Universität Stuttgart

Institute of Parallel and Distributed Systems

Machine learning and robotics department

University of Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Masterarbeit

Speech interface for human and robot collaboration

Moin Uddin Kashif

Course of Study:	Master of Science Information Technology
Examiner:	Prof. Dr. rer. nat. Marc Toussaint
Supervisor:	Ph.D. Ruth Schulz
Commenced:	February 28, 2018
Completed:	August 28, 2018

Abstract

In the past, robots and machines were mostly designed to perform specific tasks without much human interaction needed. Nowadays with the advancements in technology, intelligent robots can be designed which can perform multiple tasks, interact with the surrounding environment, assist and give valuable suggestions to humans etc. so an efficient and natural mode of communication is required for this human-robot interaction. In this thesis, we proposed an architecture to develop a speech interface for human-robot interaction. The speech interface is used to give voice commands to the robot, PR2, in order to perform 5 tasks which are designed to test the performance of the speech interface. The tasks are sorting, shaping, stacking, building and balancing of 6 objects on table-top which are designed and ordered by the level of difficulty. First two tasks are comparatively easier as the user doesn't have to follow any order to finish them, next two tasks require to follow the order and in the last task, the stack of objects must be balanced in order to finish it. The speech interface receives voice commands from the user, convert them into text, maps to the corresponding command and send to the task manager to perform the operation. After that, it processes the received command, takes the appropriate decision based on the current status of the task and available actions and sends the command to the PR2 to perform the operation. Additionally, we have designed a feedback mechanism where PR2 sends back the feedback to the task manager which is delivered back to the speech manager so that it can be converted into an audio signal and play for the user. Furthermore, the system uses a TCP connection for the exchange of data and information between the speech manager and the task manager. The speech interface is also compared with other modalities such as text input and graphical user interface with the same tasks and we have also conducted user study to evaluate the system performance. The results show that the participants prefer speech interface as it feels more natural.

Kurzfassung

In der Vergangenheit waren Roboter und Maschinen meist so konzipiert, dass sie bestimmte Aufgaben ohne viel menschliche Interaktion ausführen konnten. Mit den Fortschritten in der Technologie können intelligente Roboter entwickelt werden, die mehrere Aufgaben erfüllen, mit der Umgebung interagieren, dem Menschen wertvolle Anregungen geben, so dass eine effiziente und natürliche Art der Kommunikation für diese Mensch-Roboter-Interaktion erforderlich ist. In dieser Arbeit haben wir eine Architektur zur Entwicklung einer Sprachschnittstelle für die Mensch-Roboter-Interaktion vorgeschlagen. Die Sprachschnittstelle wird verwendet, um dem Roboter PR2 Sprachbefehle zu erteilen, um 5 Aufgaben auszuführen, die dazu bestimmt sind, die Leistung der Sprachschnittstelle zu testen. Die Aufgaben sind Sortieren, Formen, Stapeln, Bauen und Ausbalancieren von 6 Objekten auf Tischplatten, die nach Schwierigkeitsgrad gestaltet und geordnet sind. Die ersten zwei Aufgaben sind vergleichsweise einfacher, da der Benutzer keiner Reihenfolge folgen muss, um sie zu beenden. Bei den nächsten zwei Aufgaben muss die Reihenfolge berücksichtigt werden. In der letzten Aufgabe muss der Objektstapel ausbalanciert werden, um sie auszuführen. Die Sprachschnittstelle empfängt Sprachbefehle vom Benutzer, wandelt sie in Text um, bildet den entsprechenden Befehl ab und sendet sie an den Aufgabenmanager, um die Operation auszuführen. Danach verarbeitet es den empfangenen Befehl, trifft die geeignete Entscheidung basierend auf dem aktuellen Status der Aufgabe und verfügbarer Aktionen und sendet den Befehl an den PR2, um die Operation auszuführen. `Zusätzlich haben wir einen Rückkopplungsmechanismus entwickelt, bei dem PR2 die Rückmeldung an den Aufgabenmanager zurücksendet, der diese an den Sprachmanager zurückgibt, so dass er es in ein Audiosignal umwandeln und für den Benutzer spielen kann. Darüber hinaus verwendet das System eine TCP-Verbindung für den Austausch von Daten und Informationen zwischen dem Sprachmanager und dem Aufgabenmanager. Die Sprachschnittstelle wurde auch mit anderen Eingabemöglichkeiten wie Texteingabe und grafischer Benutzeroberfläche für die gleichen Aufgabenstellungen verglichen. Durch Benutzerstudien wurde die Systemleistung bewertet. Die Ergebnisse zeigen, dass die Teilnehmer die Sprachschnittstelle bevorzugen, da sie intuitiver ist.

Table of content

List of figures

List of equations

List of Algorithms

List of tables

1 Introduction

This chapter consists of three main sections. The first section describes the nature, drawbacks and severity of the existing problem which has not been addressed yet. The second section describes the motivation of this thesis, our approach towards solving this problem and the solution. Finally, the last section describes the structure of this thesis.

1.1 Problem statement

In the past, machines and robots were mostly designed for industries to perform just specific tasks without much human interaction. The human interaction was also limited in order to avoid accidents and injuries [YKI+08]. A machine operator was required for switching on/off the machine and to provide the required resources to the machine. Those machines were just usually equipped with accessories required to complete the given task, therefore, human supervision was necessary in case something went wrong.

Nowadays the concept of machines and robots have changed significantly and it has become possible to create such robots which are able to perceive their surroundings with the help of various sensors and perform multiple tasks efficiently and accurately [Kha98]. With the help of these sensors and other safety measures, the risk of human-robot interaction is also reduced significantly. Despite the fact that, robots nowadays are intelligent enough to perform tasks efficiently and are able to give suggestions to humans inappropriate manner, as discussed in [TFK13] and [LKF+10], but still they need guidance in various situations and scenarios. For example a driverless vehicle can perceive the grass as an obstacle in front of it but humans can decide better that it is just grass and is safe to go through.

There are several ways of interacting with the robots like touch screens, gestures, commands from input devices etc. [Kha98] and other studies have done a lot in this area in order to improve HRI methods but still, there is a lot of room for improvement. It is highly dependent on the situation and the user, how they like to interact with the robot while keeping the error rate as low as possible.

The human-robot collaboration turns out to be more efficient in performing tasks than performing task individually in some situations. For example [FCT+01] describes how human-robot collaborative work is beneficial in planetary missions where planetary rovers work together with humans in order to increase mission productivity by helping them with tasks such as material transportation, survey, sampling and on-site characterization. Robots are good in structured planning for which well-defined algorithm exist and they just have to follow them [FTT+03], however, unstructured planning is something in which robots are not that good especially when common sense is required to make the decision [Cla94].

As machines or robots are integrating more and more in our daily lives so it is very important to find a proper and efficient way to interact with them which gives a feeling that we are interacting with humans instead of robots. Humans are social and experts in interacting with each other, they can express and respond to feelings which are not present in robots, according to [RN96] human find the interaction with the robots more enjoyable and meaningful if they are close to human nature and social expectations. One of the most natural modes of communication with robots is understanding and communicating with them in natural language [KGH+03], but it is not easy as the main problem is the mapping and translation of the natural language to its corresponding meaning according to the situation [FCT+01].

It is not enough to just provide commands to robots to perform the specific task but a proper communication between them is really important to eliminate ambiguities [AKV+09] and to ask for proper guidance or further clarifications of the commands [DM02]. The robot needs to be able to understand the full context of the command, [Gor01] describes intelligent user interface model which is helpful in designing and creating the intelligent system which focuses on the dialogue between the human and machine as well as their interaction

with each other. It is also very important that the robot must provide feedback to the user. For example if it is not possible to complete the task because objects are not reachable, then the robot should inform the user.

1.2 Goal

There are various approaches to resolve ambiguities from the spoken sentences. One approach is to add additional nonverbal information with the verbal command such as pointing towards the object on which operation needs to perform as described in [TNK+98] where the robot listens for the command and observes the motion to determine the object or location. If the object is very far or there are many obstacles in between then this approach might not work. Another approach is to have a library of dialogue strategies proposed in [AKH+99].

Therefore this thesis aims to identify suitable algorithms which will be helpful in developing a speech interface which uses appropriate methods to eliminate ambiguities from the spoken sentences and develop a proper feedback mechanism of the robot. In this thesis, we describe the design and implementation of a speech interface which is more generic and can be distinguished from previous approaches. We focus on keeping the dialogue system multi-lingual so it can be used by various users, secondly spoken sentences are not required to be in a specific format. The integrated speech feedback algorithm will keep the user updated about the current status.

1.3 Outline

The rest of the master thesis document is structured in the following manner.

Background and related work: First of all, we discuss the basic theoretical background required to understand the work done in this thesis which includes the knowledge of natural language processing, event handling, context analyses and feedback algorithms. Then we will discuss the relevant contributions made by others in this field and how their research is utilized in our work.

System design: This section focuses on the setup of the system necessary to achieve the objective of this thesis. This includes setting up libraries required for natural language processing, software to execute the code and setting up the robot to execute given commands in order to perform the desired tasks.

Tasks: This section explains the tasks designed to evaluate the system. The tasks are categorized and ordered based on their difficulty levels and how much user attention is required to complete each task.

Implementation: This section of the thesis focuses on the design and implementation of the algorithms.

User study: This section presents the results of the study done with real users. The users are selected from different regions and background to evaluate the system performance. The study is done in the lab of machine learning department and results are shown graphically along with the descriptive explanation.

Discussion: Possible improvements, and expansion of this work are proposed in this section.

Conclusion: This section focuses on the summary of the proposed technique and its performance based on the achieved results.

2 Background and related work

This chapter consists of two main sections. The first section is theoretical background which describes the basic knowledge required for this thesis and the second section is related work which describes the previous studies and research did in this area and their relationship with our work.

2.1 Theoretical background

In this chapter, we will discuss the basic concepts and knowledge required to better understand this work. This section will also describe related terms and techniques in detail such as natural language processing, speech ambiguity, semantic analysis, syntactic rules, a speech recognizer, speech synthesizer, a task descriptive language, turn-taking and designing of a robot. Furthermore, we analyze and explain the existing related work in this field then a discussion is made about how this work is an extension of the previous research and adding value to existing work.

2.1.1 Natural language processing

In order to enable the humans to communicate with the robot in the natural language, the robot should understand that and respond accordingly which is only possible through natural language processing. This technique mainly includes speech recognition, elimination of ambiguity and synthesizer. In natural language, there are some rules and principles to follow in order to form the structure of the sentence. These set of rules determine the order of the words in the sentence, proper use of punctuation marks, proper use of tenses etc. and are known as syntax. Any sentence considered as a correct sentence if it obeys these principles. The most basic feature of the syntax of the language is the usage and appearance of the subject, verb and object in the sentences. These rules states which parts must be present in the sentence because one sentence can be said in many ways.

The process of finding out part of speech for each word is referred to as part-of-speech tagging, which is different for each language. For example in the English language, the same word can be a noun ("the book of the student") or a verb ("please book a room"). Some languages have more ambiguities than others which makes the processing of natural language and the mapping of related meaning that machines can understand, more difficult.

Another important factor is the construction of the parse tree of a given sentence. A single sentence can be said in multiple ways so the parse tree can also be created in multiple ways as well. The parsing can be divided into two types Dependency parsing and Constituency parsing. Dependency parsing is referred to as the connection of words according to their relationship. Each node in the tree is a word and child nodes are dependent on their parent node, while edges show their relationship. The figure 2-1 represents a simple dependency parse tree whereas the constituency parser breaks the text into sub-phrases until all the words are represented separately in the tree which can be seen in the figure 2-2.

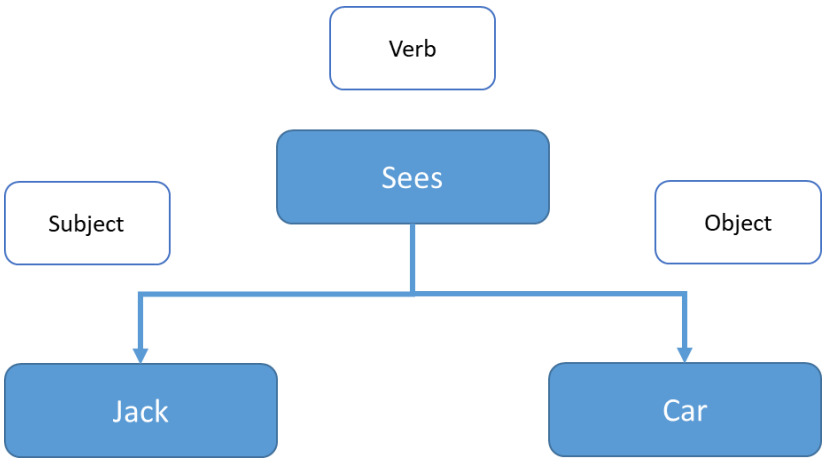

Figure 2-1: Dependency parse tree based on figure 4 [MMM+06]
The child nodes are dependent on the parent whereas the edges represent their relationship. Each node represents the word in the sentence.

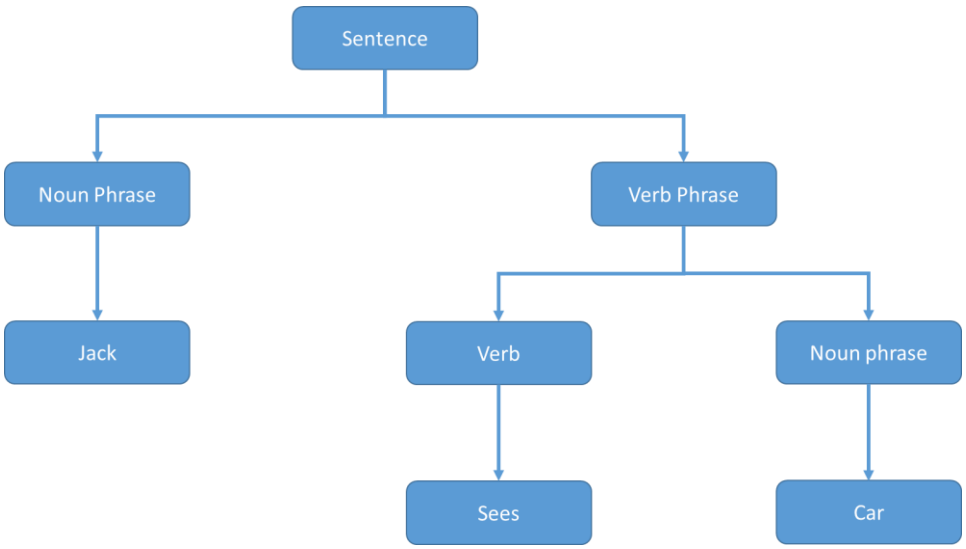

Figure 2-2: Constituency parse tree based on figure 1 [JRN07].
The end nodes represent the words in the sentence

2.1.1.1 Constituency and Dependency parsing

The performance and the accuracy of the parsing technique, which focuses on the sub-phrases in order to create the parse tree, can be improved with the help of greedy transition systems which is further described here [CJL16]. Similarly, there exist several approaches for dependency parsing like [SKT07] which uses a data-driven variant of the LR algorithm for dependency parsing while extending it with a best-first search for probabilistic generalized LR dependency parsing.

Their work extends the previous work mainly in four ways. First, they used the LR parsing algorithm [KD65]. Second, they used the best-first search strategy to generalize the standard deterministic stepwise framework to probabilistic parsing. Third, in order to improve the accuracy of the parser, the approach of the parser ensemble can be used and lastly they represent a method for parser domain adaption with the help of unlabeled data. Another approach is presented in [ADS+07] which uses DeSR for multilingual dependency parsing and domain adaption.

A process of separating the continuous text into separate words is known as word segmentation. It is easier for the language if it has defined the syntax of the words in the sentence. For example in the English language the words are usually separated by a blank space in between them. The term named entity recognition is

defined as the identification of proper names in the sentences like people and places along with their type such as a person, location etc. The syntax of some languages provides some aid to identify them. For example in English language capitalization of words give a hint that the word may be a name, location or organization but it is not always true as the first letter of the sentence is also capital.

In this thesis, the system first listens to the command and then parse it to find the relevant information such as verb and object. A verb is an action which robot have to perform like a pickup or place the object. We used 6 metal objects which identified by their sizes and colors. If the system found the required information in the sentence then it is processed and delivers to the task manager to perform the operation otherwise the speech manager asks the user to provide the missing information. Another very important factor in natural language processing is the ambiguity in the sentence, as many words have more than one meaning. Humans understand the meaning of the text by considering many other factors based on their experience which is not the case for the machine, so it is required to select the meaning which makes the most sense in context.

2.1.2 Speech ambiguity

Ambiguity is defined as the level of uncertainty in a sentence, in case of speech, as one sentence can be interpreted in many different ways. The context plays an important role in resolving the ambiguity. There are other approaches as well such as integration of non-verbal actions, initiating query etc. which are helpful in resolving ambiguities. A similar approach to resolve ambiguities is proposed in the paper [TNK+98] with the help of multimodal human-robot interface which consists of verbal and nonverbal communication. Researchers also proposed that the use of nonverbal commands would also be useful in order to improve the performance of the human-robot interaction [BKT+05] but the functionality of the robot would be a debatable topic. According to them, natural language always contains ambiguity in instructions, so non-verbal instructions improve performance instead of additional processing needed to resolve these ambiguities. The process of mapping the speech to its corresponding meaning is known as semantic analysis. In this thesis, we used a dialogue management system [AKV+09] which ask the missing information to the user to eliminate the ambiguities from the voice commands.

2.1.3 Semantic analysis

It is referred to as the process of translating and mapping the text or speech to its relevant meaning independent of language. In natural language processing, it is referred to as the understanding of the meaning of the text. During communication sometimes it becomes hard to understand what is being said, what is the actual meaning of that sentence or word? Humans understand the sentences not just by listening but also by analyzing different factors such as the tone of the speaker, facial expression of the speaker, loudness of the voice, previous knowledge about the speaker and so on, but most machines do not rely on these same techniques.

Semantic analysis processes the text and sentences based on their structure in order to identify the most relevant topic discussed in the text. It helps the system to understand that the text is about the specific topic even when that word is not present in the text at all. Junqua et al. [JC06] describes the process of customizing the speaking style of a speech synthesizer with the help of semantic analysis. The method consists of several factors like receiving the input text, determining the semantic information by semantic analysis, determining the speaking style based on semantic information and adjusting the speech synthesizer according to the identified speaking style in order to maintain the similarity between the input and the output.

2.1.4 Speech Recognizer

Nowadays there are several ways to interact with the machines in order to give commands and instructions to the machines such as display screens, buttons, audio input etc. One of the most natural modes of communication with robots is communicating with them in natural language [KGH+03], which require the mapping and translation of natural language to its corresponding meaning.

Speech recognizer plays an important role in converting the speech signals into corresponding text symbols. This technology is known as Automatic speech recognition (ASR) or Speech to text (STT). Some systems require training to be able to recognize speech input efficiently. This training is done in a manner that a speaker reads the text for the system while the system is listening, then the system analyzes the reader's specific voice to train itself and use it to fine tune the recognition of that person's speech. The recursive training results in better accuracy and performance.

There are many factors that need to be considered in the speech recognition process but the most important one is noise. Noise can be of various types like environmental, reader's condition like stress, system fault, microphone mismatch, signal processing etc. John H.L. Hansen discussed these two noisy conditions in his research paper [HJ95], stress and environmental noise, and proposed methods to address this issue in order to improve the speech recognition process.

The stressed condition of the speaker, due to workload or sadness or fight, is referred to as Lombard effect [JC93] where the production of the speech is affected and the speaker struggles to speak clearly. This results in additive noise in the recognition process. According to [HJ95], the better recognition process consists of three major parts, better training methods, advanced front-end processors and improved back-end processing algorithms. The figure 2-3 shows the working of the speech recognizer. As the natural language contains huge and redundant data, it needs to be reduced to a smaller subset which represents the whole data, this process is known as feature extraction. The relationship of the phonemes with an audio signal is represented by acoustic models. The decoder used this reduced data together with acoustic and language models to generate related text. He also proposed three approaches to achieve improvements in speech recognition, stress equalization and noise suppression, feature enhancing artificial neural network (FEANN) and morphological constrained feature enhancement.

The speech recognition systems are divided into two broad categories "speaker dependent" and "speaker independent". Speaker dependent systems are those which required training and the systems which do not require training are referred to as speaker independent. There are already several studies on these topics in order to improve the system, for example, [FS91] discussed processing techniques for speaker dependent speech recognizers.

Doddington et al. [DGE90] proposed that the speech recognition process can be improved by comparing frame-pair feature vectors which reduce the variations of the context in the pronunciation of words. Another speaker adaption method is proposed in the study [ZT94] by Zhao et al., in order to improve speaker independent speech recognition with the help of decomposition of spectral variation source which is divided into two categories, acoustic and phone-specific. Initially, the system performs acoustic normalization and then phone model parameters are adapted on the result. The system uses Gaussian mixture density based hidden Markov model which shows a significant improvement in speech recognition from 80.9% to 90.5% as compared to other basic systems with the error reduction rate of 27.5%.

The speech manager, in this thesis, listens to the voice command and uses google speech to text converter for the conversion. The converted text mapped to its corresponding value before sending to the task manager, then the task manager processes the received value and converts it into the command which robot can understand. After that, the robot executes it and send back the feedback to the task manager which delivers it to the speech manager. This text is converted back into an audio signal with the help of aspeech synthesizer so the user can listen to it.

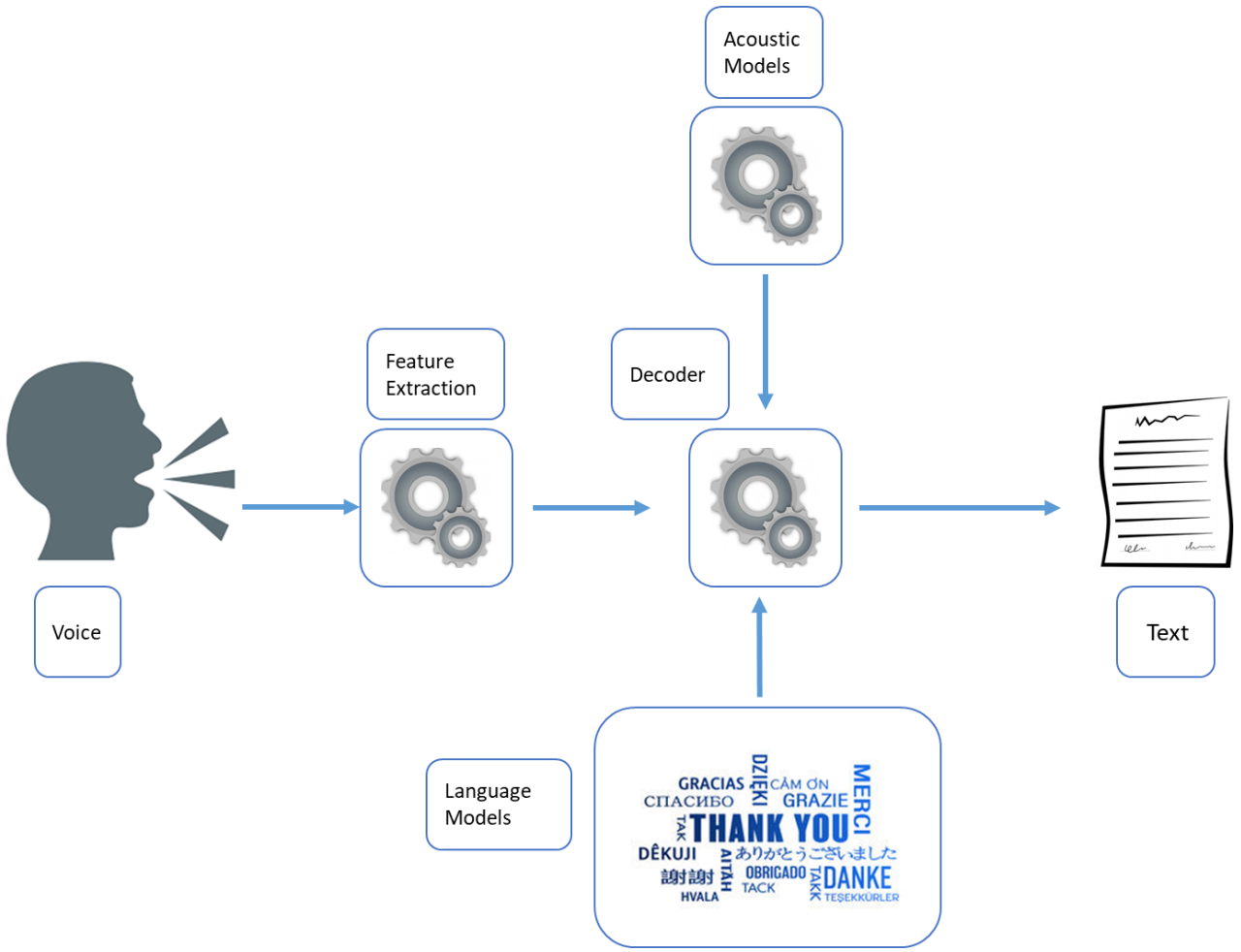

Figure 2-3: Speech recognizer based on figure 1 [BAK+01].
The feature extractor delivers the subset of speech, which represents whole data, to the decoder which converts speech into text, based on the acoustic and languages models.

2.1.5 Speech synthesizer

In any mode of communication, one of the most important factors is providing feedback so that the sender knows that the command is delivered or not. In human-robot interaction, when a person gives the instruction to the robot then some kind of feedback is required to inform the user that command is received. There are several ways to provide the feedback, like visual instructions on the displays, tactile feedback through vibrations, audio signals etc.

In order to make the audio feedback more natural and valuable, a technology is used named the speech synthesizer. Speech synthesizing is defined as the process of creating human voice artificially and the computer or system used for this purpose is known as the speech synthesizer. Artificial human speech is created by concatenating separate pieces of recorded voice.

A text to speech conversion consists of two main parts, a front-end, and a back-end. The front-end mainly focuses on converting the symbols such as numbers, abbreviations, and characters into their equivalent words known as text normalization. Then an appropriate visual representation of speech sound is assigned to each word and after that, these are divided into the meaningful chunks like phrases and sentences. The back-end then matches those visual sound symbols to related sound and then deliver the whole package to the output system [CNA02], the figure 2-4 represents this technique graphically.

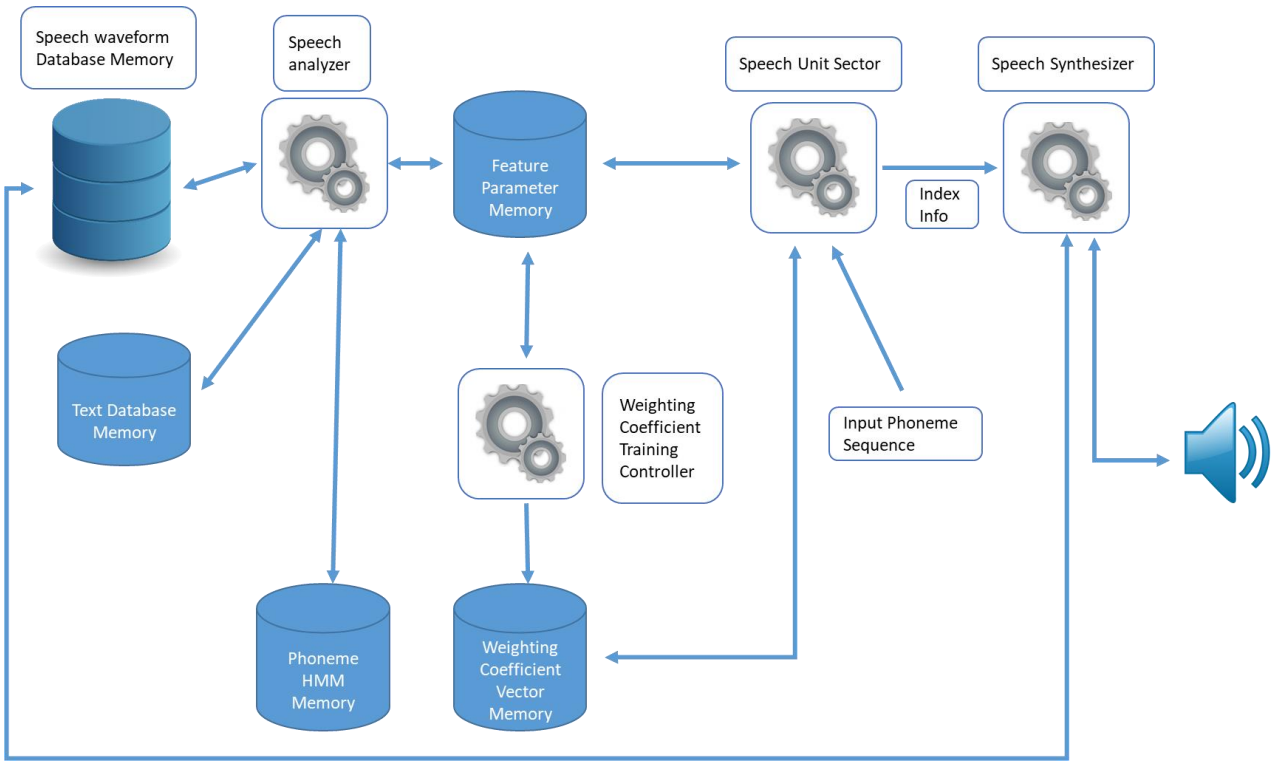

Figure 2-4: Speech synthesizer based on figure 1 [CNA02].
The speech analyzer, analyze the speech waveform stored in the database and generates phonemic symbol sequence. It also extracts the acoustic features which store temporarily in feature parameter memory. After deciding optimal weighting coefficient by the training process, speech unit selector adds index information to the speech segments which used by speech synthesizer to generate speech waveform by concatenating received speech segments based on the index.

A multi-language speech synthesizer is discussed in the study [GYO01] which is able to convert the text data of a particular language into speech data in that similar language. The quality of the speech synthesizer is analyzed on the basis of how similar it is to the human voice, speaking style and its clarity in spoken words. Customization of speaking style discussed in [JC06] based on the semantic analysis which is shown in the figure 2-5. The figure shows the selection of the speaking style based on the topic and prosodic settings. The speech synthesizer uses this information to generate an audio signal.

R.E. Donovan and P.C. Woodland proposed a new approach to synthesize speech in their research paper [DRP99] in which hidden Markov models are used consists of a set of cross-word decision-tree that are dependent on the context. The clustered states are represented by models, trees and waveform segments etc. which are obtained after the training on continuous speech database for about 1 hour. It further states that the system is already successfully trained on four voices and can be retrained on a new voice in less than 48 hours.

Another model was proposed in the paper [AR82] by Bishnu S. Atal and Joel R. Remde, which states that generation of all classes of sound signal can be done by stimulating with linear predictive coding (LPC) filter through a sequence of pulses. Their model based on two main functions, firstly, it generates the characteristics of the vocal span with the help of the linear filter and secondly shapes the spectrum of the vocal source (see figure 2-6).

The system proposed in this work uses similar techniques to convert the text into an audio signal. The code is written in the Python language which uses a text-to-speech library for the processing and conversion. Till this point we have explained speech to text conversion and vice versa but in order to plan and execute these commands, a task description language is required which we will be discussed in the next section.

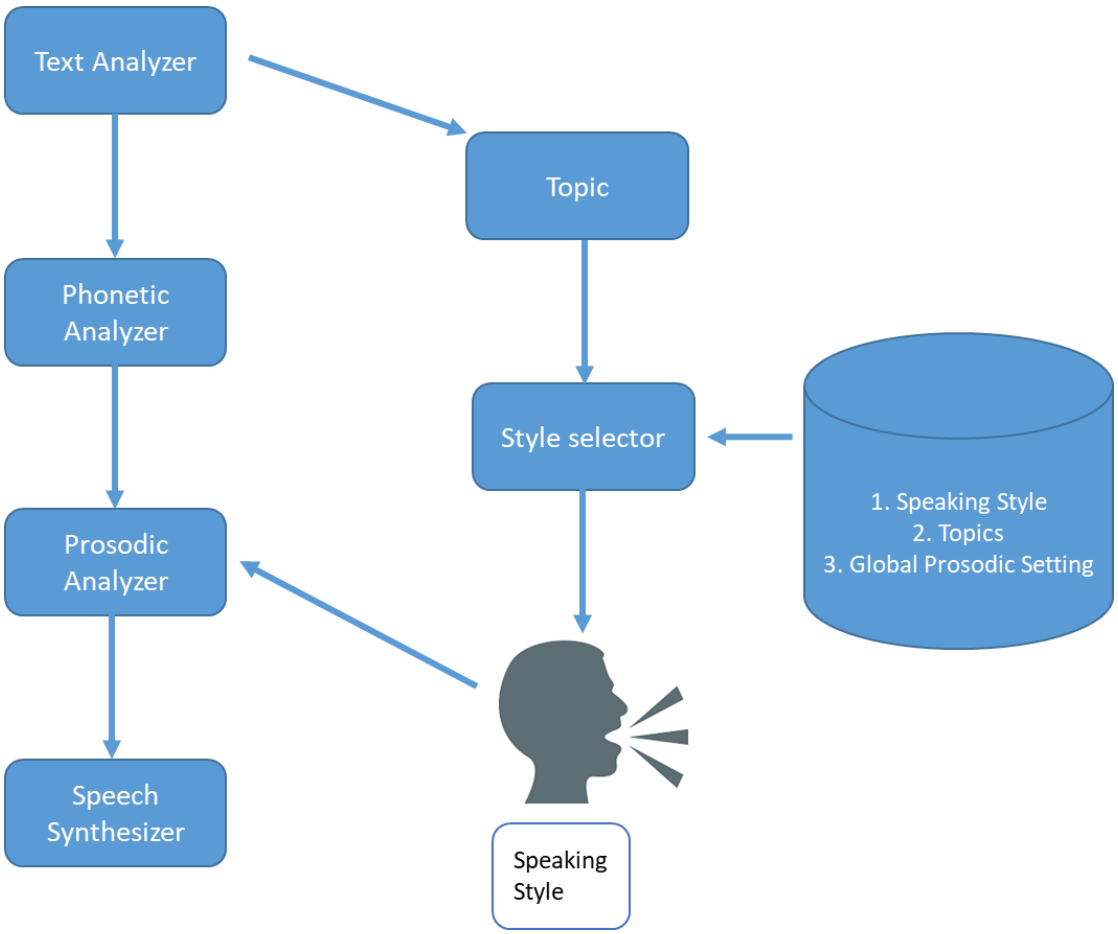

Figure 2-5: Customization of the speaking style based on semantic analysis, based on figure 2 [JC06].
The text analyzer transfer the text to phonetic analyzer which converts it into corresponding phoneme transcription. The style selector selects the speaking style based on the topic and sends to the prosodic analyzer. It process and sends the received phoneme data and speaking style to the speech synthesizer which converts it into an audio signal.

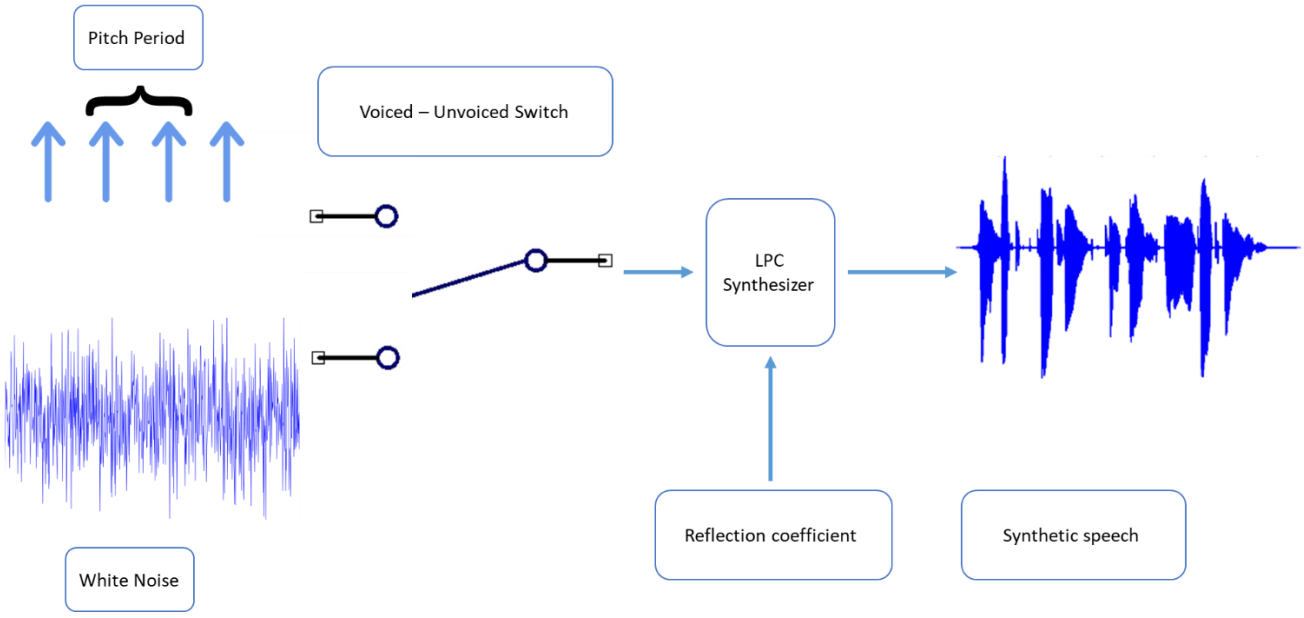

Figure 2-6: LPC speech synthesizer based on figure 1 [AR82].
The linear predictive coding (LPC) which synthesize the speech. It uses periodic pulses in order to synthesize the voiced speech and the unvoiced speech is synthesized with the help of white noise.

2.1.6 Task description language

Language enables humans to interact, exchange concepts and ideas with each other, in order to enable machines to interact similarly we require a language which provides this functionality. A task description language is referred to as a programming language which is the extension of the C++ language. It consists of asynchronous procedure calls which run concurrently allowing the system to do multitasking. Task description language is best suited for architectures where events occur asynchronously.

This language is proposed in [SA98], which states that this language supports decomposition of tasks into smaller tasks, synchronization of multiple concurrent tasks, monitoring and proper handling of exceptions occurred during the execution. It requires a compiler which transforms the TDL code into C++ code in order to execute it.

A system is considered as a good system if it is able to perform assigned tasks while remaining responsive for new tasks and able to handle exceptions efficiently which are referred to as task-level control [SR94]. It is represented by three-tiered robot control architecture which is shown in figure 2-7. The first layer is the Behavior layer which is responsible for interactions with the real world environment, collection of data from sensors, controlling hardware and displaying results.

The second layer is the executive layer which is responsible for handling exceptions, processing of received data from behavior layer, mapping of goals to their respective low-level commands, execution and monitoring of these commands. The third layer, which is the planning layer, is responsible for planning and organizing desired goals (see figure 2-7). Using conventional programming languages usually results in highly nonlinear code when implementing such task-level control functions [SA98].

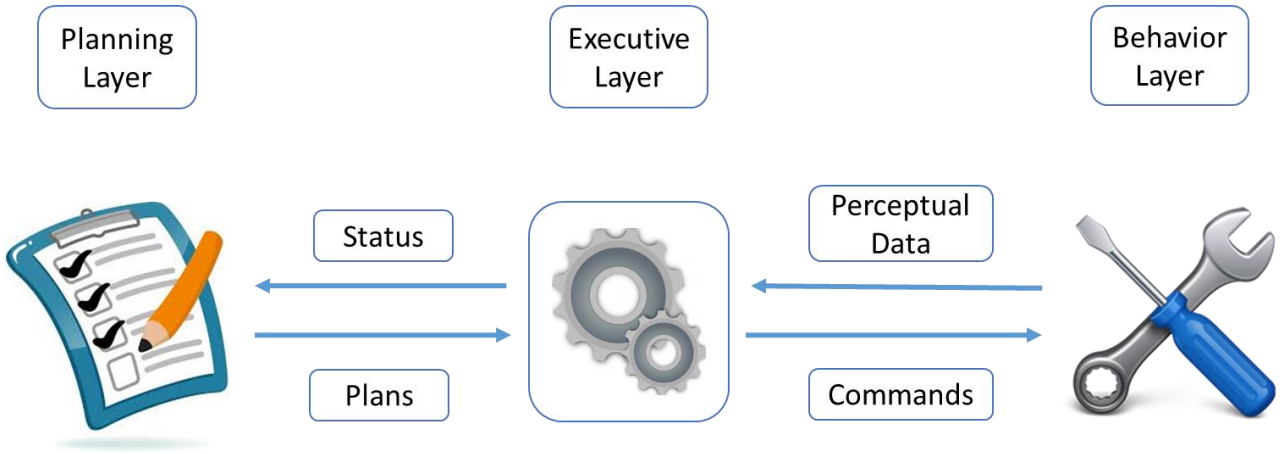

Figure 2-7: Three-Tiered control architecture based on figure 1 [SA98].
The first tier is responsible for the planning of the tasks while the second tier manages the commands and operations and the third tier performs operations.

The planning layer constructs and sends the tasks to an executive layer which converts them into the commands that the system executes in order to perform the operation. It is also responsible for receiving the data back from system and informing the planning layer about the current status of the system. This approach includes creation of task trees which divide the task into parent-child relationships. The task description language creates such task trees and executes them. The figure 2-8 shows an example of a task tree where each node of the tree has a specific action to perform. The action can be adding a new child in the tree, processing the data, interacting with the sensors etc. the result of these actions can be a success or a failure, and then further actions are performed on the obtained result. Each task tree shows the separate execution part of the whole program. The state of each task tree is divided into four states active, disabled, enabled and

completed. The state is considered as active when the node is processing or performing the task. The handling of the node switch to the disabled state if it waits for other events to occur. When all the required events occur the disabled node shift to the enabled state. Finally, when the operation finishes either with success or failure the state changed to completed. However, there is a possibility for the node to be in the enabled state but not performing actions due to insufficient resources either computational or physical.

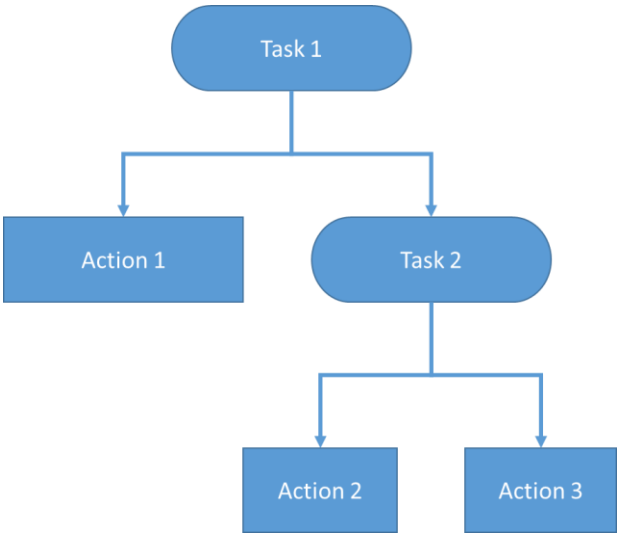

Figure 2-8: Task tree based on figure 2 [SA98]
The control flow of the task. Each node represents the action which includes computations or interaction.

2.1.7 Turn-taking

Turn-taking can be considered to be an extension of a task tree. In human-robot interaction turn-taking means actions performed by both parties when required. It is considered as the human turn if it is required to give instruction to the robot, then it is robot's turn to receive the information, analyze it, process and perform the required action. It informs the user after completion of the task and again it becomes human's turn to give another command or do some action if required. The human-robot interaction would be fluent if this turn-taking is fluent [TAC11]. A human dialogue system composed of four states seizing, passing, holding and listening. A dialogue system and response model is presented in [JHS+16]. It can be seen from the figure 2-9 that response model makes a decision about what and when to respond, in order to remove ambiguities, as a single sentence can have multiple meanings, or to provide feedback regarding the task. The proposed model was trained on human-robot dialogue data which makes the system more accurate.

In order to create better dialogue system for human-robot interaction, it is required to consider the nature of the dialogue between human and human [JMG+14]. The same turn-taking experiment is done with two teams of human-human and human-robot in order to analyze the system. In order to implement the same behavior for human-robot dialogue system, first, it is required to analyze how humans communicate with each other. In human-human dialogue, to keep the turn they use filled pauses, incomplete phrases, flat specific tone, specific gestures etc. To direct their attention they look towards the speaker, make sounds like "hmmm" etc. There are several components of an intelligent dialogue system [Gor01]. User model, it is referred to as the crucial component in an intelligent interface. It deals with the reception of information as well as the display of information.

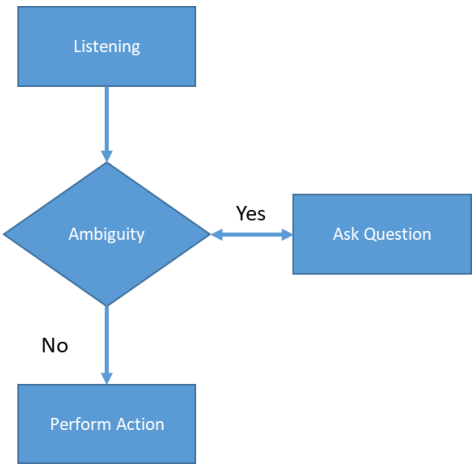

Figure 2-9: Dialogue management system based on figure 2 [JHS+16]
A dialogue management system to eliminate the ambiguities from the speech commands.

Task model describes how humans perform the tasks based on prior knowledge. They also discussed the GOMS model (Goals, Operations, Methods and Selection of rules). It describes that the human behavior can be observed as a hierarchy of primary tasks and sub-tasks while performing routine computer tasks. Dialogue management referred to as the proper set of lines of the human-robot dialogue. It ensures that both the parties can understand each other. The role of the dialogue manager is to translate the human requests so the robot can understand and conversion of robot's feedback into the natural language which is understandable by the humans.

The system proposed in this thesis implements turn-taking in similar manner. First the speech manager gives the opportunity to the user to provide the command. After receiving the command it transfers the control along with the command to the task manager so it can process it. The task manager processes it and performs the desired operation. The speech manager provides the feedback to the user, then again it becomes user's turn to provide the next command.

2.1.8 Robot design

Despite the fact that fluent human-robot interaction is heavily dependent on processing of information, the accuracy of data, artificial intelligence etc. the appearance and design of the robot also plays an important role. The shape and appearance of the robot have a great impact on the interaction of the human with it. For example, a vending machine which looks like a square box just treated as a machine used to grab the money and provide stuff, on the other hand, the machine or robot which looks like a human will be treated differently [Dau99], [CL01]. This is the reasons why more and more robots have faces with cameras placed similarly to eyes, hands, lip reading skills and other features which make the human-robot interaction more natural [DOQ02]. The design of the robot must match its intended function [FND03] that is if the robot is designed to perform tasks then its form must convey "product-ness" similarly if the interaction is more important than it should appear like humans so that the user feels comfortable interacting with it.

The robot, PR2, used in this thesis is of about human height with two arms. The arms are equipped with two grippers to pick and place the objects. The robot has attached wheels to move around. The Kinect, along with depth sensor, is attached on top of its heads to perceive the real world.

2.1.8.1 Safety

The safety is the main concern when it comes to the human-robot interaction. The robot must be designed in a manner such that it is safe and humans feel comfortable during an interaction. Previously robots were not considered safe as humans did not have much control over them and it has always been a topic of discussion that how close human can go to the robots [YKI+08], but nowadays robots become very smart and

with the help of multiple sensors and safety measures human-robot interaction reached a good degree of safety level but there is still room for improvement in many areas.

2.1.8.2 How robot should interact

Today's robots are considered very smart as they can process tons of information and are able to make decisions efficiently so they can also give advice to humans in some situations [TFK13]. Author of that paper suggests that sometimes advice is appreciated by the humans but sometimes it feels offended and irritated. They explore different ways in which a robot can give advice. The use of encouraging and polite sentences help in order to have proper communication between human and robot. Multiple researchers have explored this area and a theory was proposed namely "politeness theory" [LBL+87] which describes different ways of saying a sentence in order to avoid them sounding like threats to listeners. For example instead of saying "Pick up the box and place there" can be rephrased in a polite manner like "I think if you could please pick up the box and place it there then it would be great". The usage of Hedges in a sentence also make them less forced messages but it includes a degree of uncertainty. Another approach is the use of discourse markers (false starts, repeated words and fillers) in the speech such as "As you know," "uhm," "well" etc. Regarding the social interaction of the robot, there are many studies which show social as well as collaborative aspects of human-robot interaction [FKH+06]. Kerstin et al. [KGH+03] made a survey to investigate the social interaction of the robots.

A Similar methodology is used in this thesis. When the speech manager detects ambiguity in the command it initiates the dialogue with the user. For example if the user said "Place the large object" then it is not clear that which object needs to be placed so speech manager asks the user "please specify the color of the object" as the objects used in the user study of this thesis, distinguished based on the size and the color.

2.1.8.3 Perception

It is defined as the ability to see, hear, feel or become aware of something with the help of senses in case of humans, but this is achieved with the help of multiple sensors in case of machines, for example, in order to visualize to environment video cameras are used, to get audio input microphones are used, to get the information of depth infrared sensors are used and so on. Machine perception is the ability of the machines to interpret the data in a similar manner as humans. This can be achieved with the help of lots of sensors. The accuracy of the sensors is crucial in order to increase the performance of the system. The authors of the paper [CTB+04] discussed how the robots will perform more complex tasks efficiently by integrating perception, action and cognition through mental simulation in robots.

The main idea is to develop an architecture which helps the robots in proper planning, reasoning and mobility algorithms by managing mental simulations as most of the problems in robotics arise due to the complex processing of data received from multiple sensors. They called their architecture Polybot which is based on the Polyscheme cognitive architecture in order to solve integration problems [CL01].

The Polyscheme differs from traditional cognitive architectures in many ways, its fundamental approach is to develop the algorithm which enables proper reasoning and planning using perceptual based mental simulations along with traditional artificial intelligence. It uses reactive components which only triggers and sends the data when they actually interact with the objects or environment in the real world.

In order to obtain emotional related responses from the robots and other machines, the primary step is to create an accurate method for emotion recognition, this approach is presented in this paper [AZS17]. The deep learning model of Convolutional neural networks plays a crucial role in implementing this technique. The proposed technique claims to achieve the accuracy of 71.33% for six emotions (fear, happiness, anger, surprise, disgust and sadness). According to [RN96] human find the interaction with the robots more enjoyable and meaningful if they are close to human in perceiving the environment and performing actions.

2.1.8.4 Interface

An interface is referred to as the boundary through which separate components share information with each other. The two parties sharing the information can be humans, robots, devices and their combinations. Some interfaces provide the functionality of both sending and receiving the data while others just provide either receiving or sending the data. In this thesis, we have used two interfaces to interact with the robot speech and text which will be discussed in detail later in chapter 5.

2.2 Related work

This section focuses on major contributions of past researchers in the field of human-robot interaction. We will discuss their findings and make a comparison of how this work resembles their work and how this work is an extension of previous research.

2.2.1 Human-robot interaction

Researchers are proposing methodologies and algorithms to make the interaction of human and robot more efficient, safe, natural and user-friendly. They investigate different modalities to find out ways to improve the performance. Sinder et al. [SKL+04] investigate the interaction of human and robot in three different ways: spoken language, beat gestures with its arm and head gestures to track the user and objects. The system uses several algorithms to perform its tasks: face detection, sound location, speech detection, object recognition and fuse the data obtained from the sensors. There are lots of data to be processed coming from different sensors so in order to reduce the computation an idea is presented in a research paper by Hara et al. [HAA+04] which suggest that it would be better if robots are directed towards the humans with the help of human skin. In this study, two methods were used for this purpose. The first technique is known as the skin-colored model. The human skin is made up of chrominance which is easily distinguishable. RGB representation of colors needs to be normalized to detect the skin color because the devices used to capture images, not only capture colors but also the brightness. In order to identify the skin color easily, the color space can be normalized as shown in the following equation [HAA+04].

$$R = \frac{r}{(r+g+b)} \qquad \text{Equation 2.1}$$

$$G = \frac{g}{(r+g+b)} \qquad \text{Equation 2.2}$$

After normalization it becomes easier to represent the skin color model with the help of two-dimensional Gaussian model $\mathrm{N}(\mathrm{m}, \sum^2)$ where "m" is the mean vector of (R, G) and "$\sum$" is the covariance matrix. The second method is known as kernel-based tracking is real time human tracking and in order to increase the performance, it is better to keep the computational complexity as low as possible.

Most of the studies done by several researchers focus on human-robot interaction with a single system and few users so Tews et al. [TMS03] proposed a scalable approach to human-robot interaction to address this problem. The main idea behind the study is the development of a general interaction infrastructure which supports large scale human-robot applications. Their proposed infrastructure is basically a server-client infrastructure which supports both kind of interaction that is many to many and one to one (see figure 2-10). Single user architecture allows the user to use full available resources of the system while these resources are shared in multi-user system architecture. The server is the centralized hub which controls all the operations such as allocation of resources, providing services to the clients, handling of requests, maintaining the database and so on. The contact information of the server is known to all the other systems within the

infrastructure and UDP connection is used in order to transfer the data. Server actively checks all the systems in the infrastructure to make sure they are still serviceable and if any system didn't respond under predefined time then it is removed from the database of available systems. Whenever new users connect to the infrastructure, a list of all the services are provided to them. When the user generates the request to select the service, this request is analyzed by the server to check whether the requested service is available or not. The server grants the service to the user if it is available otherwise the request is added in the queue, the figure below illustrates the proposed architecture in this study.

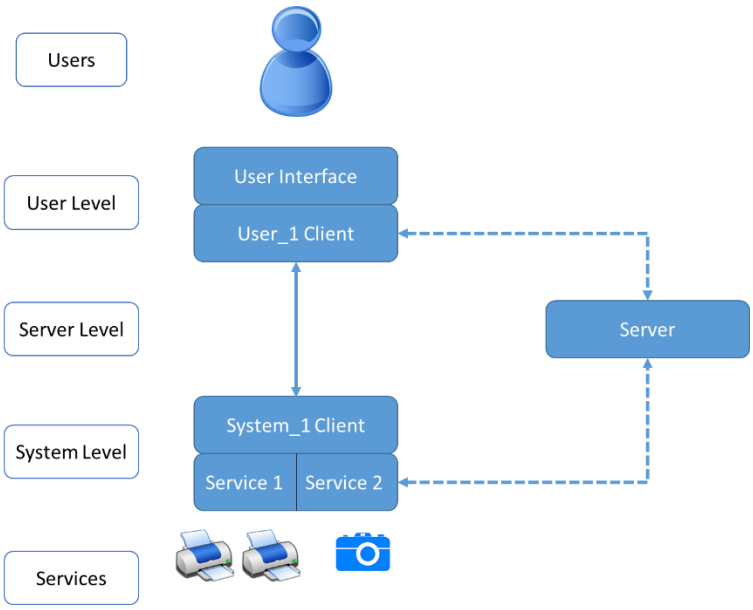

Figure 2-10: Multi-user system architecture based on figure 1 [TMS03].
*An overvie*w of the multi-system architecture which allows multiple users to connect multiple system services. The solid lines represents the one-to-one interaction while dotted lines represents the many-to-many interaction.

A Similar approach of [TMS03] is followed in this thesis, the speech manager and the task manager creates a one-to-one TCP connection in order to exchange the data. The speech manager acts as a server, implements server -ide socket and waits for the client to become active. The task manager acts as a client, implement client-side socket sends the request to the server. The server accepts its request and sends the acknowledgment to the client. After receiving the acknowledgement both the parties become ready to send and receive the data. This method is further extendable to implement many-to-many server-client interaction.

2.2.2 Human-robot dialogue system

Atrash et al. [AKV+09] present a dialogue system for human-robot interaction. The main goal of their work was to address two major challenges in the development of a speech interface for human-robot interaction. First was the processing of voice commands for which they have proposed a complete methodology that includes natural language processing, syntactic analyses, semantic parsing and algorithm of decision making to generate the feedback for the user. Second, they proposed the architecture to develop the tools and standards for the testing of robots which helps to test human-robot interaction especially speech recognition. The architecture of their software consists of several modules such as a module to handle different modalities of communication (speech and tactile), a module to handle the translation and parsing of the speech signal with the help of semantic grammar, an interaction manager which is responsible to make decisions, a behavior manager which is responsible to handle the mapping of parsed speech signals and a control manager which control the movement of the robot. To implement the speech recognition, they used two open source

speech recognition systems CMU's Sphinx-4 (2004)[1] and HTK (2006) HTK 3.4[2]. A small vocabulary is created in order to test the system which also improved the accuracy and the performance of speech recognition. The error rate of the sentences was reported as 45.2% with Sphinx-4 and 46.7% with HTK while the error rate of the word was reported as 16.1% and 16.6% respectively. The interaction manager gets the parsed command from the user and sends it to the behavior manager to perform the action. They methodology to map the commands is given as, z = (Command value = v_1, Command type = v_2, Direction value = v_3 and so on) which can be written as z = (v_1, v_2 , v_3, …) the assignment of values to the slots can be written as

$$P(s_i|v_1, v_2, v_3, \dots) = \frac{P(v_1, v_2, v_3, \dots|s_i)P(s_i)}{P(v_1, v_2, v_3, \dots)} \quad \text{Equation 2.3}$$

$$P(s_i|v_1, v_2, v_3, \dots) = \frac{P(v_1|s_i)P(v_2|s_i)P(v_3|s_i) \dots P(s_i)}{P(v_1)P(v_2)P(v_3) \dots} \quad \text{Equation 2.4}$$

These values, $P(v_j|s_i), P(v_j), P(s_i)$ can be calculated from the data collected during testing.

The speech manager proposed in this thesis also uses small vocabulary similar to [AKV+09], this vocabulary consists of colors and sizes. It matches the color and the size mentioned in the sentence to determine the object. After determining the object the command is sent to the task manager for further execution.

The communication in natural language is very easy but there are always many ambiguities in spoken sentences, researchers proposed many approaches to designing an intelligent dialogue management system which takes care of these ambiguities. Roy et al. [RPT00] proposed such a dialogue management system with the help of probabilistic reasoning. The main idea underneath the creation of dialogue strategies, using Partially Observable Markov Decision Process (POMDP) style, is to consider the intentions of the user rather than the state of the system. The proposed approach in this paper is basically an extension of Markov Decision Process (MDP) which consist of these factor set of states, set of actions, set of transition probabilities, set of rewards and an initial set. POMDP adds few more factors in it such as a set of observation probabilities while replacing a set of rewards with rewards condition and initial state with initial belief. The system can eliminate the uncertainty by making the assumptions about that uncertainty which helps the system to summarize that belief with the pair of most likely state and its entropy

$$p(s) \cong < argmax\, p(s); H(p(s)) > \quad \text{Equation 2.5}$$

$$H(p(s)) = -\sum_{i=1}^{N} p(s) log_2\, p(s) \quad \text{Equation 2.6}$$

With the help of this assumption, the state and entropy pair can be used to plan the policy of POMDP for corresponding belief p(s) which helps in reducing the ambiguities within sentences of the dialogue.

2.2.3 Performance of interaction

Staudte et al. [SC11] proposed the approach of using attention mechanisms together with the speech to eliminate the ambiguities from the voice commands which results in the better performance of the system. According to them, language is very ambiguous and adding nonverbal information together with verbal

1 https://sourceforge.net/projects/cmusphinx/files/sphinx4/5prealpha/

2 http://htk.eng.cam.ac.uk/

commands helps to reduce these ambiguities. They investigate the role of "pointing" towards the object together with voice command and found that gaze has a special status in such pointing actions as usually human use their eyes to focus on something or to divert attention toward something. A similar approach is also discussed in other studies such as [CHM04] and [TMC07] which describe that humans continuously monitor the gaze of the speaker in order to eliminate the ambiguities from the spoken sentences.

Norberto et al. [Nor05] performed experiments of controlling the two industrial robots by human commands in order to improve the human-robot interaction through speech. One robot is only capable to pick-up and place objects while the second one is capable to perform welding. The speech recognition was very challenging because of the noise interference as the industrial environment is very noisy and secondly, they don't have dedicated powerful computers just for human-robot interaction. To overcome the issue of noise they use short commands structure along with predefined word [Nor05]. The development of speech to text and text to speech conversion software includes Microsoft Speech Engine[3] (Microsoft Corporation, 2004), Microsoft Speech Application Programming Interface[4] (SAPI) and Microsoft's speech SDK (version 5.1)[5].

The better performance of human-robot interaction has always been a topic of research among researchers. Sinder et al. [SKL+04] suggests that the performance of the robot would increase if the conversation initiated by the robot in order to reduce unpredictable commands. They evaluate the performance of the robot on five different factors ease of interaction with the robot, knowledge of the action, engagement in the interaction, reliability of the robot and effectiveness of the movements. The performance of the robot is also dependent on other factors like its appearance, level of ease and comfort in completing tasks. Brennan et al. [BSH93] suggest that if the system has proper feedback mechanism then it will also improve the performance of the system significantly as the user knows the system's state, so the user can help the system in achieving the goal. They divided the state of the system into 8 levels which are Level 0: System is active, Level 1: Gain system's attention, Level 2: Partial result of the system, Level 3: Processing of natural language, Level 4: Produces responses of the system, Level 5: Ask the query to the user, Level 6: Performing actions and Level 7: Final result

It can be seen from the results of user study (chapter 6) that speech interface proposed in this thesis performed well in less noisy environment. The interference of noise results in larger delays in speech to text conversion and repetition of voice commands. We have also compared speech interface with other modalities of interaction such as text-input and graphical user interface.

2.2.2 Modalities of interaction

The processing of natural language is a massive challenge to deal with in human-robot interaction through speech, so mostly it is argued that why not use other modes of interaction? Kulyukin et al. [Kul06] answer this question in their study with the help of three arguments. First, human language is the most natural mode of communication. Second, other interaction modalities like graphical user interface are suitable when the operator has easy access to the hardware device. Third, Robots which are capable of natural language processing starts arguing with the human unnecessarily. They have also developed a dialogue management system which consists of speech recognition and speech synthesizer in order to process natural language commands.

2.3 Our approach

Thus it is observed that in order to develop a system which enables the user to interact with the robot in natural language, the system must consist of these factors: speech recognizer, dialogue management system

3 https://www.microsoft.com/en-us/download/details.aspx?id=27224

4 https://docs.microsoft.com/en-us/azure/cognitive-services/speech/getstarted/getstartedcsharpdesktop

5 https://www.microsoft.com/en-us/download/details.aspx?id=10121

[KU99], a speech synthesizer, task manager and feedback methodology. According to [BSH93], if a system is able to map the voice commands exactly and efficiently, then send it to the robot then speech will be a most successful input technique.

This thesis utilizes the concepts learned from previously discussed research studies. The proposed approach uses the code-base provided by Machine learning and robotics department from the Institute of Parallel and Distributed Systems (IPVS) of the University of Stuttgart6. This research work is the extension of previous research studies [SKM18], [Kar17], [Blo17] and [Kra17].

We have used 5 tasks that is sort, stack, build, balance and shape to evaluate the system performance. The first 4 were already used in the previous study [SKM18]. In this study we are using two modes of interaction that is human commands and autonomous which were designed and used in the previous study [Kar17]. Marietta Inge Bloch [Blo17], proposed an architecture for natural language processing systems that resolves the ambiguities in voice commands by asking questions to the user. It fills the missing information in the command and then executes it. The proposed approach in the study uses dependency parsing for speech recognition. Umut Kara [Kar17] extends the research work presented in [Blo17] by mapping natural language to abstract commands with the help of syntactic parsing and representation of meaning by Spatial description clauses. The planner executes recursive tasks without processing same command again and again which reduce the processing overhead.

The architecture proposed in this thesis follows a similar approach discussed in [Blo17] and [Kar17]. We used google speech to text conversion to reduce the computational overhead. After the conversion, the speech manager looks for the required information in the command and in case of ambiguity or missing information, it asks the user for the required information. Then it maps the command to its corresponding value and sends to the task manager to perform the operation. The architecture proposed in this thesis supports multiple languages which was missing in previous studies.

6 https://ipvs.informatik.uni-stuttgart.de/mlr/

3 System design

To achieve the objective of the thesis, a speech interface for human-robot interaction has been developed. The interface consists of backend logic, speech manager, server-client based communication, task manager, speech synthesizer and speech recognition, the figure 3-1 shows the system architecture. The speech interface for human-robot interaction is evaluated with PR2 by giving it instructions to complete the specific tasks. This evaluation consists of different tasks and modes of interactions which will be further discussed in details in the next chapter.

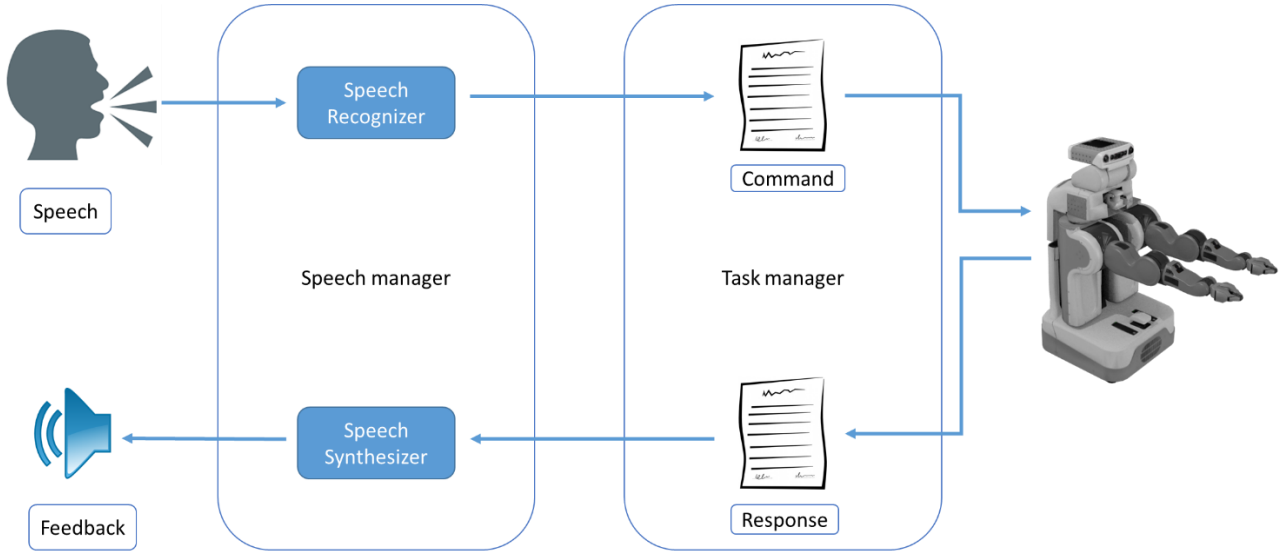

Figure 3-1: System design[7]
The speech manager interact with the user and the task manager, which interact with the robot to send commands and get feedback of the current status.

3.1 System environment

The system is evaluated inside the laboratory of machine learning and robotics department from the Institute of Parallel and Distributed Systems (IPVS) of the University of Stuttgart, where the whole setup is done. The objects are placed on the table on which different tasks have to be performed. The height of the table is 70 cm. The objects consist of three different sizes small, medium and large. These objects are also categorized with different colors like blue, green, yellow and red. A combination of color and size is used to emphasize on a particular box. For example, to perform an action on the specific block, a user can say "Place large red object".

PR2 is placed on one side of the table while the user stands on the other side, this allows both of them to easily pick and place the objects in suitable locations. The speech interface is running on the computer with a microphone attached in order to provide voice commands. The instructions are gathered from the microphone then provided to the robot after processing, PR2 performs the desired action and returns the feedback which is delivered to the user with the help of the speakers.

There, few things need to be considered, the environment should be less noisy since the speech interface takes voice input so noise is the biggest challenge. If the speech recognizer gets the input with less noise then it processes this information quickly, for this purpose we set the minimum threshold of the sound energy level to 50, so the speech recognizer will not process any sound below this threshold. A good quality of microphone also plays an important role in reducing noise from speech signals. We also set the time for

7 The image of PR2 is reproduced from http://www.openrobots.org/morse/doc/1.4/user/robots/pr2.html

listening of the audio signal to 4 seconds so that the recognizer should not fall into an infinite loop of listening. The speech manager sends the commands to the task manager which perform the respective action on the PR2. The computer is connected with PR2 through SSH.

3.2 PR2

The PR2 is a descendant of PR1, where PR stands for “Personal Robot”. It was developed in a robotics research lab called "Willow Garage". It is an open source software system written in a robot operating system (ROS). ROS interfaces are responsible to provide all the capabilities of PR2. PR2 is the first major successful robot of Willow Garage, it is near to the size of an average human (see figure 3-2). It has two arms which have 7 degrees of freedom. The grippers are attached with each arm which is used to pick up the objects. Wheels are attached at the bottom in order to move it without much difficulty. A 5-megapixel camera is mounted on its head which is used to perceive the objects with the help of other sensors including laser rangefinder and an inertial measurement Unit (IMU). The laser rangefinder determines the distance of the object with the help of the laser beam consist on the principle of "time of flight". A narrow laser pulse is projected towards the object and the sensor measure the time taken by the pulse to return back after hitting the object. Furthermore, it is equipped with two 16 core servers which have 24 GB of Ram each.

Figure 3-2: PR2 robot[8]
The PR2 robot, designed for robot researchers.

8 The image of PR2 is reproduced from https://www.ics.ei.tum.de/en/research/platforms/willow-garages-pr2/

3.3 Working of the system

In this section, we will describe the working of the system developed to achieve the goal of this thesis. The system consists of multiple parts speech manager to control the human-robot dialogue, task manager to control the execution of the tasks and connection between the two to exchange data and information. The working of different parts will be explained with the help of figures where necessary.

3.3.1 Speech manager

The speech manager is responsible to take voice input commands from the user. In order to provide the ease of communication, there is a possibility to switch language. Then the speech manager will ask the questions in that language. Then it translates these commands into the text to find out the required information from it. If the command is not clear enough or there is any ambiguity then it is also handled by the speech manager. It resolves these issues by initiating the dialogue with the user. After that, it maps that information to the corresponding value which needs to be sent to the task manager. The mapping of values is very crucial for the accuracy of the operation because the task manager sends the instructions to the robot based on the received values. The robot executes the instructions and sends back the result to task manager which is delivered back to speech manager and based on the received feedback speech manager inform the user about the success, failure or the completion of the task. To implement the natural language understanding system there are several factors which have to be taken into account such as spoken language, meaning representation, elimination of ambiguity and feedback.

3.3.2 Language of interaction

Spoken language plays an important role in order to keep the system as user-friendly as possible. The system is able to communicate in different languages with the user, however, we have implemented English and German language. We are using Python language along with Google speech API which handles the input and output of the commands. The system gives the opportunity to the user to select his desired language for communication. As the system is using Google speech API so internet connectivity is required, also a good microphone is required which will get the instructions from the user without much noise interference.

The main idea of using multiple languages is to provide ease of interaction with the robot and improve the performance. Sometimes it is very difficult for a person to pronounce the words of different language correctly and it is very frustrating for the user to repeat the same sentence again and again. For example, if the system is expecting a user to specify the color of the object such as "red color" but it always recognize it as "bread color" then the system will not further execute the command an keep asking for the color. This is just one example, but in any language, there are multiple words which sound almost similar to other words. Sometimes Humans also not hear the word clearly but they understand the meaning by analyzing the context of the sentence which is not very easy for the machines. So in order to address we implement the option for the user to change the language. The system gets the command from the user and parses it to find the required information from that sentence.

3.3.3 Meaning representation

The parsed sentence is used to provide instructions to the robot to perform the desired operation. It must contain clear identification of action, object, and location. If any of these is missing in the sentence then the system should ask explicitly for it. A good example of the clear sentence is "put the red large object on point A” which contains the action (put), object (red, large), and location (point A), whereas a bad example of a sentence is “put the large box on point A” where identification of the object is missing so system will ask to provide the required information by saying “please specify the object”.

3.3.4 Elimination of ambiguity

There are multiple ways to say a single sentence in natural language so there are huge chances of ambiguity. Our system resolves this ambiguity by initiating a dialogue with the user until the task is clear. For example, if the user says "put the red large object there", this sentence contains all the required information action, object and location but still, there is one problem in it, that is, the location is ambiguous. Human to human conversation contains such kinds of ambiguous statements which are clarified with the combination of gestures, head movements or any other form of identification but our system is not able to observe these kinds of gestures so the sentence must be clear enough to understand. In this specific case, the system asks to clarify the location by asking “please specify the location”. Then, user will provide the information about the location. If this information is sufficient then system will send the command to the robot to perform the action otherwise it will ask the location again.

3.3.5 Feedback

In the end, after parsing and resolving the ambiguity the instruction is delivered to the robot so it can perform the operation accordingly. There may be two possibilities that are either the robot is able to perform the operation or it is not able to perform it. In both of these cases, the system should provide the feedback to the user. If the task is completed successfully then the system informs the user of the message that the previous task is completed successfully and the system is ready to perform the next task. On the other hand, if the system fails to complete the task then it should inform the user about the failure along with the reason of failure if possible.

In our study, we have designed five different tasks which robot should perform. In all the tasks user instruct the robot to place the object. After receiving the command robot first inspect the current state of the objects taking into account which task it is performing. It is detected that the object is already placed or the order of the object is not right then the action is canceled and the robot sends an error message. Similarly, if the robot detects that the request operation is available to perform then it performs the action and sends the success message. Finally, after placing all the objects robot send the finish message which shows the completion of the task.

3.3.6 Design of speech interface

To implement the speech recognition we used Python language which runs on the computer also controls a microphone and speaker attached to it for input and output respectively. It is also responsible to handle the language selection which allows the user to communicate inappropriate language.

The code to control the robot is written in C++ language and speech recognizer is written in Python language so integrating these codes is one of the biggest challenges. We consider different approaches which have their own pros and cons so it is required to define a specific criterion to select an appropriate approach.

The first approach is the collaboration and the communication of these codes snippets that is one can handle speech tasks while other is responsible for the controlling of the robot and they are able to communicate with each other. The second approach is that we embed the Python code in C++ so we can call these functions directly. We consider two main aspects of the selection process which are as follows

1. The selected approach works reliably with the existing system.

2. It can be easily understandable by others and they can use it easily for their projects.

So we select the first approach. In order to make them communicate with each other, we create a server-client relationship between them where python code serves as a server while C++ serve as a client. A TCP connection is created between them so they can communicate reliably with each other. To create this connection we used IP of localhost, as both the codes are running on the same machine, and port number

1500. After the successful connection, the system asks the user for the selection of the language. The system reads the commands in a different language so the user can understand clearly and wait for the response. It will wait for a certain amount of time and read the instructions again.

After the selection of the desired language, the system provides the feedback in that language to aware the user, then the system asks the user to select the mode of operation by reading the names of these modes, similarly, it asks for the tasks by reading the names of the tasks. After every step system provide the feedback to the user so user remains updated.

3.3.7 Communication with the robot

To run the code on the robot, we have to communicate to the robot which can be done through SSH. After that, we can start the robot by executing the command "robStart". In the beginning, it does the initial calibration, that is, it stretches its torso and moves his hands and meanwhile, it shows the dashboard on the computer screen from where we can enable the breakers and reset the motors. The next thing is to clone the repository on the robot to run the code base provided by the machine learning and robotics laboratory from the Institute of Parallel and Distributed Systems of the University of Stuttgart. After the compilation of that codebase, we can start the real-time controller with the help of the following command "robStartControllerBigBird". After completing all the listed steps, the robot becomes ready to execute the commands. Once all the tasks are finished we can turn off the robot with the following command "robot stop", then we can release the resource by executing “robot release” and then finally “pr2-shutdown”.

3.3.8 Task execution

PR2 is fixed during the study so in order to perform the operation, objects should be in range. The objects are placed on the table in front of the robot so its height is adjusted accordingly with the help of motor attached to it. To perceive the objects on the table, PR2 uses a camera mounted on its head. Its head can be configured to look at the center of the table so that all the objects are in the field of view. The arms can also be configured to the initial position, close to the shoulders, with the help of the ROS so it can reach the objects easily without any obstacle.

The robot is ready to take instructions after resetting the motors, torso and arms to the initial positions. The distribution of tasks, moving the arms and determination of objects are the responsibilities of a task manager. First, it takes the command then after analysis it move the specific arm to pick up or place the particular object. The grippers are attached to the end of each arm which enable the PR2 to hold the object. The information on the XY coordinates is required to place the object which is also handled by the task manager.

After the completion of the task, the task controller reset the arms to the initial position which shows that the robot is ready to execute the next instruction. It provides the feedback through the TCP connection to the server running in the Python language. This feedback is handled by the speech recognizer which has implemented the "text to speech" function, inform the user with the appropriate message.

3.4 Programming languages, Tools and libraries

This section describes the programming languages, tools and libraries we used in this thesis in order to address the issue discussed in the beginning.

3.4.1 Programming languages

Python language to develop a speech interface and C++ to communicate with the robot and execute commands.

3.4.2 Tools

The speech interface is written in python language so we select PyCharm IDE[9] for the development. Specifications of the IDE are: PyCharm community edition, Version: 2018.2.1, Build: 182.3911.33

3.4.3 Libraries

speech_recognition

A speech recognizer is used to detect the input command. It has the ability to get the input from the microphone as well as through the audio file. The voice input received from the microphone always contains some noise which can be reduced with the help of the following command:

```
speechRecognizer.adjust_for_ambient_noise(source).
```

Google speech API

The received voice input is processed with the Google speech API which is responsible to convert it from speech to text.

TextToSpeech

The robot must inform the user about the current status, whether the instruction is executed successfully or not. In case of error, the robot will update the user so that user gives the appropriate instruction again. The robot will also inform the user after the completion of the whole task. The robot will provide the feedback in verbal form for which we need a text-to-speech library.

9 https://www.jetbrains.com/pycharm/

4 Tasks

The speech interface designed to achieve the goal of this thesis is tested on five different tasks namely sort, stack, bridge, balance and shape. The first four tasks were already been used in the previous study [SKM18] these tasks are sorted based on the difficulty level which can be defined as the order of operations needed to complete each task, the required degree of attention. All the tasks are performed with the same set of objects which are placed on the defined initial position before starting every task.

4.1 Design of tasks

The tasks are designed on the basis of complexity starting from easy to hard. Sort and shape tasks are the easiest tasks as there is no order of placement that the object needs to follow. The remaining three tasks follow the order in which objects should be placed, so the user has to remember the order otherwise system gives the error message to the user and in the last task the objects should be placed in a manner to keep the balance of the stack otherwise it will fall down. Further description of each task is described in the next section. The figure 4-1 shows the initial stage of the robot and the objects before starting each task.

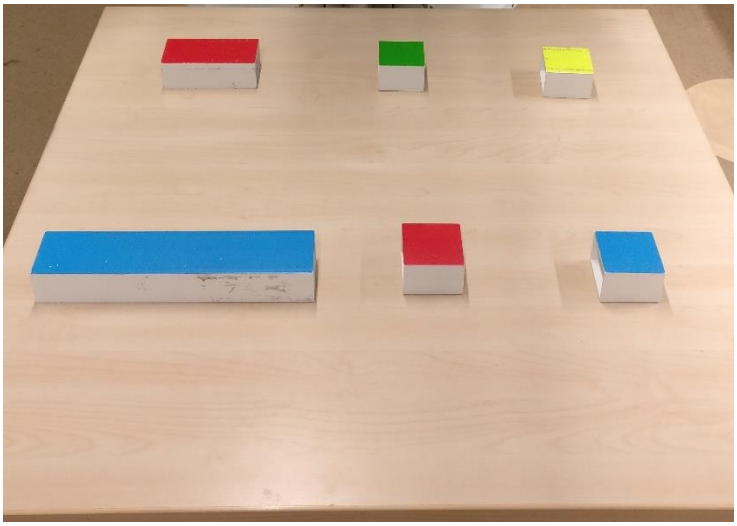

Figure 4-1: Initial position of the blocks for all tasks.

4.1.1 Sort

The goal of this task is to group the objects according to their colors, neither the sequence of grouping the specific color nor the size of the objects matters that is Red color objects can be grouped earlier than the blue objects and similarly large object can be placed earlier than the small object and vice versa. The task is considered to be completed successfully if all the objects are grouped at the end (see figure 4-2).

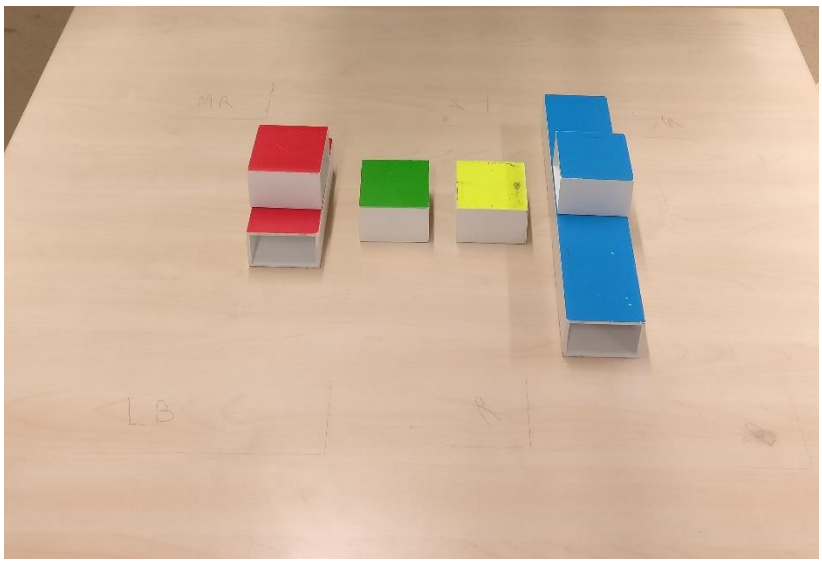

Figure 4-2: Final state of the sort task. The robot sort and place all the blocks according to their color.

4.1.2 Stack

The goal of this task is to stack objects according to their sizes, the sequence of sorting the specific color first does not matter but the size of the objects do, that is, Red color objects can be stacked earlier than the blue object and vice versa but the large object must be placed first then comes the medium and then the small object. The task is considered to be completed successfully if all the objects are stacked correctly (see figure 4-3).

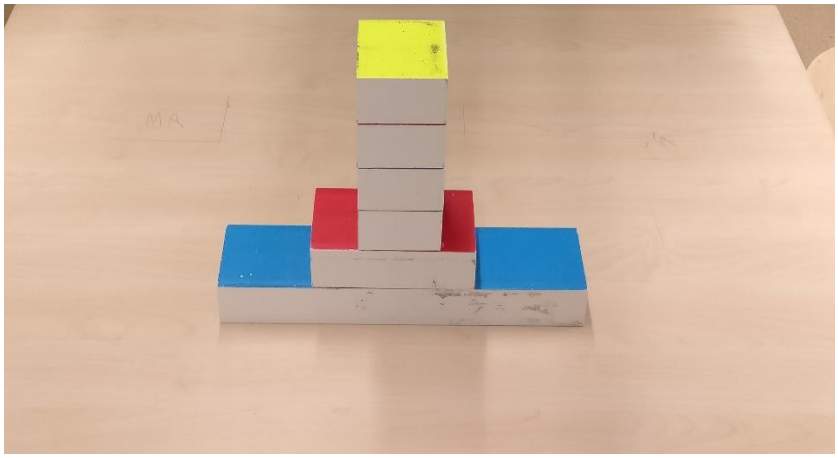

Figure 4-3: Final state of the stack task. The robot placed the blocks according to the specified order.

4.1.3 Build

The goal of this task is to build a bridge of the objects. The selection of objects is not dependent on the colors and the sizes but the pattern of the tower. The task is considered to be completed successfully if the tower is created correctly. The figure 4-4 shows the final state of this task.

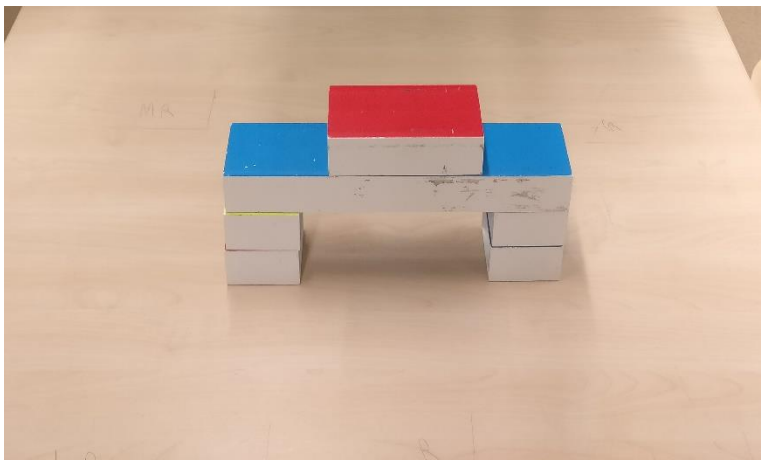

Figure 4-4: Final state of the build task. The robot placed all the blocks according to the specified order to construct the bridge.

4.1.4 Balance

The goal of this task is to place the objects on top of each other in such a manner that it requires the balancing of the previous object. If any of the objects are not placed properly and it is out of balance then it will fall down which results in the failure of the task. The selection and placement of the objects are required to follow a specific pattern in order to complete the task successfully (see figure 4-5).

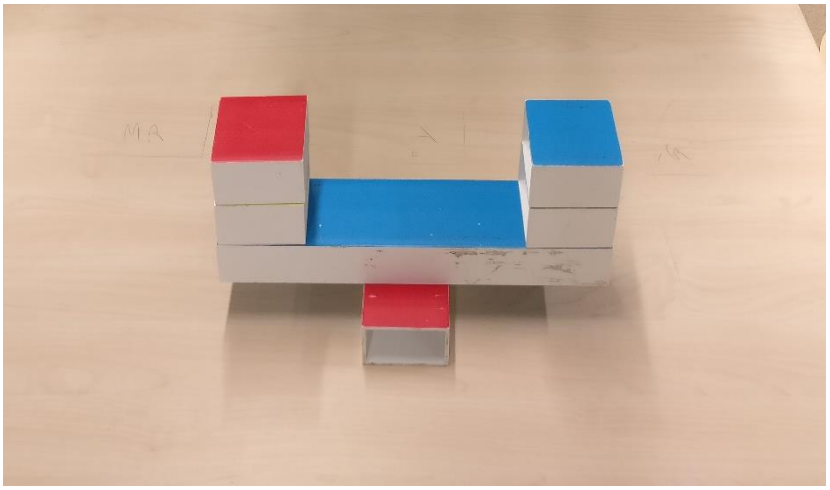

Figure 4-5: Final state of the balance task. The robot placed the blocks in such a manner that the stack remains balanced.

4.1.5 Shape

The goal of this task is to place the objects in such a manner that they will create a square. They can create any shape like a triangle, circle, etc. but we have chosen a square shape in order to use all six objects. The selection and placement of objects are not required to follow a specific pattern to complete the task successfully. The figure 4-6 illustrates the final state of this task.

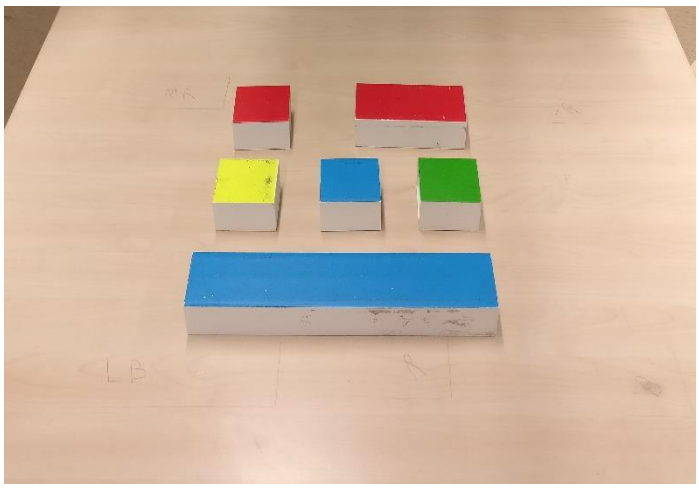

Figure 4-6: Final state of the shape task. The robot placed the blocks at specific positions to create the square shape.

4.2 Human-robot interaction

The system has three modes of operation, that is, human commands, autonomous and robot commands. Human commands mode requires that the user should provide all the commands one by one however in autonomous mode, user just has to specify the name of the task and the robot will place all the objects accordingly. In the robot commands mode, the robot starts the task and ask the human to perform the required operation. After receiving the command from the user, the robot executes it and perform the desired operation (see figure 4-7).

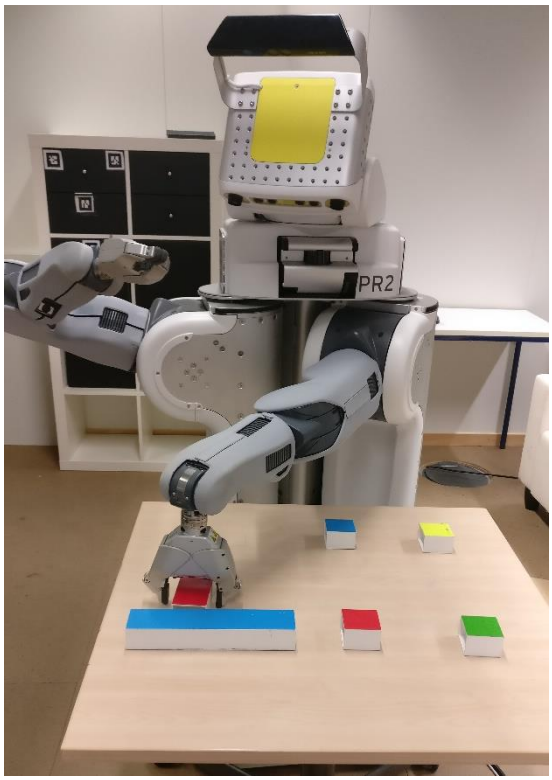

Figure 4-7: The robot performing the operation.

4.2.1 Human commands

In this mode system asks the user to provide all the commands one by one. The speech manager receives the command and delivers it to the task manager. The task manager analyze the instruction, perceive the current state and sends the instruction to the robot. After that, the robot executes the instruction, sends the feedback and wait for the next instruction.

4.2.2 Autonomous

In this mode, the system asks the user to provide the name of the task to be performed. Then this information is delivered to the robot. The robot places all the objects on their final positions and informs the user when the task is completed.

4.2.3 Robot commands

In this mode, the robot starts the task and inform the user which task is started. Robot perform the operation on its turn and then ask the user to perform the operation. The robot waits for its turn until the user finishes the operation. When all the objects placed on their final positions, the robot inform the user that the task is completed.

5 Implementation

This chapter describes in detail the implementation of the design of the system discussed in chapter 3 along with the execution of the tasks discussed in chapter 4. It further describes the algorithms used to solve the problem discussed at the beginning of the thesis. It also includes the pseudo code along with the description which helps in better understanding of the implemented algorithm.

5.1 Speech interface

A software is developed using the Python language which enables the PR2 to receive the speech commands, translation, mapping, providing feedback and sending instructions. Any other programming language can also be used like Java, C++ etc. but we select Python language because it is simple, easy to use for speech recognition and required less coding as compared to other languages. The figure 5-1 (left) shows the screenshot of the working of the speech manager. It informs the user with the keyword “say…” when it started listening. Then the received command converted into the corresponding value, here for example, “small green” is converted into 2, and sends to the robot, figure 5-1 (right). The robot send the feedback, here for example “ok”, and ask for the next command.

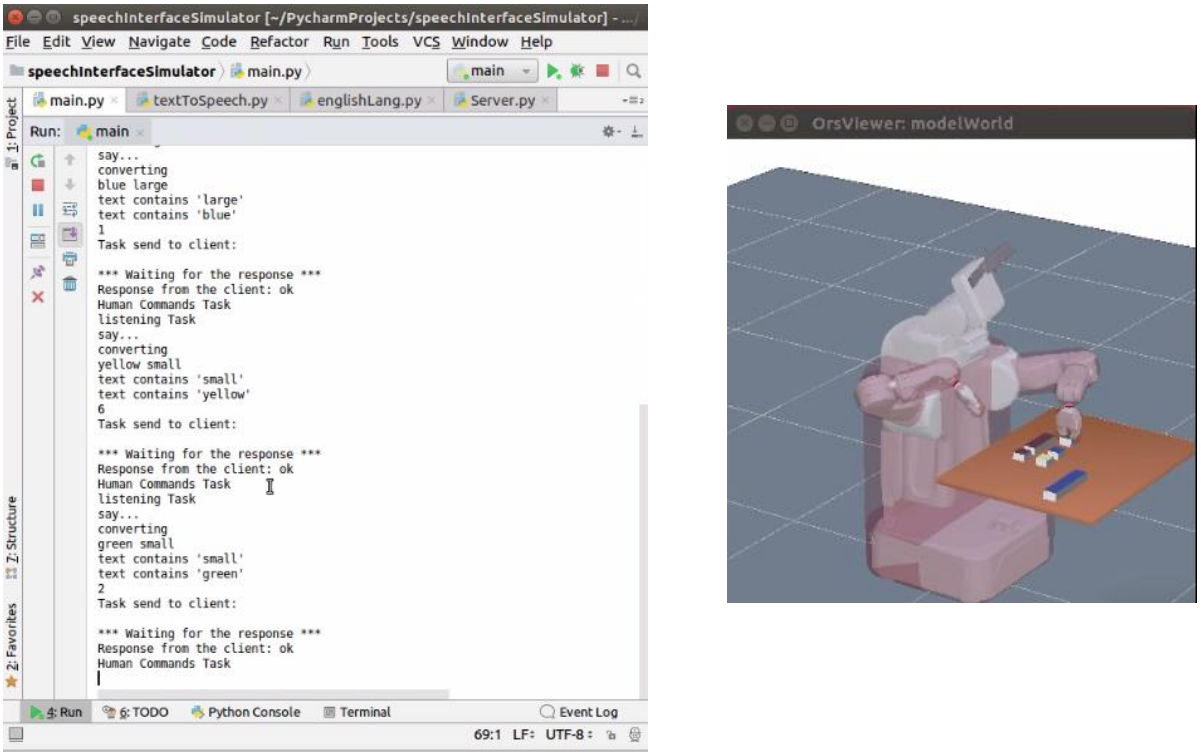

Figure 5-1: Working of speech interface (left), the simulated model of the robot executing the command (right).
The speech manager ask for the next instruction from the user. Then it is converted and delivers to the robot. The robot sends the feedback after executing the instruction which converts into an audio signal for the user.

5.1.1 Exchange of commands

Main class executes at the beginning which is responsible for executing the commands sequentially, monitoring, transmitting and receiving the data to and from the client in order to perform the operation. For example, if the user selects the German language for interaction then it keeps track of it and executes only the functions within the class refers to the German language. At the beginning of the program, the server creates a server-side TCP socket (in Python) and wait for the client (in C++) to become active and create client-side TCP socket. After the creation, a client sends the signal to the server and wait for the response. Server

accept the client and sends back the acknowledgment, at the reception of the acknowledgment, a successful connection has been established between the two and they are ready to send and receive the data. The server acts as a speech manager and controls all the listening and speaking tasks. After receiving the command from the user it converts it into text and analyzes it. If the command is clear and contains all the information which speak manager is expecting then it either execute the next instruction or maps it to its corresponding value to send it to the client. If any required information is missing then it initiates human-robot dialogue in order to resolve ambiguity and ask for the missing information, then it sends the command to the client and waits for the response. The algorithm 5-1 shows the pseudo code of the speech manager, at line no. 5 it asks the user to select the language. Then it asks the user to select the mode and task at line no. 9 and 12. The line no. 14 shows how it gets back the feedback from the robot.

5.1.2 Connection to the client

We have created a TCP connection to the client which uses the local host and port no 1500. The TCP connection is used because it sends the acknowledgment after the reception of every packet to the sender and sends the request to resend the packet if it get lost in between (see figure 5-2).

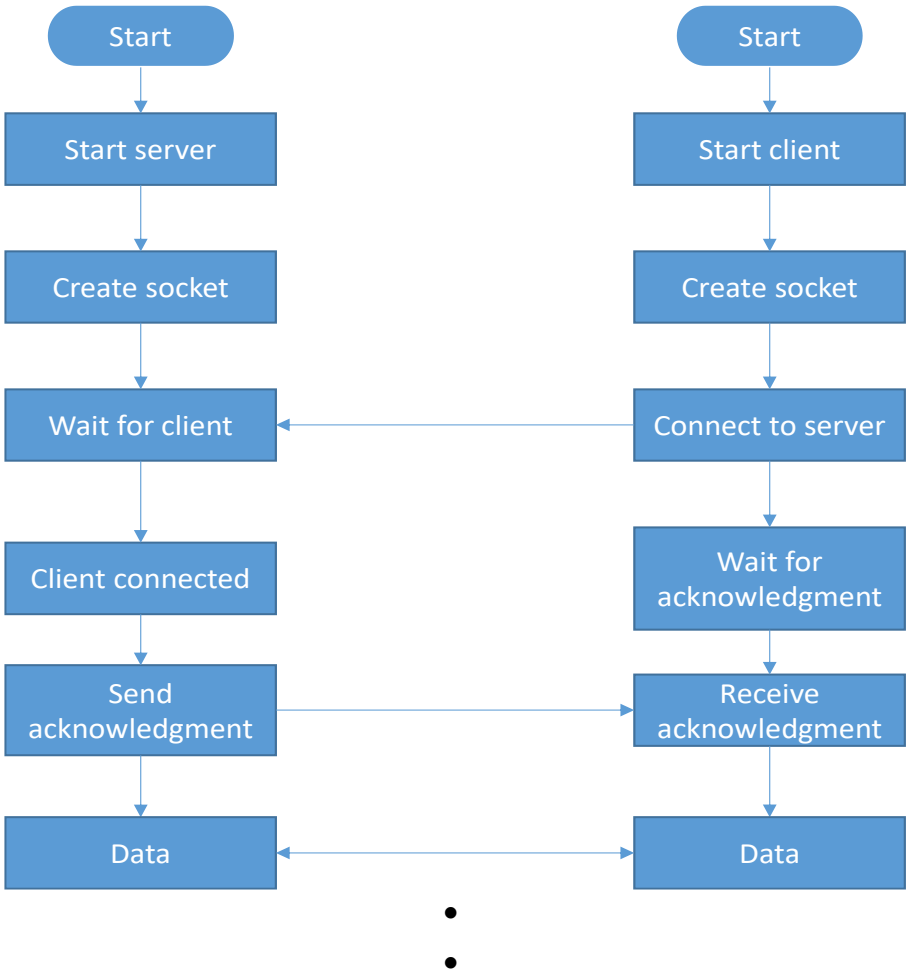

Figure 5-2: A one-to-one TCP connection between the server and client.

```
1.  BEGIN
2.      // Creating the connection with the client
3.      connection ← Server.createSocket()
4.      // Selecting the language
5.      selectLanguage ← langSelection.langselect()
6.      IF (selectedLanguage = “English”) DO
7.          WHILE true Do
8.              // selecting mode of operation
9.              mode ← englishLang.modeSelection()
10.             IF (mode = “Human Command”) DO
11.                 // selecting the task
12.                 task = englishLang.humanCommands()
13.                 IF (task = “Sort”) DO
14.                     Response ← Server.sendData(client, task)
15.                 ELSE IF (task = ”Shape) DO
16.                     Response ← Server.sendData(client, task)
17.                 .
18.                 .
19.                 .
20.                 END IF
21.                 IF (Response = “ok”) DO
22.                     textToSpeech.tts(“object placed”)
23.                 ELSE IF (Response = “error”) DO
24.                     textToSpeech.tts(“object not placed”)
25.                 ELSE IF (Response = “finish”) DO
26.                     textToSpeech.tts(“task completed”)
27.                 END IF
28.             ELSE IF (mode = “autonomous”) DO
29.                 .
30.                 .
31.                 .
32.             END IF
33.         END WHILE
34.     ELSE IF(selectedLangugae = “deutsche”) DO
35.         .
36.         .
37.         .
38.     END IF
39. END
```

Algorithm 5-1: Pseudo code of speech manager.

5.1.3 Speech to text conversion

The speech to text conversion requires to import "speech recognition" package which set up the input source to receive instructions and invoke speech recognizer. We set the sound signal threshold as 50 which helps the recognizer to determine the start and end of the speech signal (see figure 5-3). As the listening of the speech signal is blocking function so the listening timeout is set to 4 seconds in order to avoid the system to go into the infinite waiting state. Furthermore, the recognizer first adjusts itself according to the surrounding environment before taking the input command in order to reduce the noise as much as possible. For simplicity, this whole code snippet is wrapped in a function named "listen(reason)" which takes one parameter that tells why this function is called.

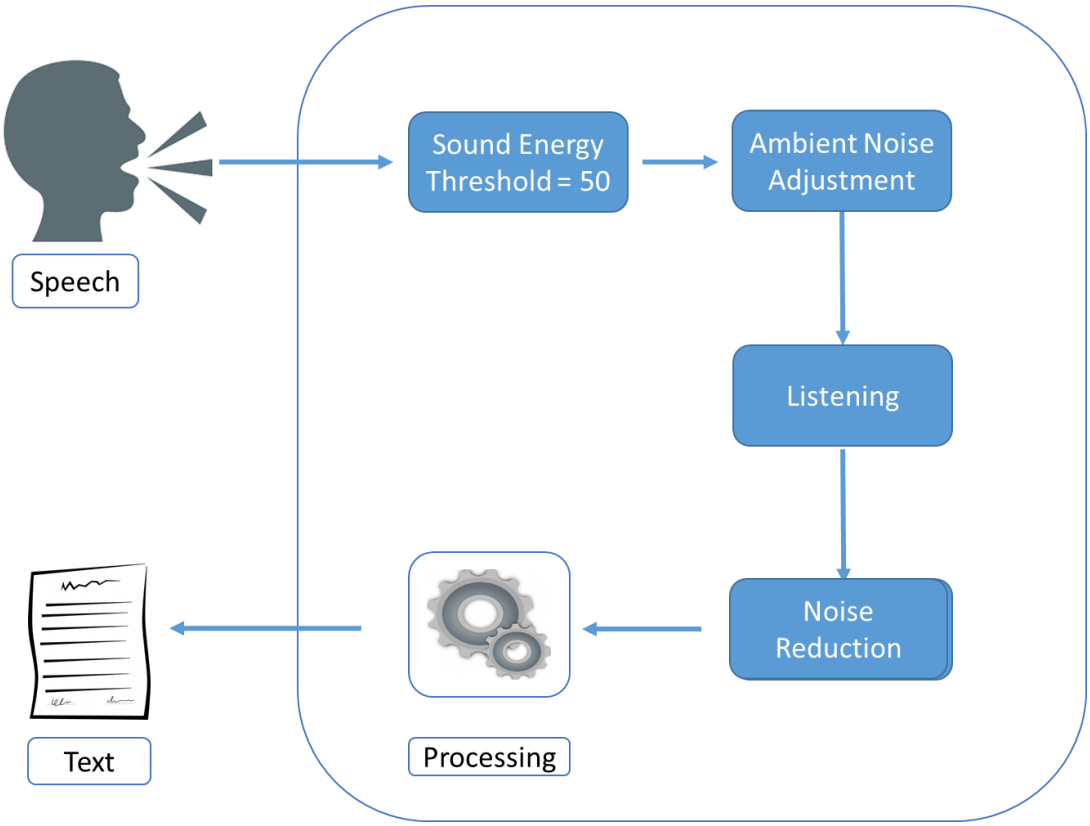

Figure 5-3: Speech to text conversion.
The figure shows the noise reduction methods which helps to improve the performance of speech to text conversion.

5.1.4 Text to speech conversion

The text to speech conversion requires to import pygame and gTTS package and any media player which is installed on the system and capable of playing audio files. We are using vlc media player for this purpose. The code snippet, shown below, gets the text input, process it and creates an mp3 file of that text input which is delivered to the vlc player so the user can hear the output signal. Then that .mp3 file is deleted. A separate class is created for text to speech conversion in order to create its instance whenever it is required instead of writing the whole code again and again. The class is created with the name “textToSpeech” so we just have to call it and pass the text as the parameter which we want the system to speak like textToSpeech.tts(“Hello world”). The figure 5-4 shows the graphical representation of the speech to text conversion process.

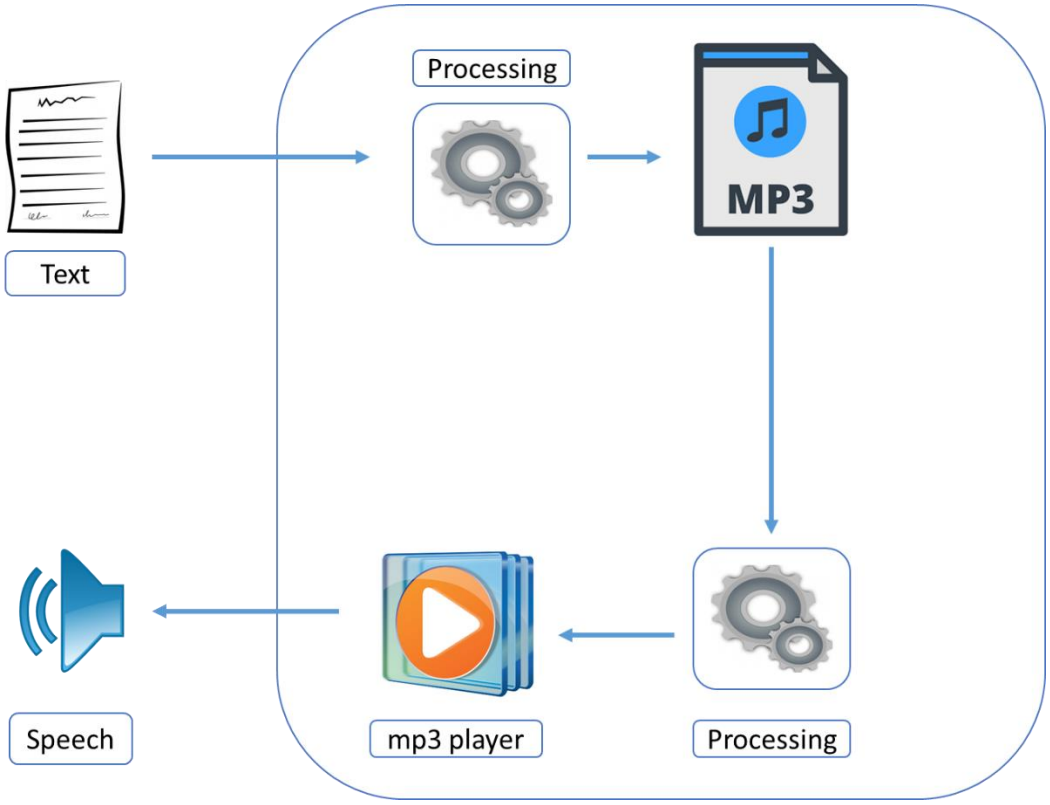

Figure 5-4: Text to speech conversion.
The system converts the text into an mp3 file format which requires the mp3 player to play it.

5.1.5 Language selection

In order to make the system robust, we give the opportunity to the user to select the desired language for interaction so the user feels more comfortable interacting with the robot. However, we have implemented two languages English and German but more languages can be added without much difficulty. Algorithm 5-2 represents the pseudo code of the language selection procedure. A separate class is created for the selection of the language in order to keep the code simple and easy to understand as much as possible. System asks the user to select the language in multiple languages so the user can understand and wait for the response, which can be seen at line numbers 2 and 3. If the user didn't give any response or if the response is not clear or if the selected language is not available then it will ask again. For the study, we used the English language. Line number 7 shows that the selected language is stored in the variable “languageInput”.

```
1.  BEGIN
2.      textToSpeech.tts("to select english language please say english after the beep", "en")
3.      textToSpeech.tts("Um die deutsche Sprache auszuwählen, sagen Sie bitte nach dem Piepton
                          Deutsch", de")
4.      beep ← pygame.mixer.Sound("beep.wav")
5.      beep.play()
6.      WHILE (languageInput = null) DO
7.          languageInput ← listen("language selection")
8.          IF (languageInput = "English") DO
9.                  selectedLanguage ← "en"
10.                 return selectedLanguage
11.         ELSE IF(languageInput = "Deutsche") DO
12.                 selectedLanguage ← "de"
13.                 return selectedLanguage
14.         END IF
15.         textToSpeech("select your language")
16.         languageInput = null
17.     END WHILE
18. END
```

Algorithm 5-2: Pseudocode of language selection

5.1.6 Mode and Task selection

The system offers two modes of interaction Human commands and Autonomous. The robot asks the user to select the desired mode in order to initiate the selected mode. Similar is the case with tasks, the system offers 5 tasks Sort, Shape, Stack, Bridge and Balance (see figure 5-5). The robot asks the user to select the desired task, to begin with. The selection of tasks followed by the mode selection. The study is performed in a manner that it starts in the human commands mode as it requires more human-robot interaction and tasks are started in a predefined sequence that is Sort, Shape, Stack, Bridge and Balance in the end.

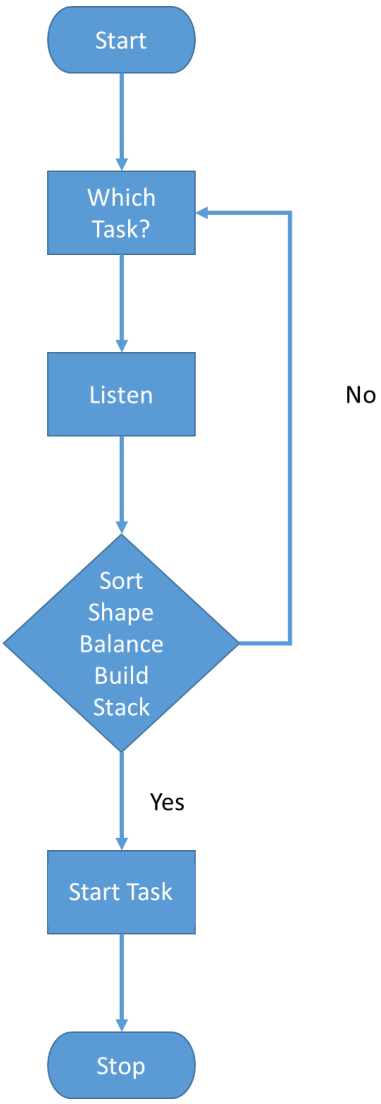

Figure 5-5: Task selection.
The task manager ask the user to select the desired task to perform.

5.1.7 Object selection

The algorithm 5-4 shows the pseudocode of the object selection process. The system initiates the dialogue system in order to resolve the ambiguity. First, it listens to the speech input and convert it into text and detect for the "size" and "color" as in our study we are using 6 objects of different sizes and colors so the system recognizes the specific object based on these two features. If the command is not understandable at all then system asks the user to rephrase the sentence otherwise it asks for the missing information only. For example, if the user said "place the object there" then this statement is very ambiguous as it is not clear which object? What is the color of the object and size? The system will process this command and cannot find the required information so it asks the user "Please specify the size of the object" then after this information system will ask the user "Please specify the color of the object", it can be seen at line number 10 and 27 respectively.

```
1.  BEGIN
2.      FUNCTION humanCommandsTask ()
3.          textToSpeech.tts("which object do you like to place)
4.          Task = listen(task)
5.          SentenceFlag ← false
6.          sizeFlag ← false
7.          colorFlag ← false
8.          IF(sizeFlag = false) DO
9.              WHILE (sizeFlag = false) DO
10.                 textToSpeech.tts("Please specify the size of the object")
11.                 task ← listen (size)
12.                 FOR a = 2 to 3
13.                     FOR b = size of array a
14.                         IF (task = array (a) (b)) DO
15.                             sizeFlag ← true
16.                             IF (a = 2) DO
17.                                 Size ← big
18.                             ELSEIF (a=3) DO
19.                                 Size ← small
20.                             END IF
21.                         END IF
22.                     END FOR
23.                 END FOR
24.             END WHILE
25.         ELSE IF(colorFlag = false) DO
26.             WHILE (colorFlag = false) DO
27.                 textToSpeech.tts("Please specify color of the object")
28.                 task ← listen (color)
29.                 FOR a = 4
30.                     FOR b = size of array a
31.                         IF (task = array (a) (b)) DO
32.                             colorFlag = true
33.                             IF (task = "blue") DO
34.                                 color ← blue
35.                             ELSEIF (task = green) DO
36.                                 color ← green
37.                             .
38.                             .
39.                             .
40.         END IF
41.         Return size and color
42.     END FUNCTION
43. END
```

Algorithm 5-3: Pseudocode of object selection

5.2 Text interface

The text interface is written in C++ language. It displays the information on the screen for the user to read and wait for the input command. It also works in a similar manner as the speech interface, the figure 5-6 (left) shows that the text interface is waiting for the instruction from the user. This instruction sends to the robot, figure 5-6 . The figure 5-7 shows the execution of the task. Initially, it asks the user to select the mode and then desired task to perform. After the selection of the task, the task manager has been invoked, which observe the objects and their positions. The task manager receives the input from the user and after analysis, determine whether the object is in the range of access which is set to 0.8 meters. Then it further analyzed the requested action along with the position of that object. If that object is already placed on the final position then it terminates the action and informs the user that object is already placed on the final position so requested action can't be performed, then asked for the next instruction. It keeps track of all the positions of the object and when all the objects are arranged then it informs the user that the task is completed successfully. For the study, human commands mode automatically selected and tasks are started in a sequence stated above in speech interface section.

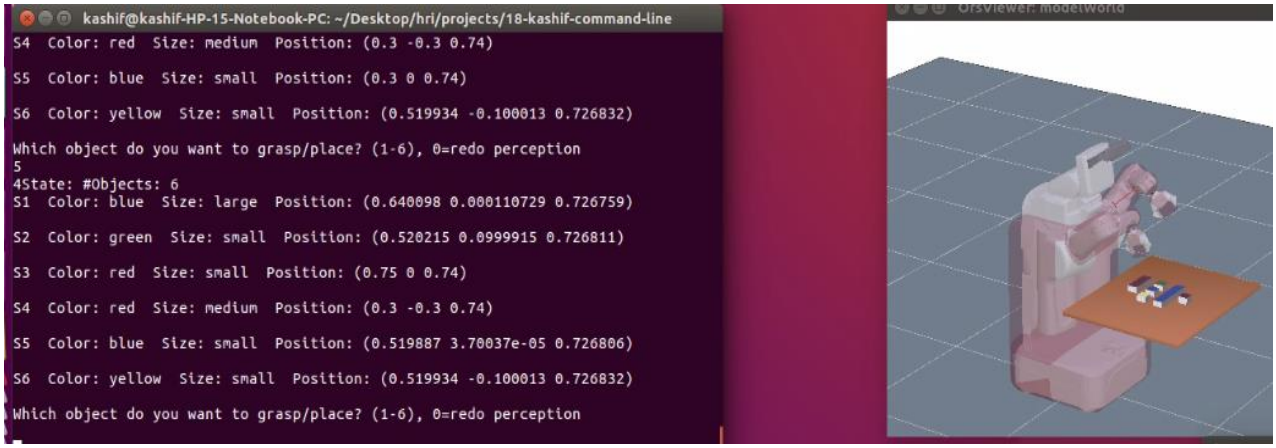

Figure 5-6: Text interface (left), the simulated model of the robot executing the command (right).
The text interface ask for the commands from the user and delivers it to the robot to perform the action. Then after receiving the feedback it asks for the next instruction.

5.2.1 Setup

The main class executes in the beginning when we run the program through the command line interface which controls the flow of the whole program, it compiles required libraries and packages required to operate the robot. After receiving the command from the user about which task needs to be started, followed by mode and task selection, it invokes the class of selected task which invokes HRI_state and HRI_task. HRI_state deals with the state of the objects. The names are assigned to the objects inside the function setByName(), whereas to check whether the object is placed on top of other object isAbove() is called, similarly isBelow() is called to check the object is below other object or not. PR2 gather the information of the object in the real world with the help of cameras placed on his face by invoking the onTableState() function. In order to update the status of the objects during execution, it uses updateFromPercepts(). The main loop of the program is in HRI_task class which keep executing performAction() function until the completion of the task, the algorithm 5-3 represents the pseudocode of this main loop. Once the task is completed the control is transferred back to the main class.

```
1.  BEGIN
2.          Input = 0;
3.          WHILE (input not equal to 10) DO
4.                  Do perception
5.                  Input = performAction()
6.          END WHILE
7.   END
```

Algorithm 5-4: Pseudocode of main loop to perform operation

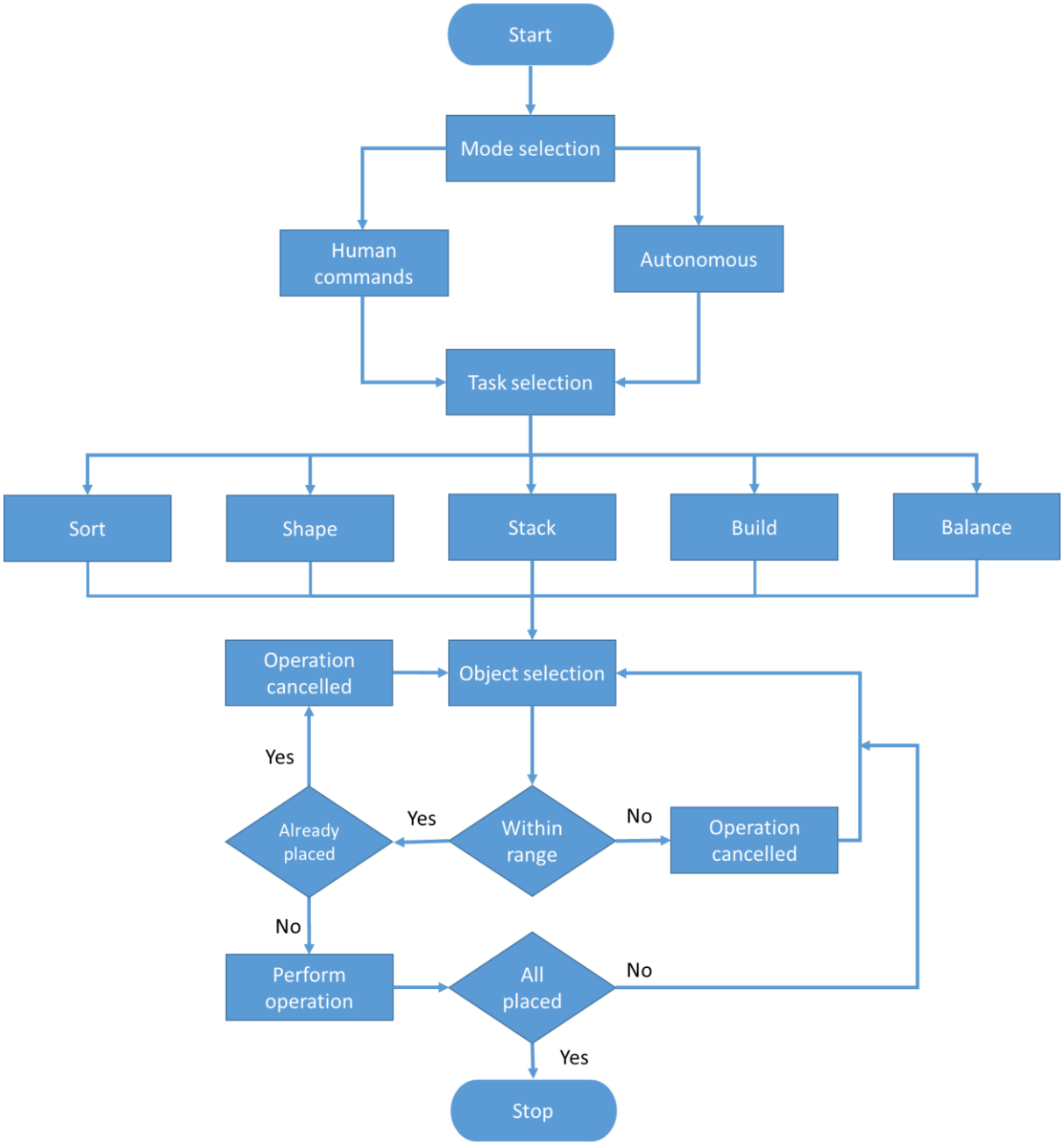

Figure 5-7: Execution of tasks
The simple block diagram shows the algorithm to execute the tasks. After selecting the mode and task, the task is started and each task is completed if all the objects are placed at correct position.

5.2.2 Task execution

At the beginning of the task, task manager check the state of each object on the table and ask the user to give the command. If the autonomous mode is selected then it just asks for the task to perform and if human commands mode is selected then it asks the user for each instruction. The figure 5-8 shows the execution of the task in detail shown in figure 5-7.

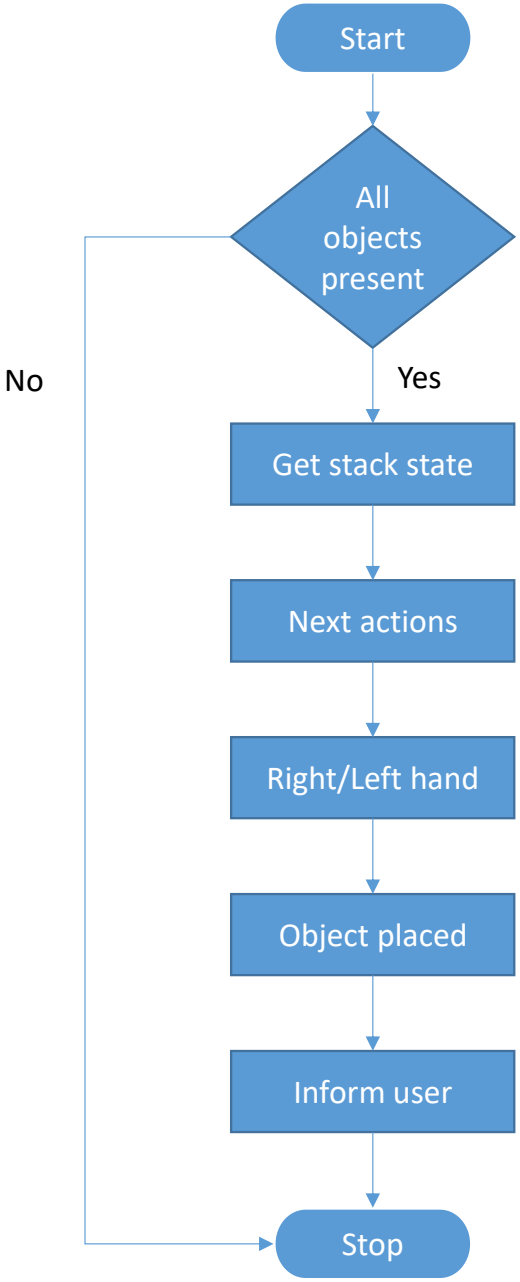

Figure 5-8: Task execution in detail
At the beginning of each task and after placing of each object, the system gets the current status of the objects then decides which hand should be used to perform the next operation.

In the study, we are using six objects so first, it checks whether all the objects are present on the table or not and make sure no duplicate object must be present, by calling the function isPlausible(), It returns true if there is no duplicate object otherwise returns false. After receiving the command the task manager calls the getStackState() function in order the determine the current state of the task, it checks the location of the objects and compares it with the final locations defined in the program. In order to determine the available next action, task manager executes nextActions() function which basically used the information received from

getStackState() and return back the next possible action that can be performed. Then task manager determines which hand should be moved in order to pick up the object and place in the right position by calling placeObj() function. After placing the object, it informs the user that the object is placed successfully.

5.3 Tasks

The tasks are sorted based on the difficulty level. Sort and Shape tasks are not ordered dependent, the user can select any object, so the task manager does not have to maintain the stack state. Whenever it receives the command to perform the action it just calls the nextActions() function to determine which next action is available. Rest of the three tasks are ordered based so task manager calls the getStackState() function to identify next available action.

5.3.1 Sort

In this task, the user can place any object in any order. It just keeps track of next available actions. The figure 5-9 illustrates the execution of the program.

5.3.2 Shape

This task is similar to the Sort task in execution. The figure 5-9 illustrates the execution of the program.

5.3.3 Stack

In this task, the task manager accepts the command in a defined order. The figure 5-10 illustrates the execution of the program

5.3.4 Build

This task is similar to the Stack task in execution. The figure 5-10 illustrates the execution of the program

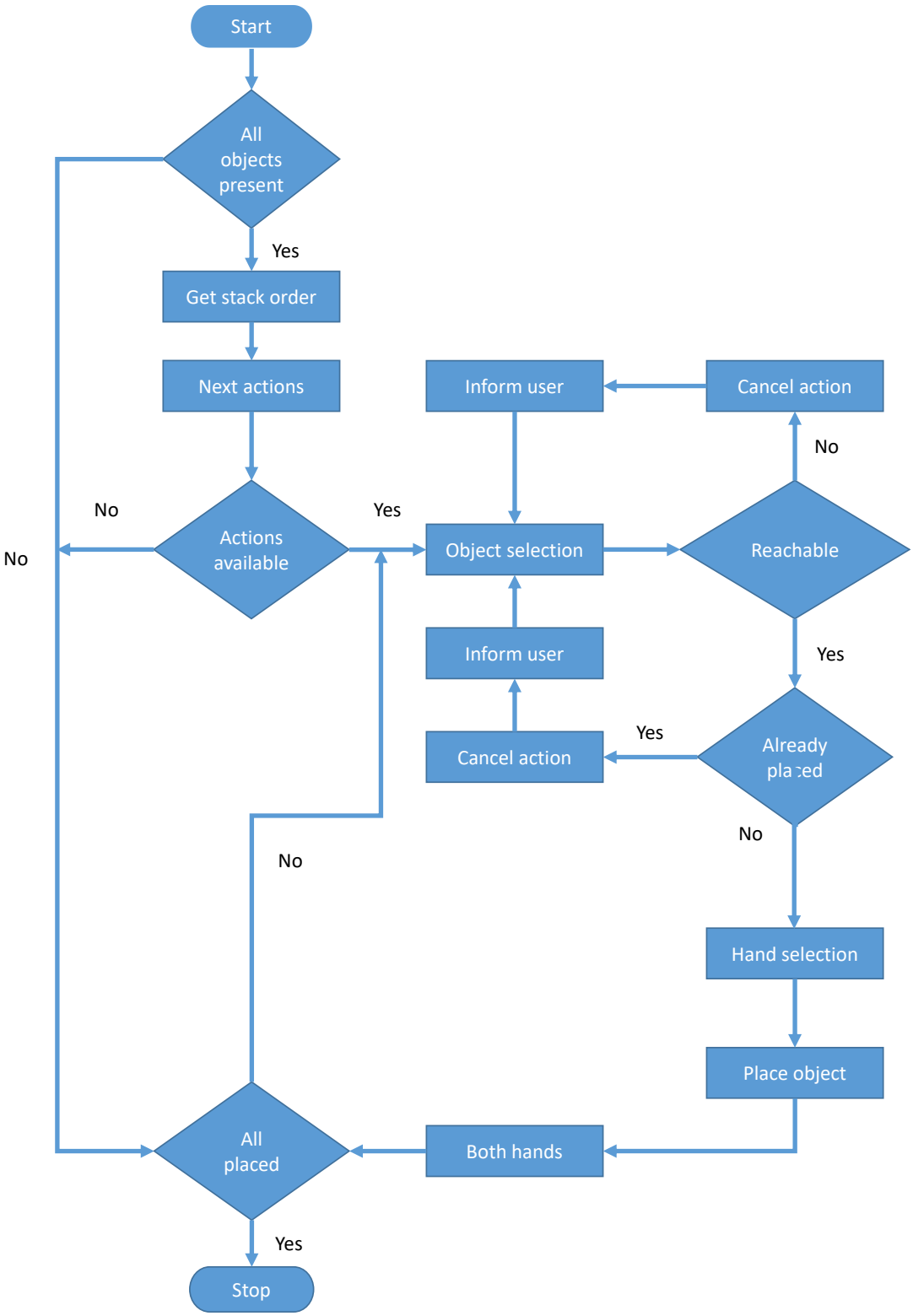

Figure 5-9: Execution of sort and shape task.
It can be seen that none of the task requires the user to follow any specific order.

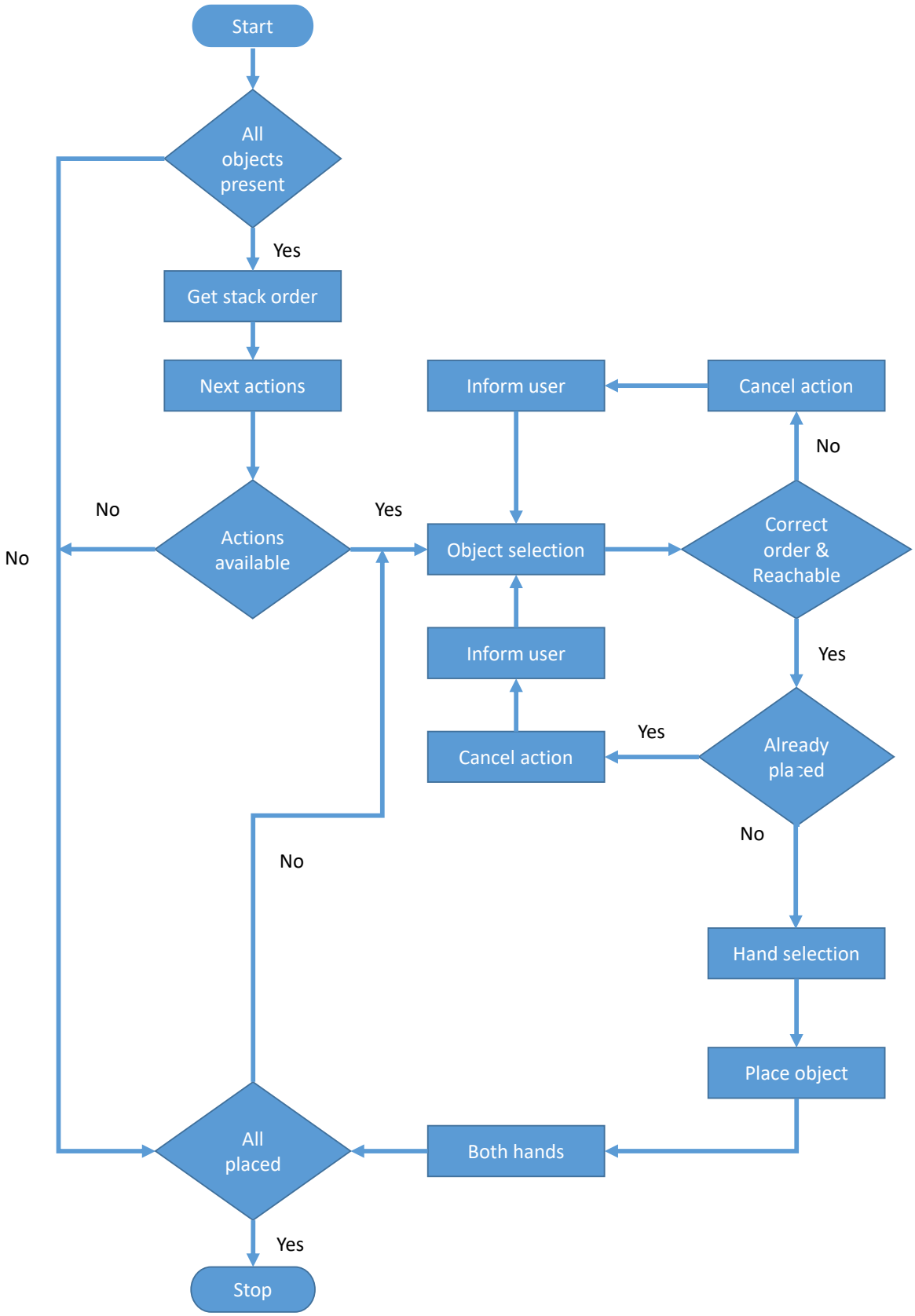

Figure 5-10: Execution of build and stack tasks
It can be seen from the figure that both of these tasks requires the user to follow specific order.

5.3.5 Balance

This task is a bit different from other tasks in a way that the robot must place the objects while maintaining the balance of the stack otherwise it falls down (see figure 5-11). So the robot asks the user for two objects, after placing two initial objects, so it can use both hands and place the objects simultaneously.

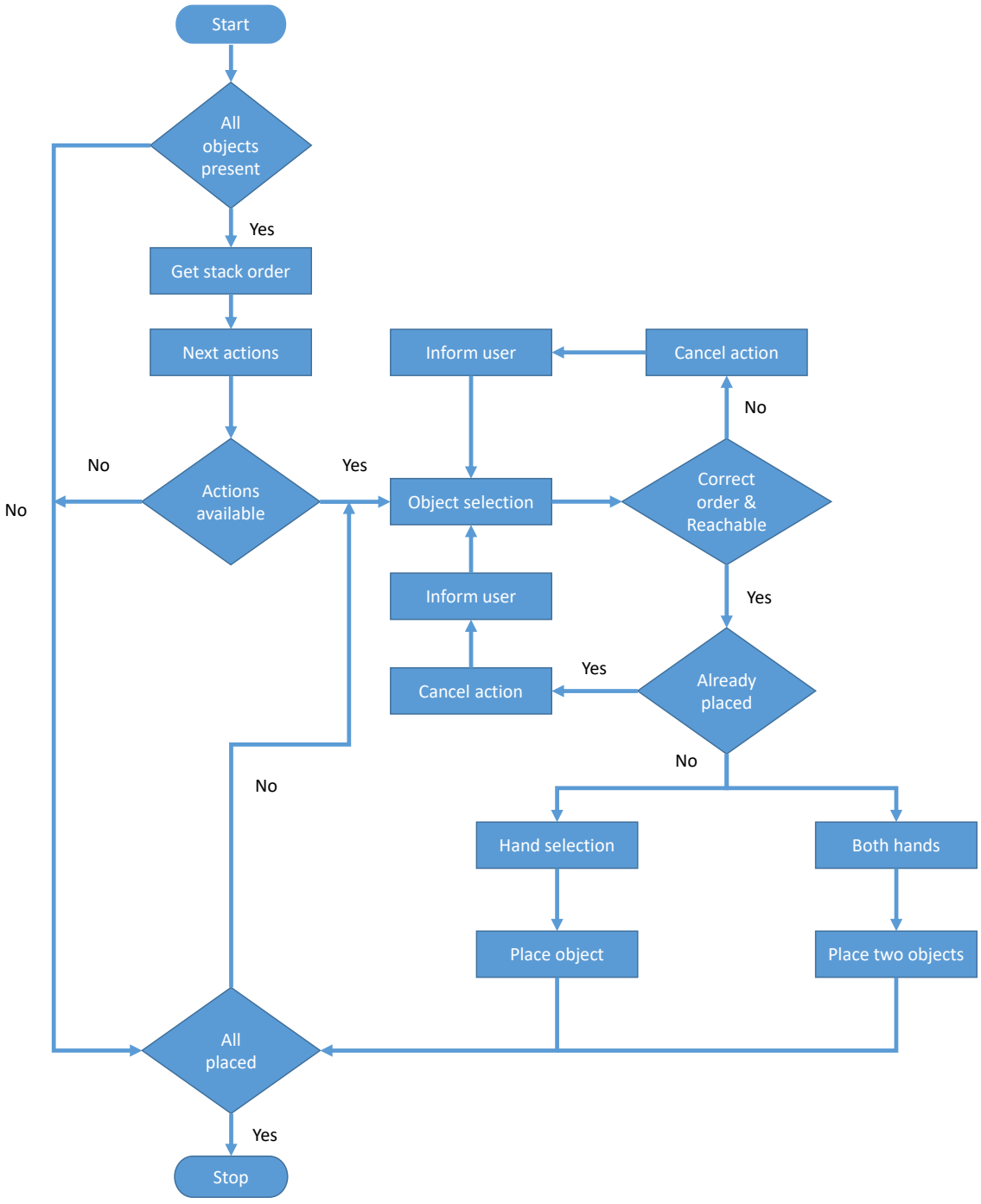

Figure 5-11: Execution of the balance task.
The robot follows the specific order to build the stack and maintain its balance.

6 User study

This chapter describes the results, evaluations and limitations of our system presented in the previous chapter, with the help of user study conducted with 12 participants. All the participants were students except for two. The study consists of the comparison between three modalities of interaction that is speech interface, graphical user interface and text input. The participants were divided into two groups of 6 people. For comparing speech and text input we have selected one group, that had 1 female and 5 males, belongs to the age group of 25 to 30 (Mean = 26.83, S.D = 1.94). For the comparison between speech and GUI, we have selected the second group that had 3 females and 3 males, belongs to the age group of 25 to 68 (Mean = 40, S.D = 19.48). In order to get better results we have changed the order of the interfaces between the participants. The first 3 participants used speech interface first while others used text input first. Similarly, we have changed the interfaces for the participants, for GUI vs speech. The user has to perform five tasks using the above-mentioned modalities. We have arranged the tasks based on the complexity levels, and they start automatically one after the other. We also present the number of errors they made and time taken by each participant to complete these tasks, all the results are shown graphically for better understanding and visualization.

6.1 Setup

We have conducted the study in the laboratory of machine learning and robotics department from the Institute of Parallel and Distributed Systems (IPVS) of the University of Stuttgart, where all the participants were invited. Initially, we explained the goal of the study and briefly explained all the tasks which have to be performed with the help of instructions sheet. We have also described important notes to keep in mind during the study. Then we handed over the consent form. The participants had to fill up the questionnaire after completing all the tasks. Participants were also asked whether they have already interacted with PR2 before or any other robot, some of them had never interact with the robot while others interacted less frequently. All the participants had to answer 6 statements for speech vs text-input and similarly for speech vs GUI:

I was more comfortable working with…
The interaction felt more natural working with...
The interactions were more fluent working with…
The interactions were more efficient working with…
The robot was a better partner with…
The task was easier with…
What were the good things about interacting with the robot using speech interface?
What were the bad things about interacting with the robot using speech interface?
Which modality do you prefer?
Do you have other comments about the study (robot, tasks, strategies etc.)

6.2 Results of Speech vs Text input

The results of the study are shown below, we have asked 8 qualitative questions from each participant along with their comments about the improvements and limitations of the system. The questions were related to measuring the quality of the system which includes efficiency, fluency, comfort, nature, ease, interaction and how intuitive the system was.

The Figure 6-1 shows how comfortable participants were while performing the tasks with both of these modalities, for Sort, Build and Balance tasks 3 out of 6 participants found speech interface more comfortable whereas for Shape task both modalities performed equally and text input was more suitable for Stack task. However, it can be seen from the column chart that no one felt comfortable with text input for Build task. The

reason for this might be that the user has to read the on-screen instructions in the text input interface in order to perform the task whereas in speech interface the system speaks the instructions for the user. For example in the speech interface, the system first provide the feedback to the user regarding the previous operation and then ask for the next operation. Figure 6-2 shows that how natural the human-robot interaction was for the participants. It can be seen from the chart that data is spread almost equally which shows that participants felt the interaction quite natural.

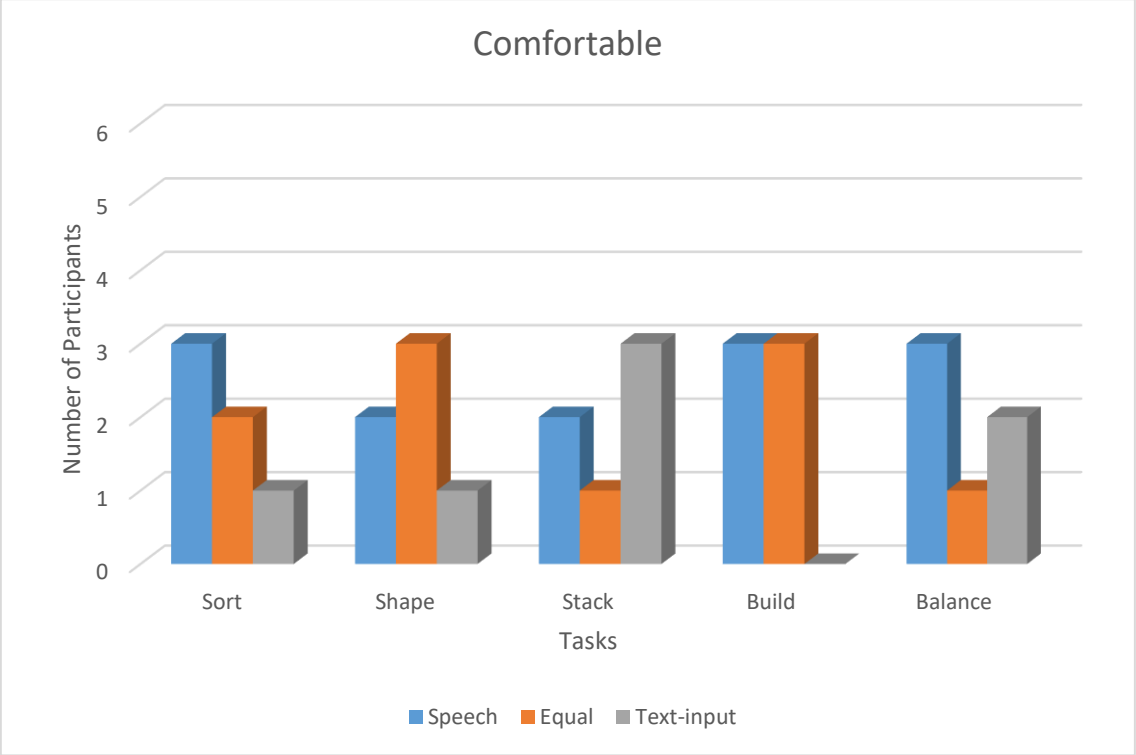

Figure 6-1: Comparison based on the user's comfort level (speech vs text).
The participants felt more comfortable in performing tasks using speech interface.

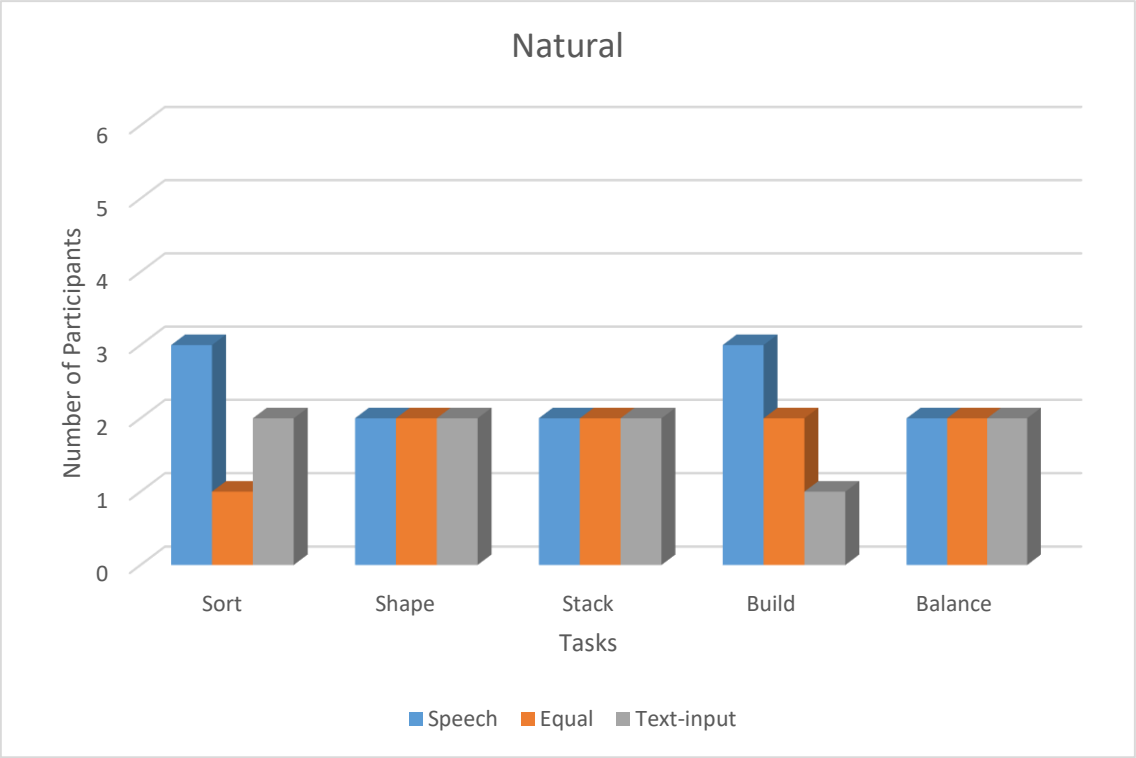

Figure 6-2: Natural or un-natural (speech vs text).
The interaction felt more natural with speech interface as the communication was in natural language.

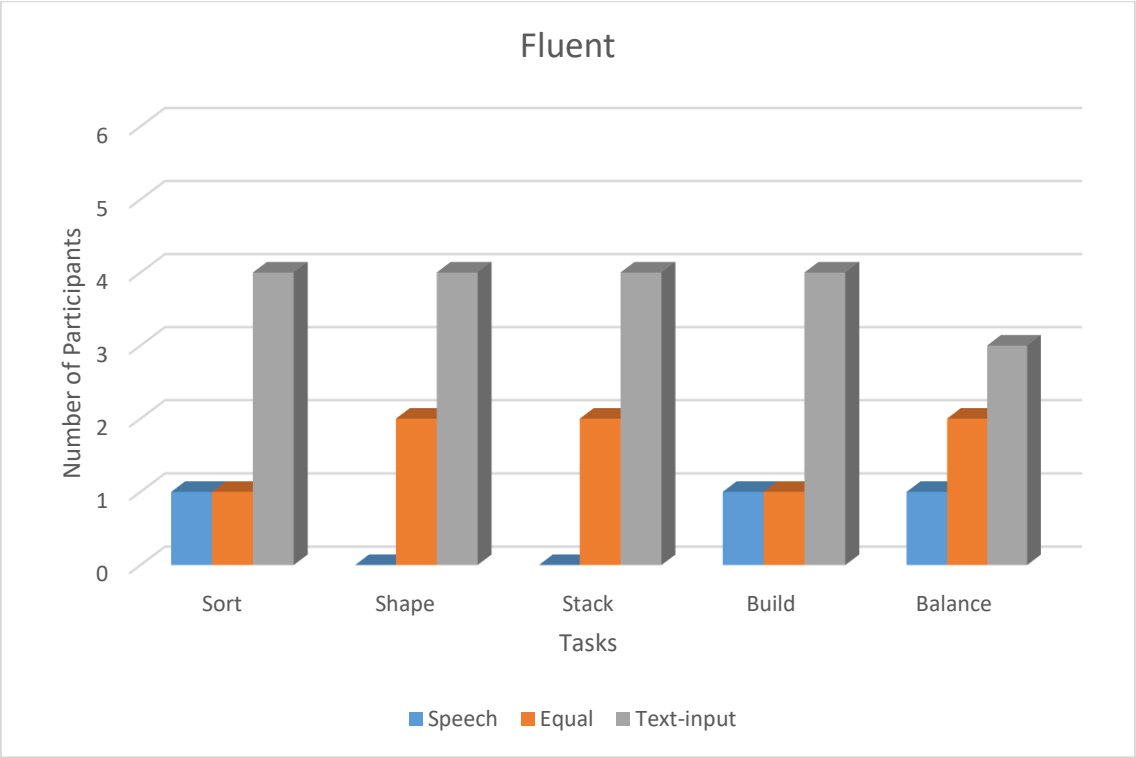

Figure 6-3: Comparison based on system's fluency (speech vs text).
The participants found text input more fluent than speech interface.

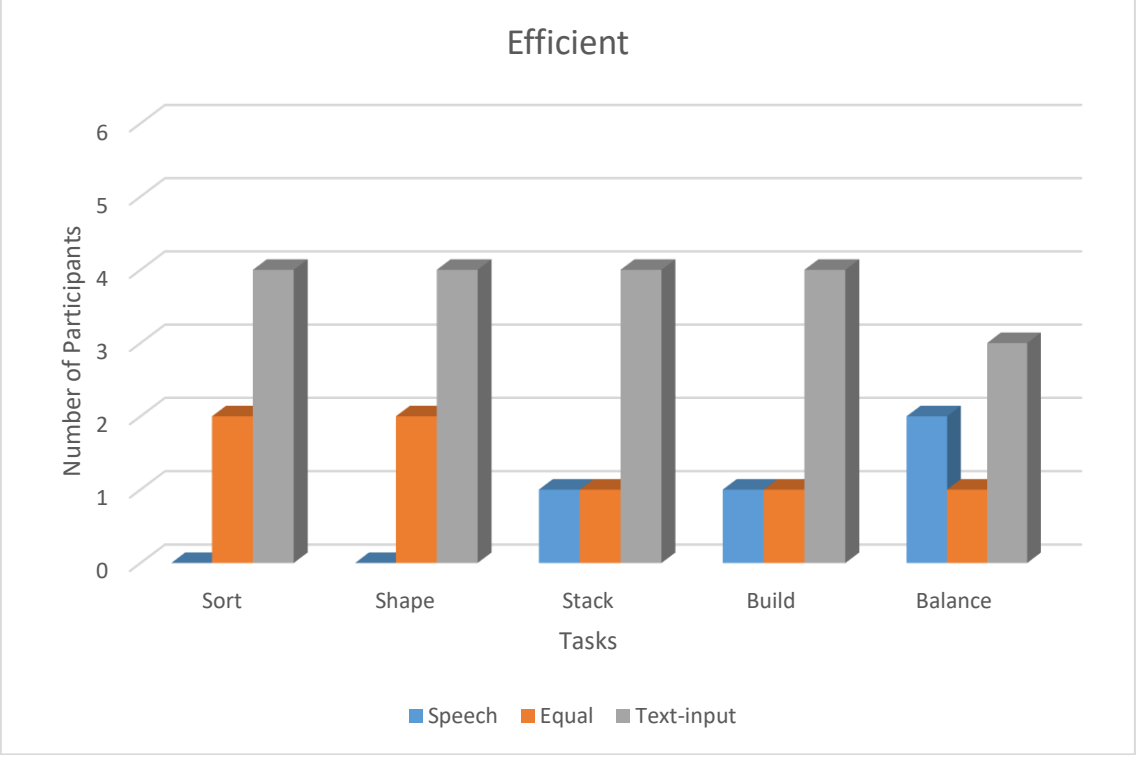

Figure 6-4: Comparison based on system's efficiency (speech vs text).
The participants found text input efficient than the speech interface due to the delay in speech to text conversion.

When it comes to fluency of the operations, it can be seen clearly from the figure 6-3 that text input is more fluent than speech interface as four of the six participants up-voted this interface for almost every task. The reason behind this is speech processing, translation, and mapping to the corresponding value of the operation. The speech to text conversion required the internet, so the speed of the internet connection also

contributes to this issue. The increase in the waiting time for speech processing made it less fluent than text input which can also be seen later in this chapter in the chart of task completion time in figure 6-7.

As discussed earlier in the above paragraph that where long processing time of speech recognition effects fluency it also affects the efficiency of the speech interface. Similar behavior of the participants can be seen in figure 6-4. Another additional factor which degraded the efficiency of speech interface is the noise in the signal which increased the error rate and sometimes participants had to repeat the same instruction again and again.

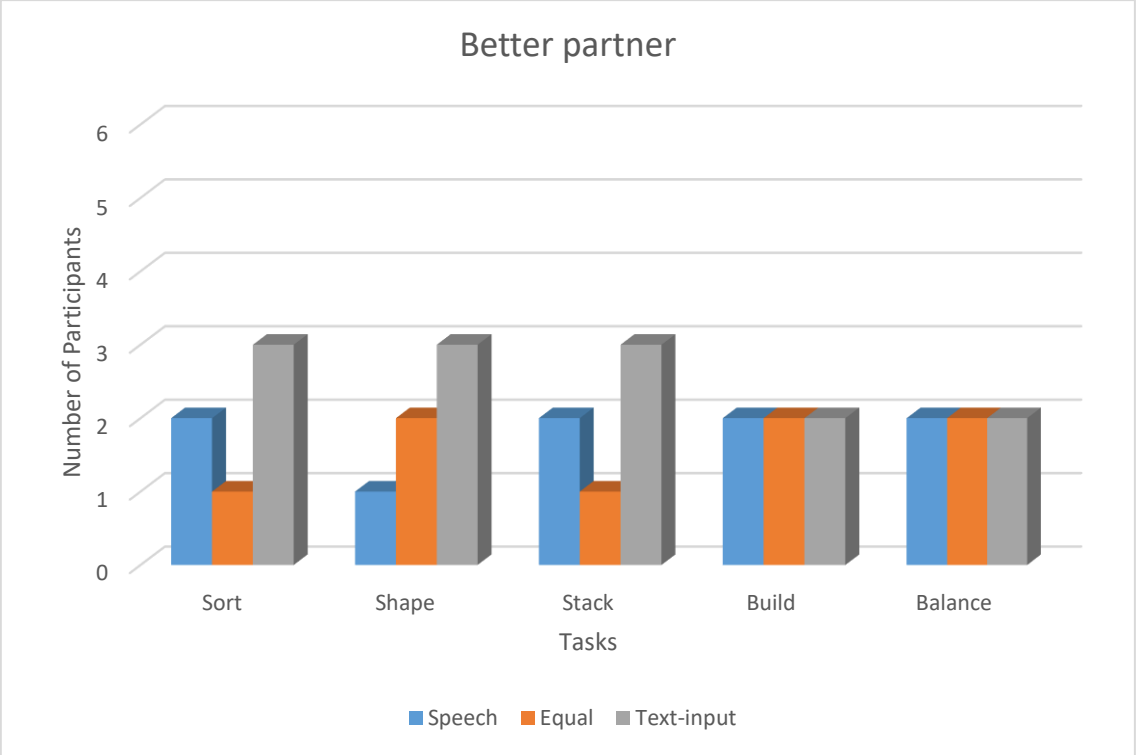

Figure 6-5: Comparison based on partnership in operation (speech vs text).

The figure 6-5 illustrates that participants support both speech interface and text interface almost equally in the partnership with the robot whereas figure 6-6 shows that most of the participants felt the interaction with the robot was easier with a speech interface. The reason behind this might be the audio feedback mechanism as humans usually prefer the communication in natural language.

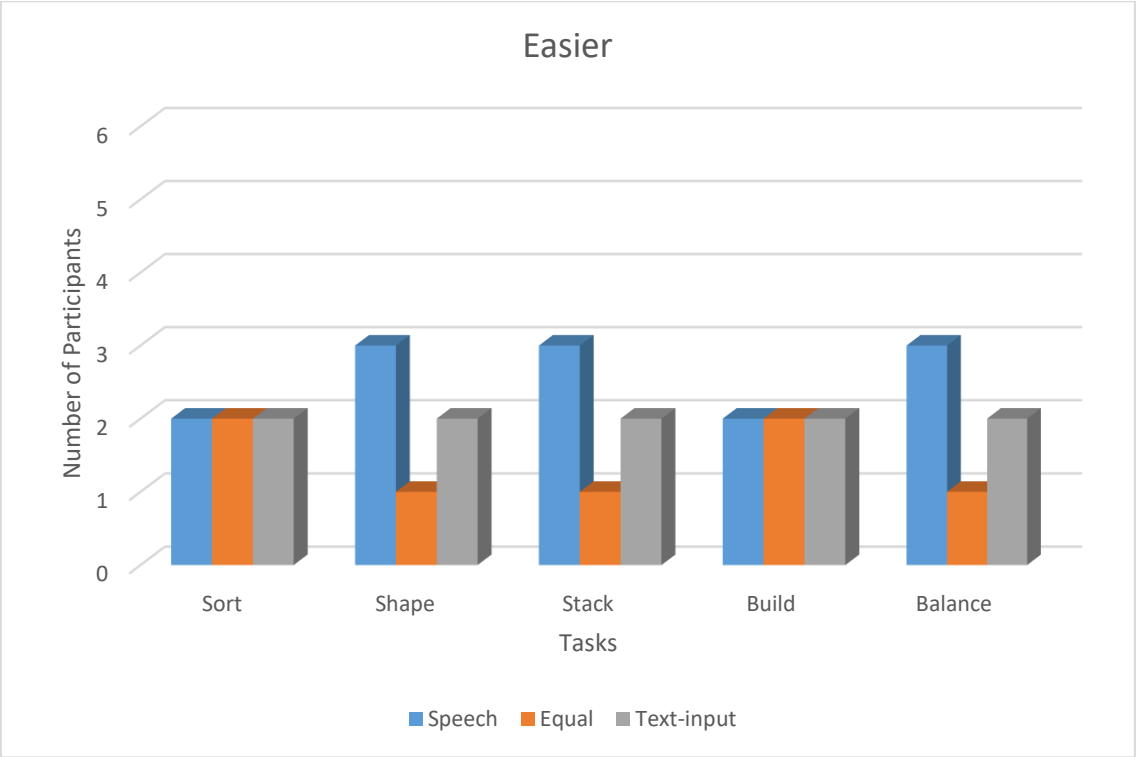

Figure 6-6: Comparison based on ease of human-robot interaction (speech vs text).
The participants found speech interface easier to use.

	Intuitive									
	Sort		Shape		Stack		Build		Balance	
	Mean	S.D	Mean	S.D	Mean	S.D	Mean	S.D	Mean	S.D
Speech	4.33	2.25	3.83	1.17	3.67	1.51	4.00	2.53	3.33	2.34
Text-input	4.67	1.86	4.50	1.87	4.67	1.86	5.50	0.55	3.67	2.88

Table 6-1: Intuitive (speech and text-input).
The table shows the mean and standard deviation of the ratings ranged from 0 to 6, where 0 represents strongly disagree and 6 represents strongly agree.

We have asked the participants whether they find the speech interface intuitive or not. The received results were quite spread that is some participants were strongly disagreed while others were strongly agreed. However, for text input, it can be said that most of the participants found it intuitive.

6.2.1 Tasks completion time and errors (speech vs text)

The figure 6-9 shows the comparison of the tasks completion time between speech interface and text input. It can be seen that speech interface took longer for each task but the differences are not that big. The performance of the system can be improved by reducing the delays in speech processing. One thing can be noticed from the chart that execution time of almost every next task reduced as compared to the previous task which shows the increment of the performance of the system as the user becomes familiar with it. The second figure, figure 6-10, shows the average of the total number of errors created by the users in both modalities, similarly the error rate reduced with the increase in the user experience. However, the error rate of the first two tasks is quite high for speech interface as compared to the text input, where surprisingly, there were no errors in shape task. The most common error in the balance task occurs when two objects need to be placed but most of the participant usually selected one object.

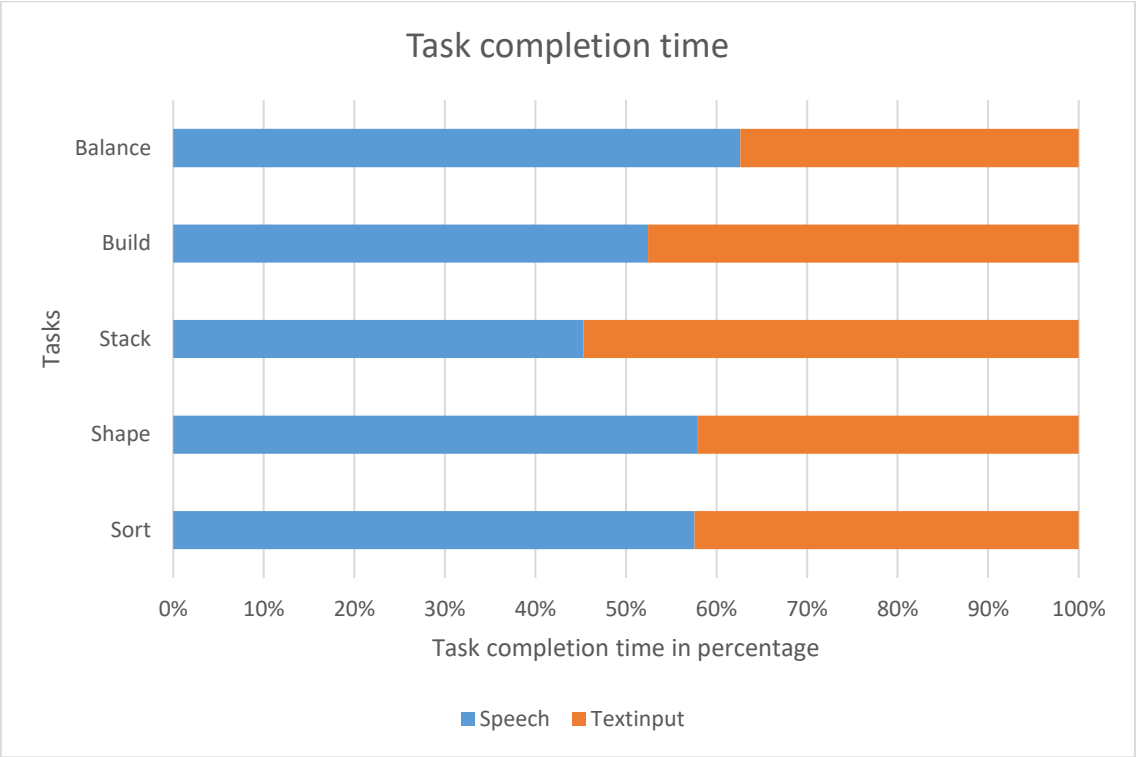

Figure 6-7: Task completion time (speech vs text).
The results show that speech interface took longer to complete the tasks because of the delay in speech to text conversion.

Figure 6-8: Comparison based on errors (speech vs text).
The frequency of errors was higher in speech interface as sometime it didn't recognize the commands and participants have to repeat it.

6.2.2 General comments (speech vs text)

The questionnaire designed for the participants also include the section for general comments and a few other questions. One participant mentioned that it would be better if the audio feedback resembles more to the human's natural sound instead of robotic sound which helps to make the interaction more natural. One common issue observed from the comments of the participants that sometimes speech recognizer didn't understand the command correctly so they have to repeat it which they don't like, secondly the delay in a speech to text conversion is something which needs to be reduced up to the extent that it becomes negligible. One of the participants suggests that the system should be tested for complex tasks where the robot should move from point A to point B. The participants preferred speech interface because it feels more natural and the interface was able to detect voice commands independent of the accent of the speaker (see figure 6-9). One problem noticed with the text input, that it did not supports multiple languages. If the user does not understand English language then it would be difficult for the user to operate the robot. The speech interface supports multiple languages, however, in this work only two languages, English and German, were implemented.

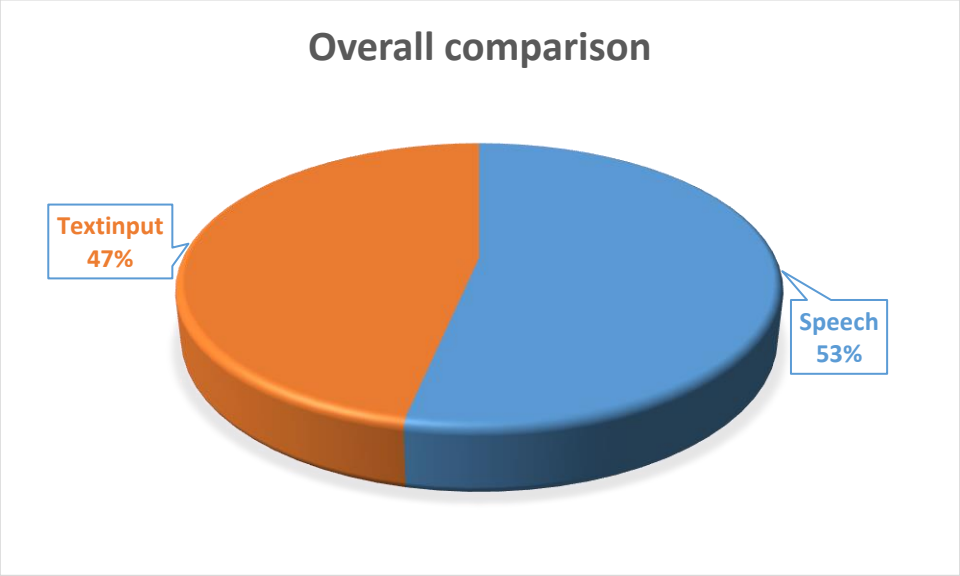

Figure 6-9: Overall comparison (Speech vs text).
The participants preferred speech interface because it feels more natural.

6.3 Results of GUI VS Speech

We have also conducted a study with another 6 participants to compare the speech interface with the graphical user interface, and the result of that study are presented below. We have asked the same set of questions, which we have asked to other participants.

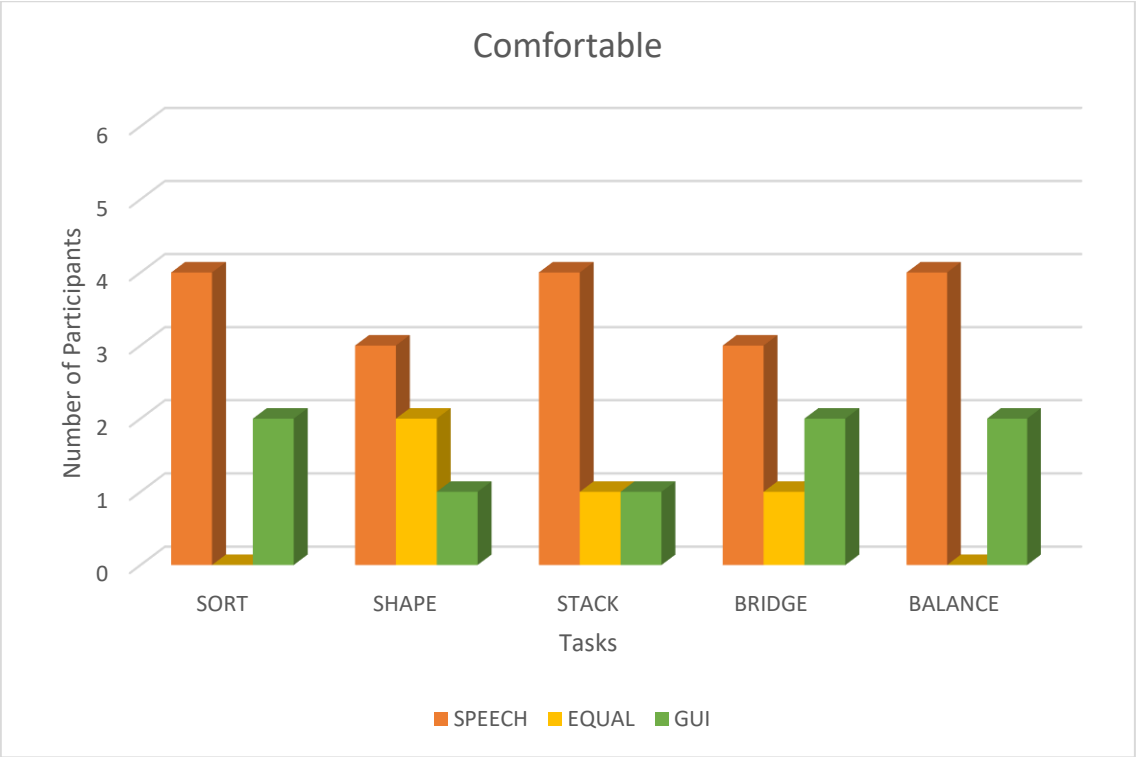

Figure 6-10: Comparison based on user's comfort level (speech vs GUI).
The results show the comparison between speech interface and graphical user interface based on the level of comfort of the user in completing the tasks. It can be seen that participants were more comfortable while using speech *interface.*

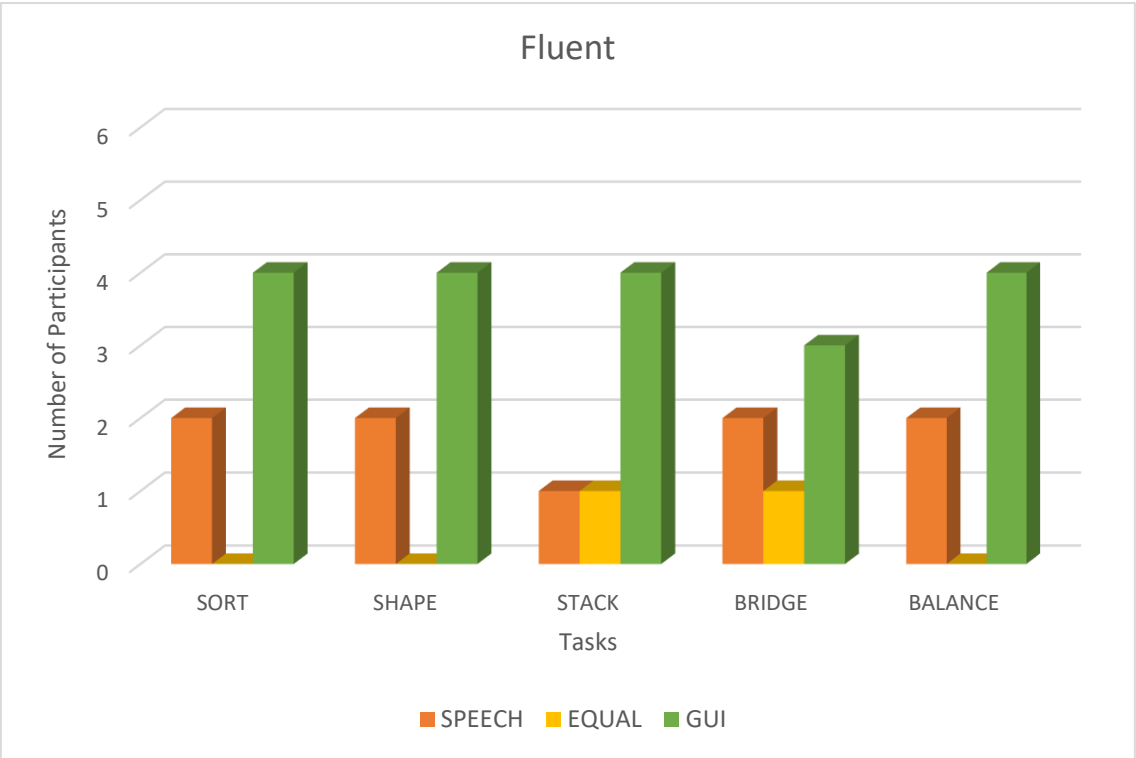

Figure 6-11: Comparison based on system's fluency (speech vs GUI).
The participants found GUI more fluent

Figure 6-12: Comparison based on system's efficiency (speech vs GUI).
The results shows that GUI was more efficient.

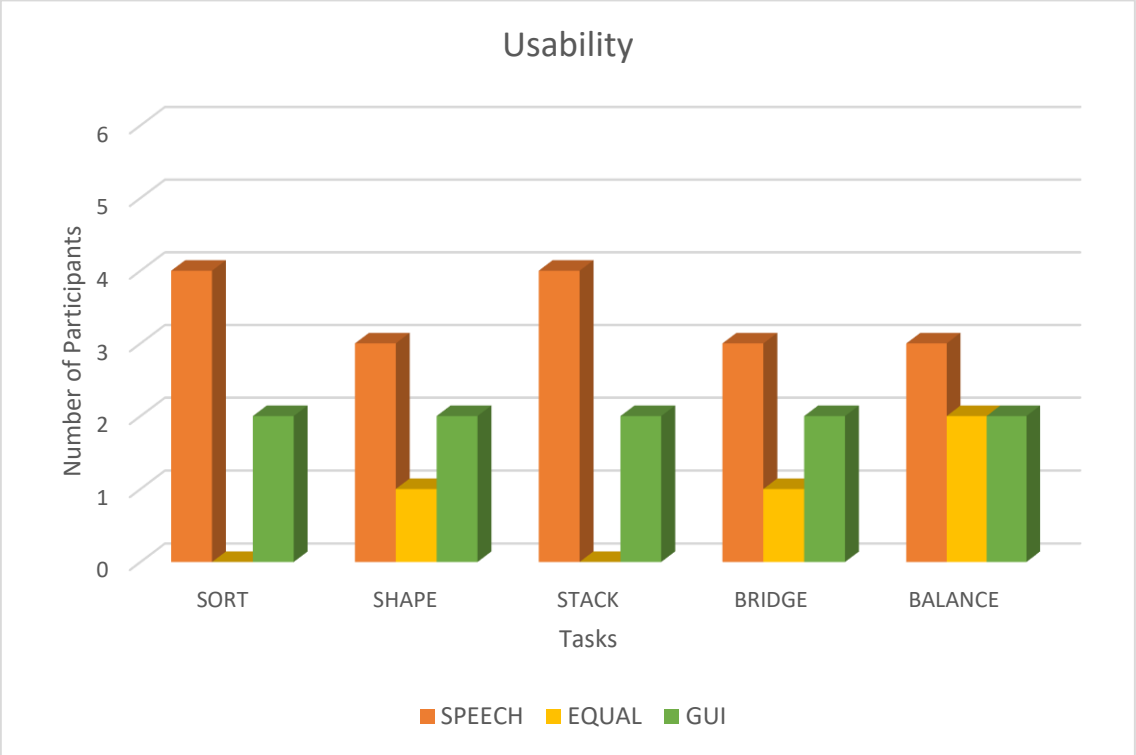

Figure 6-13: (speech vs GUI).
The participants found speech interface easier to use than GUI.

As discussed earlier that human feel more comfortable in communicating in natural language, it can be seen from figure 6-10 which shows that more participants were comfortable with speech interface. However, natural language processing is time consuming process which made it less fluent and efficient than graphical user interface (see figure 6-11 and 6-12) because the selected event can be processed very quickly as compared to speech signal. Furthermore, it can be seen from figure 6-13 that GUI turns out to be more

complex as users had to do additional stuff like read information from separate tab or closing the feedback pop-up again and again, whereas in speech interface they just had to speak in order to provide commands and listen the audio for feedback.

6.3.1 Tasks completion time and errors (speech vs GUI)

The figure 6-14 shows the comparison of the tasks completion time between speech interface and GUI. Similar results can be seen here, the delay in speech to text conversion results in greater execution time. The second figure, figure 6-15, shows the average of the total number of errors created by the users in both modalities, similar results can be seen here as well. Most of the participants made errors in first two tasks. In the GUI the participants just have to click the button and system executes that command. Whereas in the speech interface, the speech to text conversion was the biggest challenge and then the mapping of speech commands to the corresponding actions. These two factors increased the execution time and error rate.

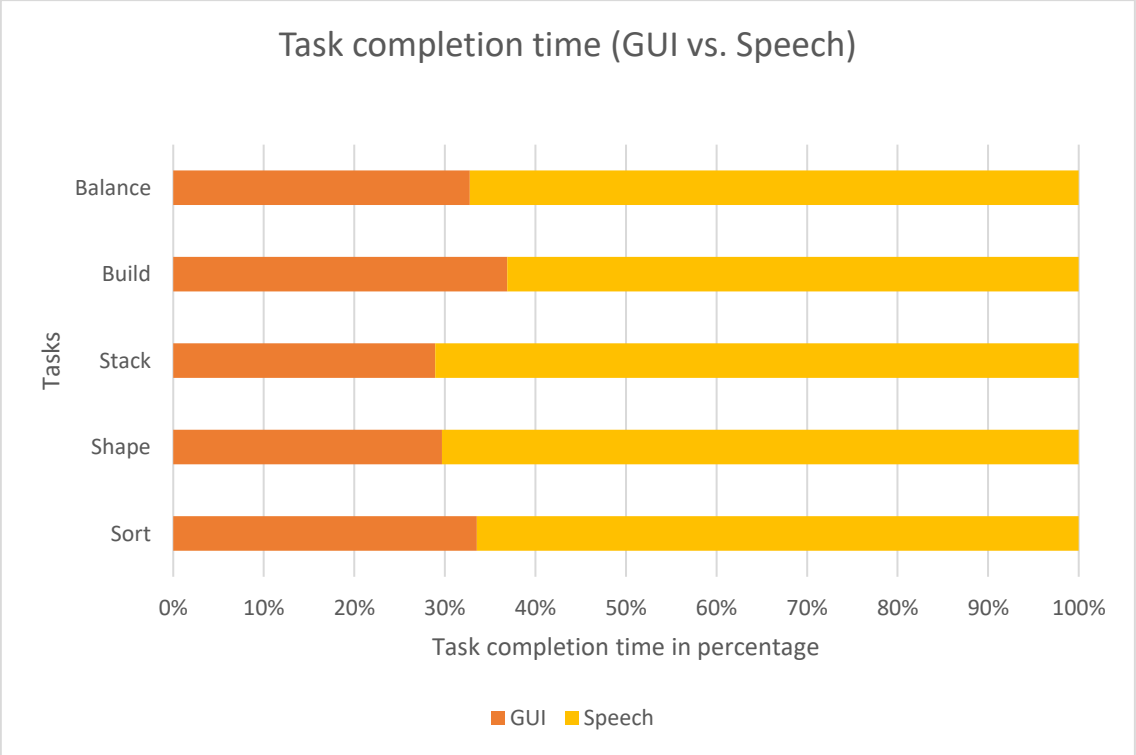

Figure 6-14: Task completion time (speech vs GUI).
The results show that speech interface took longer to complete the tasks because of the delay in speech to text conversion.

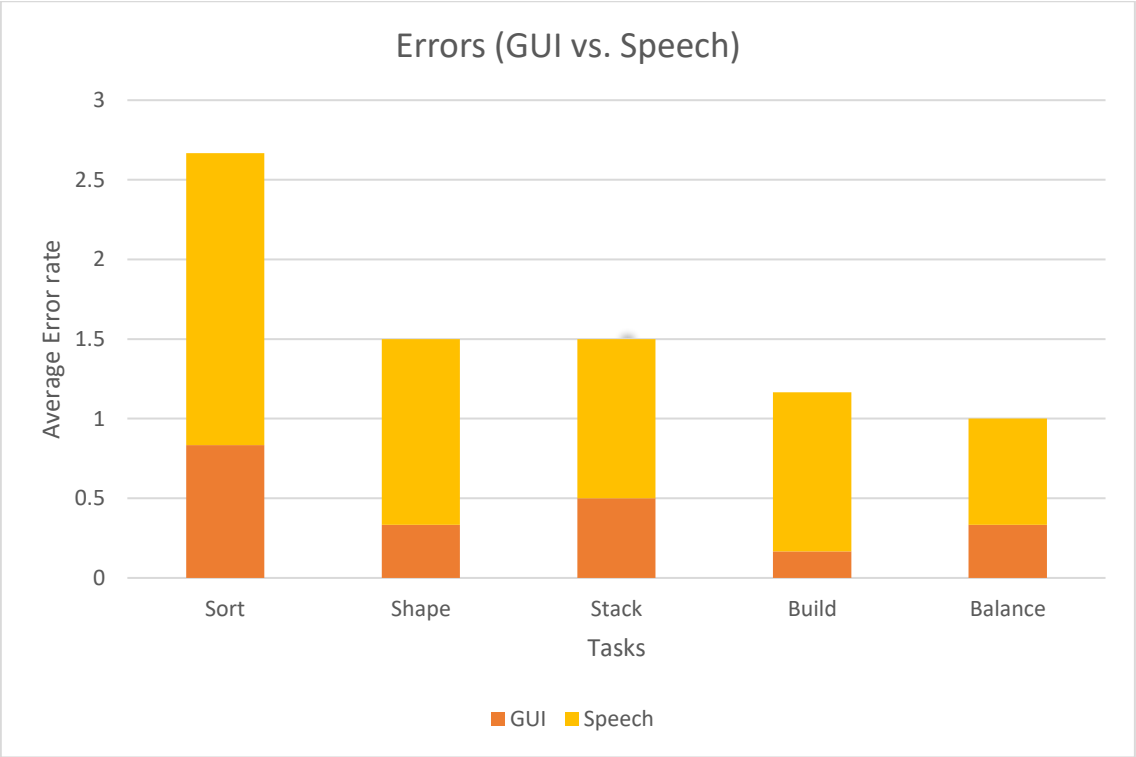

Figure 6-15: Comparison based on errors (speech vs GUI).
The frequency of errors was higher in speech interface as sometimes it didn't recognize the commands and participants have to repeat it.

6.3.2 General comments (speech vs GUI)

After the study, the participants were asked for their comments about the study. One participant mentioned that for the feedback he would prefer an audio signal while for the GUI the popup notification would be better. Some participants preferred speech interface as they did not had to interact with any hardware to operate the robot. Most of the participants supported multi-language feature in speech interface which was not present in GUI.

7 Discussion

We proposed a system in this thesis to design a speech interface which enables the user to communicate with the robot, more specifically PR2, through natural language. The proposed system in this thesis is the extension of the previous study [SKM18] and implemented as suggested in the future work section [Kra17]. The system consists of following sections which work together to achieve the goal which was discussed at the beginning of the thesis in chapter 1.

The first section of the system is the speech recognizer which receives the voice commands of the user and converts them into corresponding text. There are a few challenges to deal with in speech to text conversion in which the noise is the biggest challenge. As the robot can be used in any environment such as homes, industries, shopping centers, clubs etc. and these environments can be very noisy for the robots to detect the speech signal of humans. To deal with the noise short commands structure can be used [Nor05] along with the ambient noise adjustments. In our system, we first adjust the ambient noise by listening to the environment for 1 second before taking the input commands from the user. It helps the system to differentiate between the speech signal and background noise. Another factor in improving the performance of the speech recognizer is the usage of good quality microphones which also deals with the noise reduction from the speech signals. Furthermore, the speech recognizer uses Google speech API for the conversion, so the good internet speed results in less waiting time for speech to text conversion which ultimately increase the performance of the whole system. The proposed system in this thesis is able to communicate with the user in multiple languages, however, only the English and German language are implemented.

The converted text is then transferred to the task manager which is responsible to perform the operations. The data transferred through the server-client method by establishing a TCP connection as it sends the acknowledgment after reception of every packet, where the speech manager acts as a server and task manager acts as a client. We made the communication bi-directional because the client also sends back the feedback to the server which delivers to the user. The task manager assigns the operation to the PR2 by keeping track of the current state of the task and next available actions. After every operation, the task manager sends back the feedback to the speech manager that is whether the operation was successful or failed as well as at the completion of the task. The received feedback, in the form of text, is then converted into an audio signal with the help of text to speech converter.

The proposed system in this thesis evaluated with five tasks through the user study. These tasks consist of sort, shape, stack, build and balance which are designed and ordered based on the difficulty level. These tasks were performed in the laboratory on the table top and the robot does not have to move around in order to complete these tasks. The system consists of two modes of operation that is human commands in which user have to give all the commands to the robot, one after the other, in order to complete the task and autonomous mode where the user just have to tell the name of the task which needs to be performed. The study was conducted with human commands mode as it requires more interaction with the robot (PR2). The system was also tested with other modalities like graphical user interface and text input to make the comparison between them.

The results included in above chapter clearly shows that almost every participant found speech interface more natural, easy and comfortable to use however the delay in speech conversion and repetition of commands due to noise interference, reduced its performance. Despite the fact that participants preferred interacting with the robot in natural language but none of them like to repeat the same command again and again. The participants made many errors in the first two tasks than later tasks even they were easier (see figure 6-8), the reason of this behavior might be that they were not aware how to use the interface. The figure shows a nice declined line of errors from first task till the end, so it can be expected that if all the tasks were performed again with same participants then the number of errors would be much lesser.

7.1 Future work

As discussed earlier in above section that system is able to communicate in multiple languages with the user but currently only two languages are integrated so further languages can also be integrated in order to make it more user-friendly and language independent. Moreover, we are using a small dictionary for the mapping of the commands which can be extended to enable the robot to understand and perform more operations. User study helped a lot to identify the areas which needs further improvements like the delay in speech interface can be reduced up to the extent where it becomes imperceptible for the user because it is the most common comment from the participants regarding the improvement. Secondly, speech recognizer can be further improved as sometimes it missed the speech signal and user have to repeat it again. The tasks can also be further improved in the future in a way that robot has to move around to complete them in order to make the human-robot interaction more natural.

Furthermore, possible research topics for future work consists of the integration of artificial intelligence so that robot can also suggest human about next suitable actions. Another possibility of future research in the field of speech interface is that the robot might be able to identify and differentiate the users with the help of their voice and customize itself according to the personal preference of the user. As the system used a server-client model, with one server and one client, it can further extend in the future to support multi server-client relationship.

8 Conclusion

In this thesis, we proposed a system to develop a speech interface for human-robot interaction. The system allows the user to select the language of communication. It was observed that speech interface works well under less noisy environment along with good quality of the microphone and good internet speed as it is using Google speech API for speech to text conversion. The proposed system consists of two main parts speech manager and task manager. The speech manager is responsible to communicate with the user and task manager handles tasks execution based on the commands received from the user. They both communicate with each other over TCP connection so good internet connection is also required for this communication to reduce the chances of packet loss or delay.

It can be concluded, based on the results gathered from the user study, that the human-robot interaction feels more natural if the communication is done in natural language. However, some participants mentioned that the audio signal generated from the computer does not sounds natural, which can be improved to resemble the human voice. Furthermore, a dialogue management system was able to eliminate the ambiguities and provide useful information to the user along the queries so they can better understand the situation. It can be seen from the figure 6-8, that the number of errors were higher in the beginning tasks which decreased linearly, so it can be expected that most of the errors were caused due to the fact that most of the participants were not familiar with the speech interface. As the PR2 robot was able to understand voice commands and perform the tasks, designed for the study, so more complex tasks can be designed to test the efficiency of the system. Another important factor in human-robot collaboration is safety, so the example of the task might be the scenario in which the robot have to move around to complete the operation, in this scenario the level of safety can be measured that how close the humans can go to the robot.

Bibliography

[AZS17] ALU, D., ZOLTAN, E., & STOICA, I. C. (2017). Voice Based Emotion Recognition with Convolutional Neural Networks for Companion Robots. SCIENCE AND TECHNOLOGY, 20(3), 222-240.

[ADS+07] Attardi, G., Dell'Orletta, F., Simi, M., Chanev, A., & Ciaramita, M. (2007). Multilingual dependency parsing and domain adaptation using DeSR. In Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL).

[AKH+99] Araki, M., Komatani, K., Hirata, T., & Doshita, S. (1999). A dialogue library for task-oriented spoken dialogue systems. In Workshop on Knowledge and Reasoning in Practical Dialogue Systems.

[AKV+09] Atrash, A., Kaplow, R., Villemure, J., West, R., Yamani, H., & Pineau, J. (2009). Development and validation of a robust speech interface for improved human-robot interaction. International Journal of Social Robotics, 1(4), 345.

[AR82] Atal, B., and J. Remde. "A new model of LPC excitation for producing natural-sounding speech at low bit rates." Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'82.. Vol. 7. IEEE, 1982.

[BAK+01] Biem, Alain, Shigeru Katagiri, Erik McDermott, and Biing-Hwang Juang. "An application of discriminative feature extraction to filter-bank-based speech recognition." IEEE Transactions on Speech and Audio Processing 9, no. 2 (2001): 96-110.

[BKT+05] Breazeal, C., Kidd, C. D., Thomaz, A. L., Hoffman, G., & Berlin, M. (2005, August). Effects of nonverbal communication on efficiency and robustness in human-robot teamwork. In Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on (pp. 708-713). IEEE.

[Blo17] M. I. Bloch. "Natural language understanding and communication for human-robot collaboration." MSc Thesis, University of Stuttgart, (2017)

[BSH93] Brennan, Susan E., and Eric A. Hulteen. "Interaction and feedback in a spoken language system." AAAI-93 Fall Symposium on Human-Computer Collaboration: Reconciling Theory, Synthesizing Practice. 1993.

[CHM04] Clark, Herbert H., and Meredyth A. Krych. "Speaking while monitoring addressees for understanding." Journal of memory and language 50.1 (2004): 62-81.

[CJL16] Cross, James, and Liang Huang. "Span-based constituency parsing with a structure-label system and provably optimal dynamic oracles." arXiv preprint arXiv:1612.06475 (2016).

[CL01] Cassimatis, Nicholas Louis. Polyscheme: a cognitive architecture for integrating multiple representation and inference schemes. Diss. Massachusetts Institute of Technology, 2001.

[Cla94] Clarke, R. 1994. Asimov's Laws of Robotics: implications for information technology. IEEE Computer 26(12) and 27(1)

[CNA02] Campbell, Nick, and Andrew Hunt. "Concatenation of speech segments by use of a speech synthesizer." U.S. Patent No. 6,366,883. 2 Apr. 2002.

[CTB+04] Cassimatis, N. L., Trafton, J. G., Bugajska, M. D., & Schultz, A. C. (2004). Integrating cognition, perception and action through mental simulation in robots. Robotics and Autonomous Systems, 49(1-2), 13-23.

[Dau99] Dautenhahn, K. (1999). Socially intelligent agents and the primate social brain-Towards a science of social minds. Adaptive Behaviour, 7(3-4), 3-4.

[DGE90] Doddington, George R., and Enrico Bocchieri. "Speaker independent speech recognition method and system." U.S. Patent No. 4,908,865. 13 Mar. 1990.

[DM02] Denecke, Matthias. "Rapid prototyping for spoken dialogue systems." Proceedings of the 19th international conference on Computational linguistics-Volume 1. Association for Computational Linguistics, 2002.

[DOQ02] Dautenhahn, K., Ogden, B., & Quick, T. (2002). From embodied to socially embedded agents–implications for interaction-aware robots. Cognitive Systems Research, 3(3), 397-428.

[DRP99] Donovan, Robert E., and Philip C. Woodland. "A hidden Markov-model-based trainable speech synthesizer." Computer speech & language 13.3 (1999): 223-241.

[FCT+01] Fong, T., Cabrol, N., Thorpe, C., & Baur, C. (2001). A personal user interface for collaborative human-robot exploration. In 6th International Symposium on Artificial Intelligence, Robotics, and Automation in Space (iSAIRAS) (No. LSRO2-CONF-2001-001).

[FKH+06] Fong, T., Kunz, C., Hiatt, L. M., & Bugajska, M. (2006, March). The human-robot interaction operating system. In Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction (pp. 41-48). ACM.

[FND03] Fong, T., Nourbakhsh, I., & Dautenhahn, K. (2003). A survey of socially interactive robots. Robotics and autonomous systems, 42(3-4), 143-166.

[FS91] Furui, Sadaoki. "Speaker-dependent-feature extraction, recognition and processing techniques." Speech Communication 10.5-6 (1991): 505-520.

[FTT+03] Fong, Terrence, Charles Thorpe, and Charles Baur. "Collaboration, dialogue, human-robot interaction." Robotics Research. Springer, Berlin, Heidelberg, 2003. 255-266.

[Gor01] Goren-Bar, D. (2001). Designing model-based intelligent dialogue systems. In Information modeling in the new millennium (pp. 268-284). IGI Global.

[GYO01] Guji, Yoshiki, and Koji Ohtsuki. "Multiple language speech synthesizer." U.S. Patent No. 6,243,681. 5 Jun. 2001.

[HAA+04] Hara, I., Asano, F., Asoh, H., Ogata, J., Ichimura, N., Kawai, Y., ... & Yamamoto, K. (2004, October). Robust speech interface based on audio and video information fusion for humanoid HRP-2. In Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on (Vol. 3, pp. 2404-2410). IEEE.

[HJ95] Hansen, John HL. "Analysis and compensation of speech under stress & noise for environmental robustness in speech recognition." Speech under Stress. 1995.

[JC06] Junqua, Jean-claude. "Customizing the speaking style of a speech synthesizer based on semantic analysis." U.S. Patent No. 7,096,183. 22 Aug. 2006.

[JC93] Junqua, Jean-Claude. "The Lombard reflex and its role on human listeners and automatic speech recognizers." The Journal of the Acoustical Society of America 93.1 (1993): 510-524.

[JHS+16] Johansson, M., Hori, T., Skantze, G., Höthker, A., & Gustafson, J. (2016, November). Making turn-taking decisions for an active listening robot for memory training. In International Conference on Social Robotics (pp. 940-949). Springer, Cham.

[JMG+14] Johansson, Martin, Gabriel Skantze, and Joakim Gustafson. "Comparison of human-human and human-robot turn-taking behaviour in multiparty situated interaction." Proceedings of the 2014 Workshop on Understanding and Modeling Multiparty, Multimodal Interactions. ACM, 2014.

[JRN07] Johansson, Richard, and Pierre Nugues. "Extended constituent-to-dependency conversion for English." (2007).

[KD65] Knuth, Donald E. "On the translation of languages from left to right." Information and control 8.6 (1965): 607-639.

[KGH+03] Kerstin, Anders Green, Helge Hüttenrauch and Severinson-Eklundh. "Social and collaborative aspects of interaction with a service robot." Robotics and Autonomous systems42.3-4 (2003): 223-234.

[Kha98] Khan, Z. (1998). Attitudes towards intelligent service robots. NADA KTH, Stockholm, 17.

[KU99] Kölzer, A., & Ulm, D. (1999). Universal dialogue specification for conversational systems.

[Kul06] Kulyukin, V. A. (2006, March). On natural language dialogue with assistive robots. In Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction (pp. 164-171). ACM.

[LBL+87] Levinson, P., Brown, P., Levinson, S. C., & Levinson, S. C. (1987). Politeness: Some universals in language usage (Vol. 4). Cambridge university press.

[LKF+10] Lee, M. K., Kiesler, S., Forlizzi, J., Srinivasa, S., & Rybski, P. (2010, March). Gracefully mitigating breakdowns in robotic services. In Human-Robot Interaction (HRI), 2010 5th ACM/IEEE International Conference on (pp. 203-210). IEEE.

[MMM+06] De Marneffe, Marie-Catherine, Bill MacCartney, and Christopher D. Manning. "Generating typed dependency parses from phrase structure parses." In Proceedings of LREC, vol. 6, no. 2006, pp. 449-454. 2006.

[Nor05] Norberto Pires, J. (2005). Robot-by-voice: Experiments on commanding an industrial robot using the human voice. Industrial Robot: An International Journal, 32(6), 505-511.

[RN96] B. Reeves, C. Nass, The Media Equation, CSLI Publications, Stanford, 1996

[RPT00] Roy, N., Pineau, J., & Thrun, S. (2000, October). Spoken dialogue management using probabilistic reasoning. In Proceedings of the 38th Annual Meeting on Association for Computational Linguistics (pp. 93-100). Association for Computational Linguistics.

[SA98] Simmons, R., & Apfelbaum, D. (1998, October). A task description language for robot control. In Intelligent Robots and Systems, 1998. Proceedings., 1998 IEEE/RSJ International Conference on (Vol. 3, pp. 1931-1937). IEEE.

[SC11] Staudte, M., & Crocker, M. W. (2011). Investigating joint attention mechanisms through spoken human–robot interaction. Cognition, 120(2), 268-291.

[SKL+04] Sidner, C. L., Kidd, C. D., Lee, C., & Lesh, N. (2004, January). Where to look: a study of human-robot engagement. In Proceedings of the 9th international conference on Intelligent user interfaces (pp. 78-84). ACM.

[SKM18] Schulz, Ruth, Philipp Kratzer, and Marc Toussaint. "Preferred interaction styles for human-robot collaboration vary over tasks with different action types." Frontiers in neurorobotics 12 (2018): 36.

[SKT07] Sagae, Kenji, and Jun'ichi Tsujii. "Dependency parsing and domain adaptation with LR models and parser ensembles." Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL). 2007.

[SR94] Simmons, Reid G. "Structured control for autonomous robots." IEEE transactions on robotics and automation 10.1 (1994): 34-43.

[TAC11] Thomaz, Andrea L., and Crystal Chao. "Turn-taking based on information flow for fluent human-robot interaction." AI Magazine 32.4 (2011): 53-63.

[TFK13] Torrey, C., Fussell, S., & Kiesler, S. (2013, March). How a robot should give advice. In Proceedings of the 8th ACM/IEEE international conference on Human-robot interaction (pp. 275-282). IEEE Press.

[TMC07] Tomasello, Michael, and Malinda Carpenter. "Shared intentionality." Developmental science 10.1 (2007): 121-125.

[TMS03] Tews, A. D., Mataric, M. J., & Sukhatme, G. S. (2003, September). A scalable approach to human-robot interaction. In Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on (Vol. 2, pp. 1665-1670). IEEE.

[TNK+98] Takahashi, T., Nakanishi, S., Kuno, Y., & Shirai, Y. (1998, October). Human-robot interface by verbal and nonverbal behaviors. In Intelligent Robots and Systems, 1998. Proceedings., 1998 IEEE/RSJ International Conference on(Vol. 2, pp. 924-929). IEEE.

[Kra17] Philipp Kratzer. “Robot assistance for collaborative task execution” MSc Thesis, University of Stuttgart (2017)

[Kar17] Umut Kara. “Natural text to abstract concept mapping for collaborative HRI.” BSc Thesis, University of Stuttgart (2017)

[YKI+08] Yamaoka, F., Kanda, T., Ishiguro, H., & Hagita, N. (2008, March). How close?: model of proximity control for information-presenting robots. In Proceedings of the 3rd ACM/IEEE international conference on Human robot interaction (pp. 137-144). ACM.

[ZT94] Zhao, Yunxin. "An acoustic-phonetic-based speaker adaptation technique for improving speaker-independent continuous speech recognition." IEEE Transactions on Speech and Audio Processing 2.3 (1994): 380-394.

Acknowledgment

I am sincerely thankful to my supervisor Ph.D. Ruth Schulz from the machine learning and robotics department from Institute of Parallel and Distributed Systems of the University of Stuttgart for her wonderful support and guidance during all the phases of my master thesis. I would also like to thank Prof. Dr. rer. nat. Marc Toussaint for giving me this wonderful opportunity to do my master thesis at this institute.

I am also thankful to my family and friends for their moral support throughout my master thesis.

Moin Uddin Kashif

Declaration

I hereby declare that the work presented in this thesis is entirely my own and that I did not use any other sources and references than the listed ones. I have marked all direct or indirect statements from other sources contained therein as quotations. Neither this work nor significant parts of it were part of another examination procedure. I have not published this work in whole or in part before. The electronic copy is consistent with all submitted copies.

Place, date, signature