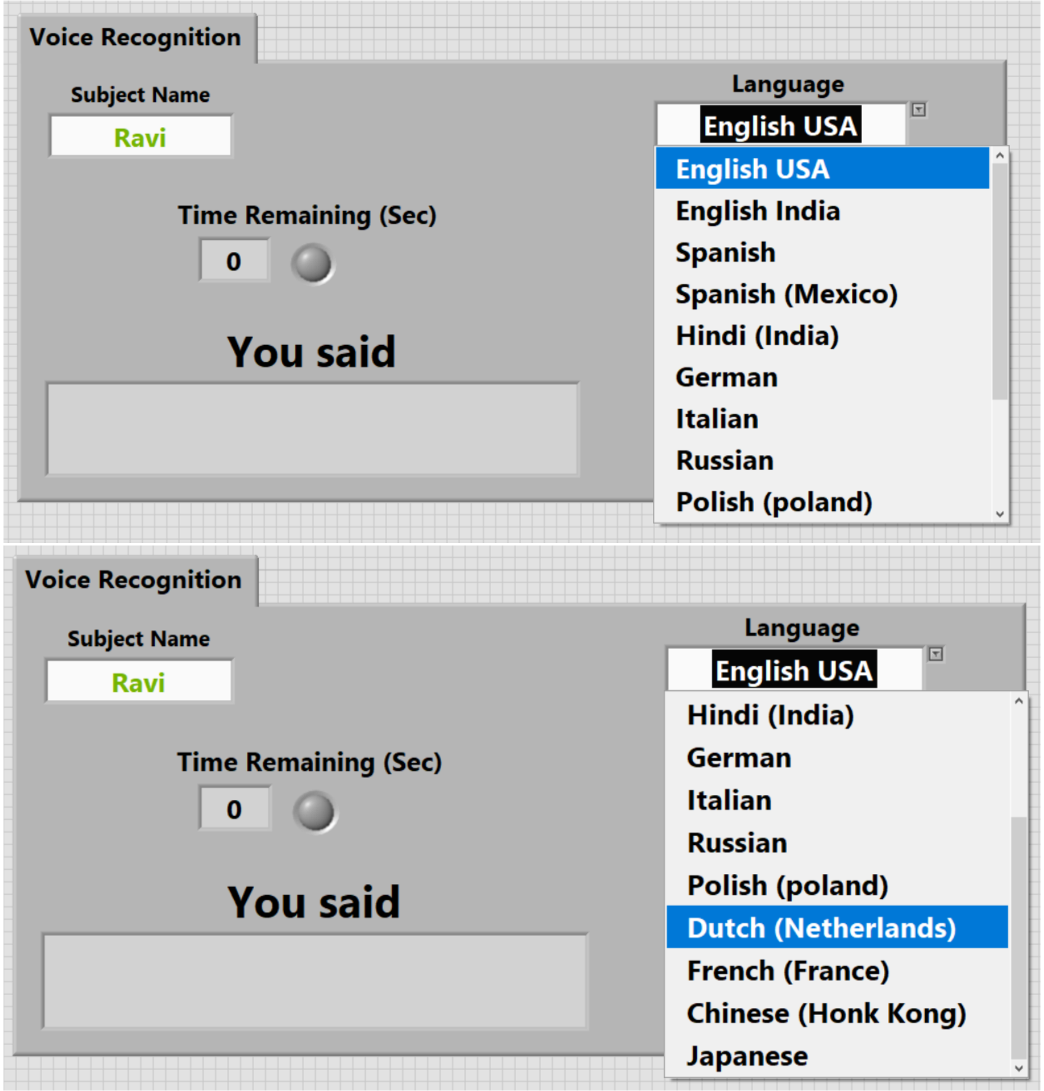
Even though recognizing rhyming words is addressed using an extra set of word to provide context to the voice command, an ANN model specifically for rhyming words is recommended in order to address incorrect recognition. It is also recommended to expand the object database, by including additional object terms instead of the limited database of cylinders and spheres.

For this research, it was assumed that using the BAH for grasping objects would be performed at a uniform relative height to each other. The development of a vision system can provide information on the position of the object with respect to BAH. This can further improve grasping in scenarios when the objects are not at a optimal location to be grasped.

As mentioned earlier that the speech recognition from Microsoft can incorporate 29 different languages, a database in different languages is required to be developed, to dynamically perform a search operation and identify the spoken language. A person with background of different languages could help in incorporating translation of different languages.

A custom vision system based on Microsoft Computer Vision is recommended to be developed. A custom deep learning model based on images of objects can be trained on the Microsoft server in order to acquire higher accuracy in object recognition.

APPENDIX A



Subject Name: Ravi

Language: English USA

Time Remaining (Sec): 0

Figure A.1. GUI for speech recognition using cloud services with different language support.

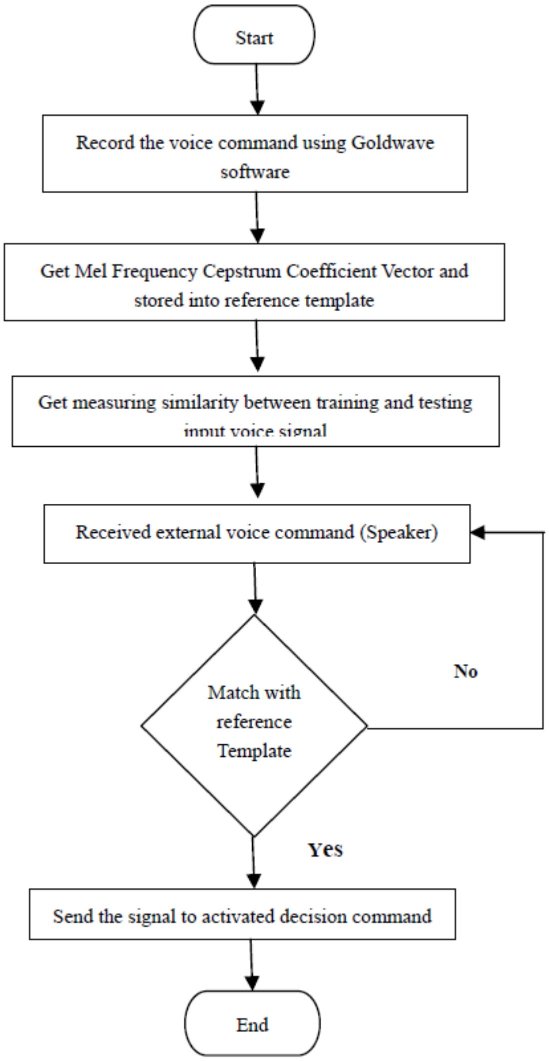

Start
Record the voice command using Goldwave software
Get Mel Frequency Cepstrum Coefficient Vector and stored into reference template
Get measuring similarity between training and testing input voice signal
Received external voice command (Speaker)
Match with reference Template
No
Yes
Send the signal to activated decision command
End

Figure A.2. Voice Algorithm Flow Process [2].

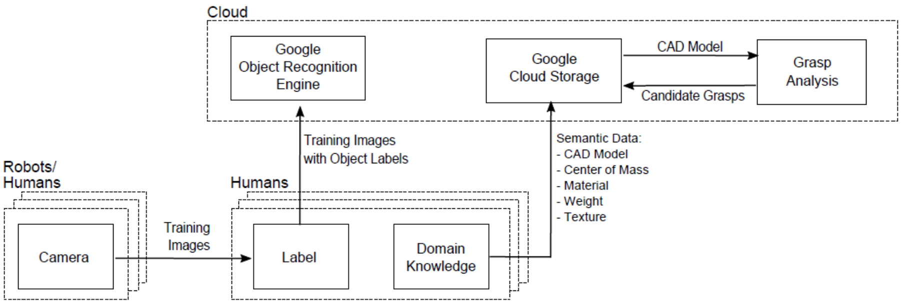


Figure A.3. System Architecture of offline phase proposed in [4].

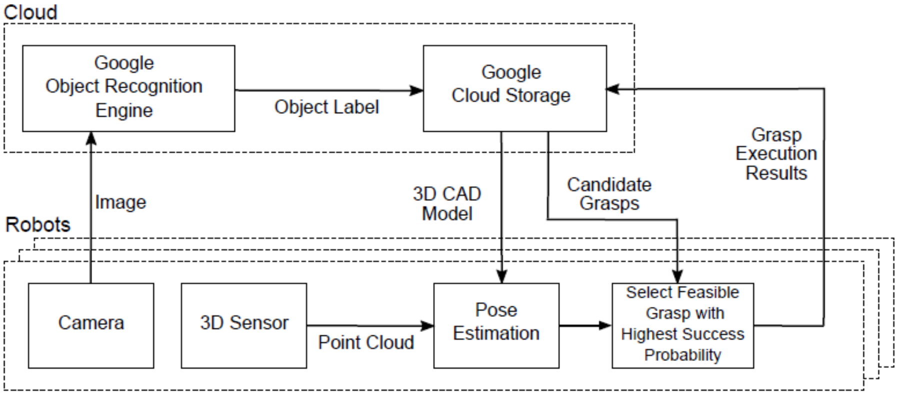


Figure A.4. System Architecture of online phased proposed in [4].

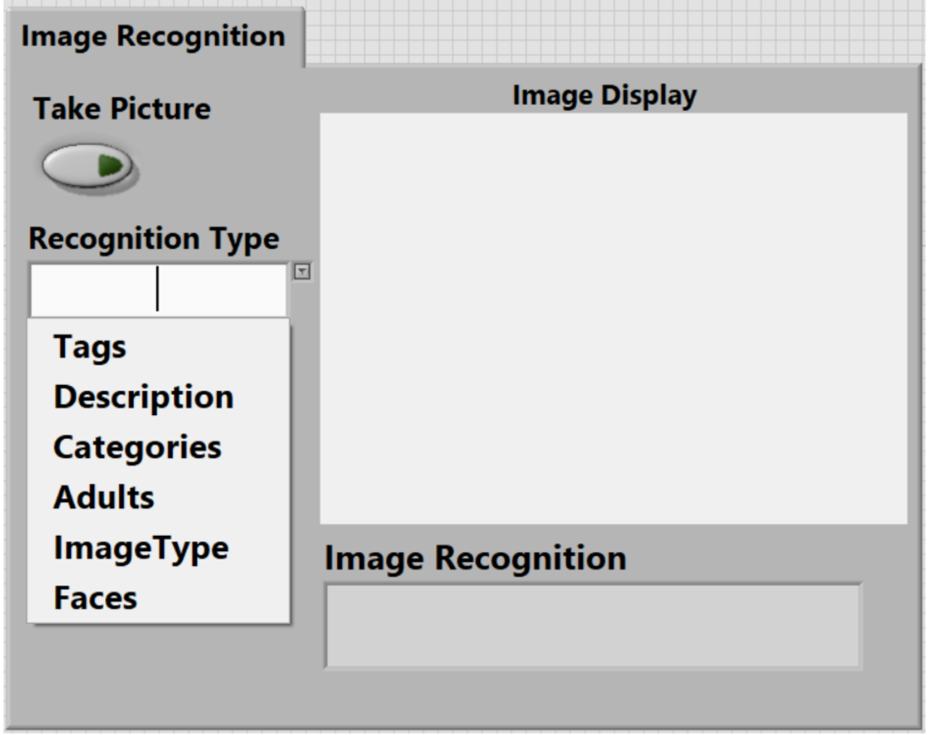


Figure A.5. Graphics interface for object recognition system using Microsoft Computer Vision API.

REFERENCES

[1] J. M. Sackier, C. Wooters, L. Jacobs, A. Halverson, D. Uecker, and Y. Wang, “Voice activation of a surgical robotic assistant,” *The American Journal of Surgery*, vol. 174, no. 4, pp. 406–409, oct 1997.

[2] L. Muda, M. Begam, and I. Elamvazuthi, “Voice recognition algorithms using mel frequency cepstral coefficient (mfcc) and dynamic time warping (dtw) techniques.”

[3] F. Ren, “Robotics cloud and robotics school,” in *Natural Language Processing andKnowledge Engineering (NLP-KE), 2011 7th International Conference on.* IEEE, 2011, pp. 1–8.

[4] B. Kehoe, A. Matsukawa, S. Candido, J. Kuffner, and K. Goldberg, “Cloud-based robot grasping with the google object recognition engine,” in *Robotics and Automation (ICRA), 2013 IEEE International Conference on.* IEEE, 2013, pp. 4263–4270.

[5] K. Gundogdu, S. Bayrakdar, and I. Yucedag, “Developing and modeling of voice control system for prosthetic robot arm in medical systems,” *Journal of King Saud University-Computer and Information Sciences*, 2017.

[6] G. T. Sung and I. S. Gill, “Robotic laparoscopic surgery: a comparison of the da vinci and zeus systems,” *Urology*, vol. 58, no. 6, pp. 893–898, 2001.

[7] S. Maksymova, R. Matarneh, and V. V. Lyashenko, “Software for voice control robot: Example of implementation,” *Open Access Library Journal*, vol. 4, no. 08, p. 1, 2017.

[8] P. Weber, E. Rueckert, R. Calandra, J. Peters, and P. Beckerle, “A low-cost sensor glove with vibrotactile feedback and multiple finger joint and hand motion sensing for human-robot interaction,” in *Robot and Human Interactive Communication (RO-MAN), 2016 25th IEEE International Symposium on.* IEEE, 2016, pp. 99–104.

[9] A. Mishra, P. Makula, A. Kumar, K. Karan, and V. K. Mittal, “A voice-controlled personal assistant robot,” in *2015 International Conference on Industrial Instrumentation and Control (ICIC).* IEEE, may 2015.

[10] C. M. Oppus, J. R. R. Prado, J. C. Escobar, J. A. G. Marinas, and R. S. Reyes, “Brain-computer interface and voice-controlled 3d printed prosthetic hand,” in *2016 IEEE Region 10 Conference (TENCON).* IEEE, nov 2016.

[11] K. Kumar, S. Nandan, A. Mishra, K. Kumar, and V. K. Mittal, “Voice-controlled object tracking smart robot,” in *2015 International Conference on Signal Processing, Computing and Control (ISPCC).* IEEE, sep 2015.

[12] R. Robins and M. Stonehill, *Investigating the NeuroSky MindWave^{TM} EEG Headset*, 3 December 2014.

[13] M. A. Goodrich and A. C. Schultz, “Human-robot interaction: A survey,” *Foundations and Trends® in Human-Computer Interaction*, vol. 1, no. 3, pp. 203–275, 2007.

[14] R. Paul, A. Barbu, S. Felshin, B. Katz, and N. Roy, “Temporal grounding graphs for language understanding with accrued visual-linguistic context,” in *Proceedings of the 26th International Joint Conference on Artificial Intelligence.* AAAI Press, 2017, pp. 4506–4514.

[15] G. Baron, P. Czekalski, M. Golenia, and K. Tokarz, “Gesture and voice driven mobile tribot robot using kinect sensor,” in *2013 International Symposium on Electrodynamic and Mechatronic Systems (SELM).* IEEE, may 2013.

[16] S. Nakagawara, H. Kajimoto, N. Kawakami, S. Tachi, and I. Kawabuchi, "An encounter-type multi-fingered master hand using circuitous joints," in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*. IEEE, 2005, pp. 2667–2672.

[17] T. B. Martin, R. O. Ambrose, M. A. Diftler, R. Platt, and M. Butzer, "Tactile gloves for autonomous grasping with the nasa/darpa robonaut," in *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, vol. 2. IEEE, 2004, pp. 1713–1718.

[18] A. C. Tenorio-Gonzalez, E. F. Morales, and L. Villaseñor-Pineda, "Dynamic reward shaping: training a robot by voice," in *Ibero-American Conference on Artificial Intelligence*. Springer, 2010, pp. 483–492.

[19] G. Langevin, "Inmoov open source 3d printed life-size robot."

[20] U. Shah, "Human robot interaction - knowledge based approach," Master's thesis, University of Texas at Arlington, 2016.

[21] C. E. Ábrego, P. S. Shiakolas, and M. R. Sobhy, "Developing an educational and research human robot interaction environment for a mechanical finger/hand," in *ASME 2015 International Mechanical Engineering Congress and Exposition*. American Society of Mechanical Engineers, 2015, pp. V04AT04A023–V04AT04A023.

[22] "Labview," *National Instrument*, 2017. [Online]. Available: www.ni.com/en-us/shop/labview.html

[23] "myrio," *National Instrument*, 2017. [Online]. Available: www.ni.com/myrio

[24] N. Manish and B. J. Reddy, "Review of voice control robot applications," *Int. J. Adv. Eng*, vol. 1, no. 9, pp. 671–674, 2015.

[25] A. Poncela and L. Gallardo-Estrella, "Command-based voice teleoperation of a mobile robot via a human-robot interface," *Robotica*, vol. 33, no. 1, pp. 1–18, 2015.

[26] N. Desai, K. Dhameliya, and V. Desai, "Recognizing voice commands for robot using mfcc and dtw," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 3, no. 5, 2014.

[27] X. Lv, M. Zhang, and H. Li, "Robot control based on voice command," in *2008 IEEE International Conference on Automation and Logistics.* IEEE, sep 2008.

[28] B. Kehoe, S. Patil, P. Abbeel, and K. Goldberg, "A survey of research on cloud robotics and automation," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 2, pp. 398–409, apr 2015.

[29] J. Wan, S. Tang, H. Yan, D. Li, S. Wang, and A. V. Vasilakos, "Cloud robotics: Current status and open issues," *IEEE Access*, vol. 4, pp. 2797–2807, 2016.

[30] U. o. U. Chris Coleman, School of Computing, "Enabling a new future for cloud computing." [Online]. Available: https://www.nsf.gov/news/news_summ.jsp?cntn_id=132377&org=NSF

[31] A. M. Saumyo Ghosh, "Elevation in robotics and automation using level structured cloud cadre," *International Journal of Innovative Science, Engineering & Technology, Vol. 2 Issue 9, September 2015*, 2015.

[32] K. Goldberg *et al.*, "Beyond the web: Excavating the real world via mosaic," in *in Second International WWW Conference.* Citeseer, 1994.

[33] K. Goldberg, M. Mascha, S. Gentner, N. Rothenberg, C. Sutter, and J. Wiegley, "Desktop teleoperation via the world wide web," in *Robotics and Automation, 1995. Proceedings., 1995 IEEE International Conference on*, vol. 1. IEEE, 1995, pp. 654–659.

[34] D. Berenson, P. Abbeel, and K. Goldberg, “A robot path planning framework that learns from experience,” in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 3671–3678.

[35] M. Ciocarlie, C. Pantofaru, K. Hsiao, G. Bradski, P. Brook, and E. Dreyfuss, “A side of data with my robot,” *IEEE Robotics & Automation Magazine*, vol. 18, no. 2, pp. 44–57, 2011.

[36] R. Arumugam, V. R. Enti, L. Bingbing, W. Xiaojun, K. Baskaran, F. F. Kong, A. S. Kumar, K. D. Meng, and G. W. Kit, “DAvinCi: A cloud computing framework for service robots,” in *2010 IEEE International Conference on Robotics and Automation*. IEEE, may 2010.

[37] “Ten dollar robot,” *The African Robotics Network (AFRON)*. [Online]. Available: http://robotics-africa.org/2012-design-challenge.html

[38] M. Waibel, M. Beetz, J. Civera, R. d’Andrea, J. Elfring, D. Galvez-Lopez, K. Häussermann, R. Janssen, J. Montiel, A. Perzylo *et al.*, “Roboearth,” *IEEE Robotics & Automation Magazine*, vol. 18, no. 2, pp. 69–82, 2011.

[39] “Microsoft azure.” [Online]. Available: https://azure.microsoft.com/

[40] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[41] G. Costache, P. Corcoran, R. Mulryan, and E. Steinberg, “Method and component for image recognition,” May 11 2010, uS Patent 7,715,597.

[42] “Amazon elastic cloud.” [Online]. Available: https://aws.amazon.com/ec2/

[43] “Google compute engine.” [Online]. Available: https://cloud.google.com/compute/

[44] “Ibm watson.” [Online]. Available: https://www.ibm.com/watson/

[45] "Get started with speech recognition by using the rest api." [Online]. Available: https://docs.microsoft.com/en-us/azure/cognitive-services/speech/getstarted/getstartedrest?tabs=Powershell

[46] "Labpython." [Online]. Available: http://labpython.sourceforge.net

[47] "Post (http)," *Wikipedia*. [Online]. Available: https://en.wikipedia.org/wiki/POST_(HTTP)

[48] "Supported languages for speech," 2017. [Online]. Available: https://docs.microsoft.com/en-us/azure/cognitive-services/speech/api-reference-rest/supportedlanguages

[49] "Basic concepts understanding speech recognition." [Online]. Available: https://docs.microsoft.com/en-us/azure/cognitive-services/speech/concepts

[50] "Google cloud translate api." [Online]. Available: https://cloud.google.com/translate/

[51] "Computer vision api," 2017. [Online]. Available: https://docs.microsoft.com/en-us/azure/cognitive-services/computer-vision/home#a-nametaggingtagging-imagesa

[52] "Vision and motion," *National Instrument LabVIEW*, 2016. [Online]. Available: http://zone.ni.com/reference/en-XX/help/370281AC-01/imaqvision/vision_and_motion_pal/

[53] "Language understanding (luis)," 2017. [Online]. Available: https://docs.microsoft.com/en-us/azure/cognitive-services/luis/home

BIOGRAPHICAL STATEMENT

Ravi Patel was born in Gujarat, India in 1993. He received his Bachelor of Engineering in Mechanical Engineering from Sal Institute of Technology and Engineering Research, Gujarat, India in 2014. He then joined the University of Texas at Arlington to pursue his Master of Science degree in Mechanical Engineering in Summer 2016. While at UTA, Ravi worked as a Graduate Teaching Assistant for sophomore level course Experimental Methods and Measurements. His research interest includes robotics, automation, and control systems and hopes to work in a related field after receiving his Master of Science in Mechanical Engineering Degree.