Voice Interfaces in Mobile Human-Robot Collaboration for Advanced Manufacturing Systems

Submitted By:

Shane Pather

(BScEng, UKZN)

214508491

November 2019

Prof. Glen Bright and Mr. Christian Basson, Mr. Avern Athol-Webb

Submitted in fulfilment of the academic requirements for the degree of Master of Science in Engineering at the School of Mechanical Engineering, University of KwaZulu-Natal

EXAMINER'S COPY

DECLARATION

Declaration by Author

I, Shane Pather, declare that

(i) The research reported in this dissertation, except otherwise indicated, is my original work.
(ii) This dissertation has not been submitted for any other degree or examination at any other university.
(iii) This dissertation does not contain other persons' data, pictures, graphs, or other information unless specifically acknowledged as being sourced from other persons.
(iv) This dissertation does not contain other persons‟ writing unless specifically acknowledged as being sourced from other researchers. Where other written sources have been quoted then:

    a) their words have been rewritten but the general information attributed to them has been referenced;
    b) where their exact words have been used, their writing has been placed inside quotation marks, and referenced.

(v) Where I have reproduced a publication of which I am an author, co-author, or editor, I have indicated in detail which part of the publication was written by myself alone and have fully referenced such publications.

This dissertation does not contain text, graphics, or tables copied and pasted from the internet unless specifically acknowledged, and the source being detailed in the dissertation and in the References sections.

Signed: ______________________________ Date: ______________________________

Mr. S. Pather

Declaration by Supervisor

As the candidate's Supervisor I agree to the submission of this dissertation.

Signed: ______________________________ Date: ______________________________

Professor G. Bright

Declaration by Co-Supervisor

As the candidate's Co-Supervisor I agree to the submission of this dissertation.

Signed: ______________________________ Date: ______________________________

Mr. C.I Basson

ACKNOWLEDGEMENTS

First and foremost, I would like to thank the Lord for the knowledge and strength he has given me to complete the research presented in this dissertation. I would also like to acknowledge the love and support of my family during the undertaking of this degree. A special thanks also goes out to my supervisors Prof. Glen Bright and Mr. Christian Ivan Basson for providing me with the platform to further my research in the field of Engineering. I would also like to thank Dr. Alexandra Birch, a Senior Research Fellow at the University of Edinburgh for her guidance with problems faced by modern voice interface technology.

DEDICATION

Dedicated to the Lord, my family and the memory of my father

"Necessity is the mother of invention"

~ Plato ~

PREFACE

This paper investigated a conceptualized methodological approach for the integration of voice interface technology into a manufacturing environment. Voice interface technology was investigated to develop an operational benchmark – when subjected to simulated manufacturing conditions. The study subsequently established an optimization for the accuracy and reliability of a voice interface system. A proposed method of application for manufacturing environments using a mobile robot design was also investigated. The design of the voice interface optimization system focused on phrase accuracy and reliability; when subjected to common manufacturing noise conditions. The voice interface optimization system was subjected to a simulated manufacturing environment to create synthetic data, the data was empirically analyzed and tested to formulate appropriate conclusions. The mobile robot design paired with voice interface technology was developed to create a platform for the implementation of voice interfaces into manufacturing environments. The mobile robot was subjected to path accuracy testing to establish the validity of its embedded navigation system. Testing of the mobile robot design was carried out in a simulated manufacturing environment.

DECLARATION OF PUBLICATIONS

Presented below are the peer-reviewed publications that include the research presented in the dissertation. The undersigned agree that the following submissions were made and accepted as described and that the content therein is contained in this research.

Publication 1: IEEE Africon – 2019 (Accepted)

Pather, S. Basson, C and Bright, G. *"An Elementary Review of Cloud Computing with Implications for Manufacturing Systems"*, IEEE Region 8 flagship conference in Africa. Published August 2019, 4 Pages.
Shane Pather was the lead author of this paper and conducted all research the supervision of Professor Glen Bright and Mr. Christian Basson.

ABSTRACT

Proliferating instances of voice interface technology outside the manufacturing domain has posed the question; why does the manufacturing domain lack voice interface technology? The fourth industrial revolution (Industry 4.0) has changed the face of modern manufacturing. Progressive development in the manufacturing domain proliferates with instances of Cyber-Physical Systems (CPS). These systems have proven to refine the manufacturing domain through effectively analyzing, monitoring, and creating an interoperable, automated, cyber-machine-centred environment. A fundamental aspect of CPS is data acquisition. Naturally, sensor-based tools and industrial communication technology must be considered to create an environment aligned to the ideals of Industry 4.0. The research that follows was motivated by the need for versatile and flexible mediums of communication within CPS. The research efforts aligned itself to establishing an effective means of incorporating voice interface technology into the manufacturing domain.

Due consideration was taken into the high levels of flexibility, reconfigurability, and adaptability associated with customizable manufacturing. The demand of intricacy and precision associated with robotic operations requires highly flexible communication interfaces. The research that follows firstly attempted to understand the reasons for absence, and implication of voice interface technology in manufacturing environments. The purpose of this paper was to investigate the performance of leading voice interface technology when subjected to manufacturing environment conditions. Additionally, an attempt was made to develop a structured conceptualized methodological approach for the implementation of voice interface technology into manufacturing environments. This area of research was explored to develop a platform for flexible robotic operations; as compared to the conventional tool-centered approach.

The field of voice interface technology introduces itself as a relatively new communication medium, and though these systems rapidly advance, methods of development present themselves as proprietary work of the technological institutes facilitating advancement. As such, the methodology structure that resulted was developed from a combination of work presented from academic literature, information gained from enrolment into courses facilitated by industrial experts, attending a workshop facilitated by a senior research expert in the field of voice technology, interviews with academic personnel, interviews with industrial experts and consultation with industrial engineers. Research and consultation presented critical areas of concern within voice interface technology for manufacturing environments being; lack of implementation methods in manufacturing environments, high levels of language fluency requirements, low interpretability constraints, being highly sensitive to the environment of use, and a low adaptability to different languages. Due to the lack of performance data, leading voice interface technology, “OK Google”, was introduced into a simulated manufacturing environment. Performance data was extracted using Googles Web-based voice data-logging system. A python programme was designed for the analysis of the accuracy and reliability of the extracted data – with key focus being placed on the phrase recognition rate. Thereafter, the data was subjected to a probability distribution analysis, namely, the probability density function. This was done to identify the validity of the results and a means of identifying inconsistencies. Results showed that “OK Google” was capable of operating at a minimum phrase recognition rate of 69% when attempting to identify a phrase in a quiet room environment. The maximum phrase recognition rate for a quiet room environment being 93.5%. Additionally, “OK Google” proved incapable of executing a task when subjected to 70 and 80 decibel (dB) ranges. As such, the results obtained displayed a need for an optimization design of “OK Google”, so as to allow for implementation validity in manufacturing environments. The optimization design encompassed the use of web-based servers, and a cloud-based internet platform (IoT platform). This was done in an attempt to conform to the fourth industrial revolution, and to increase the reputability of voice technology in environments previously deemed unfit for implementation. The voice interface optimization design was subjected again, to a probability distribution analysis, and was introduced into a mobile navigation robot design to facilitate the ability to carry out task-based operations based on voice commands. The optimization system proved to increase the ability of “OK Google” when subjected to 70 and 80 dB ranges. The minimum and maximum recognition rates obtained for the above range was found to be 12% and 91.17 %, respectively. All tests were carried out in the Mechatronics and Robotics workshop at the University of KwaZulu-Natal so as to simulate a manufacturing workspace.

Results of the voice interface optimization design proved to display highly favourable results when compared to the operational benchmark considered outside the optimization. When subjected to interference levels previously displaying low-levels of phrase accuracy and reliability, the optimized system incorporating cloud-based IoT platforms and Web-based servers displayed optimization results with increases of up to 100 %. Additionally, the integration of the mobile navigation robot with the voice interface technology proved highly accurate, though precision was limited to a minimum of 111.5 cm and a maximum of 285.5 cm from the desired geographical location. The results acquired proved the validity of voice interfaces for manufacturing environments. The research promoted the scientific contribution of the investigation, generation, analysis, and design of an intelligent voice-controlled mobile communication system, with the ability to be implemented into manufacturing and plant-based environments.

Keywords — Voice-Interfaces, Mobile Robotics, Flexible Manufacturing

ACRONYMS

AMS	Advanced Manufacturing Systems
CMS	Cellular Manufacturing Systems
CNS	Computer Numeric control
CNS	Convolutional Neural Networks
Db	Decibels
DMS	Dedicated Manufacturing Systems
DPM	Deformable Parts Model
EssER	Essential Element Region
FMS	Flexible Manufacturing Systems
FR	Facial Recognition
IaaS	Infrastructure-as-a-Service
IFTTT	If Then, Then That
NGMS	New Generation Manufacturing Systems
NI	National Instruments
NIST	National Institute of Standards and Technology
OFIQ	Objective Face Image Quality
PaaS	Platform-as-a-Service
PRR	Phrase Recognition Rate
RFIQ	Relative Face Image Quality
RIQ	Face Image Quality
RMS	Reconfigurable Manufacturing Systems
SaaS	Software-as-a-Service

STT	Speech to Text
TTS	Text to Speech
VRTC	Vibration Research and Testing Centre
W3C	World Wide Web Consortium
WCAG	Web Content Accessibility Guidelines
WER	Word Error Rate
Wi-Fi	Wireless Fidelity

NOMENCLATURE

$\int$	Region of desired density function
D	Number of words Deleted
$f(x)$	Probability density function
$f(X)$	Probability density function of continuous random variable
I	Number of words Inserted
N	Number of words detected
PRR	Phrase recognition rate
R	System Reliability
S	Number of words Substituted
t	Time (seconds)
WER	Word Error rate

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1 Background

The rationale for this research originated from the need to align manufacturing systems to the proliferating technological age. The ability of a manufacturing environment to grow and thrive in the industrial domain is highly dependent on innovation. Innovation can directly relate to product innovation or manufacturing innovation; conventionally. This dissertation however, contributed to innovation in manufacturing environments, by creating a platform to refine the manufacturing workforce. This dissertation aimed at creating a system that improves the way people work, with the goal of improving manufacturing through innovative systems for people. Flexibility is key for an innovative environment - not just for manufacturing machinery, but for manufacturing environments as a whole. This characteristic is essential due to its ability to reduce production times and create a more conductive environment for product customizability. Additionally, if manufacturing environments are considered, it provides a highly evolved platform for process optimization [1]. Research conducted in this dissertation accounted for the areas of concern presented by modern manufacturing and aimed at scientific contributions accentuated towards these concerns.

1.2 Manufacturing challenges

The age of product customizability dawns over modern manufacturing. Much is known about what needs to be done, however, very little is known of how it can be done. When delving into the crux of customizability, essentially, one factor predominantly emerged - flexibility. In modern times, customers opt for products that need not specifically apply to aesthetics, but changing physical characteristics. As such, production equipment needs to not just grow, or be optimized, but these machines need to evolve. Evolution however, seems to have plateaued. It was found that a variety of contributing factors accompanied this state of limbo – but not for the lack of trying, these factors included; [2] [3] [4] [5]

- Development of flexible systems being confined to the academic domain with less focus placed on industrial applications,
- An absence of a methodological structure for implementing flexible systems into the industrial domain,
- An absence of due consideration being given to human integration of flexible systems,
- Excessive programming times required for system reconfigurability,
- High costs associated with the development of a highly flexible manufacturing environment, and

- High costs associated, with adapting pre-existing systems to provide flexibility.

1.3 Motivation for the study

The initial stages of the dissertation saw the authors attempting to gauge the state of manufacturing environments as a whole. The approach taken was not solely focused on optimization of pre-existing systems, but rather, in establishing why the areas of concern presented in Section 1.2, persisted. The field of engineering spans across the globe in a variety of fields and at present, engineering progresses along the lines of digitalization; with systems of collaborative hybrid human and robotic operations, additive and subtractive processes, metal and composite materials, and cyber and physical systems [6]. When due consideration was taken into the high levels of reconfigurability, adaptability, and flexibility required of machinery for customizable manufacturing, the authors of the dissertation opted to focus on machine flexibility associated with the fourth industrial revolution. As such, focus was placed on cyber-physical systems. The reason for this choice was due to modern industries moving operational functions towards a digital platform. It was established that the demand of intricacy and precision associated with robotic operations required highly flexible communication interfaces. Communication interfaces were researched, and it was found that the advent of voice-assistants to the modern age had the potential to completely shift conventional tool-centered manufacturing processes to a new, and more versatile, flexible approach. Voice assistants could change the way manufacturers view communication. These assistants focused on automation, but the style of automation revolved around an essential, human-centered approach. Years of preparation and advances in the field of machine learning and artificial intelligence had contributed to the highly efficient systems we know them to be. Improved interactions of human-to-human and human-to-computer systems were but a few instances of commonly ventured system integrations of voice assistants [7]. There were however, a reasonable area of concern within the field of voice assistants, but more specifically to the field of academia and, as this dissertation dictates, manufacturing. Though voice assistants proliferated in instances of cellular devices, they seemed to be confined within the cellular domain. Additionally, though much advancements have been made in the field of voice assistants, these advances remain proprietary to the facilities creating this growth, namely corporations such as Google and Amazon. Due to the nature of these facilities and the environment in which they operate, much less effort is placed on integration of voice-assistants into manufacturing platforms. As such, the research that follows firstly attempted to understand the reasons for absence, and implication of voice interface technology in manufacturing environments. The purpose of the paper being to investigate the performance of leading voice interface technology when subjected to manufacturing environments. Additionally, an attempt was made to develop a structured conceptualised methodological approach for the implementation of voice interface technology into manufacturing environments. Research found motivated the development of a platform for flexible robotic operations; as compared to the conventional tool-centered approach.

1.4 Purpose of research

The research was motivated by the need for a versatile and flexible medium of communication within CPS. The research efforts aligned itself to establishing an effective means of incorporating voice interface technology into the manufacturing domain. The research that follows attempted to understand the reasons for absence, and implication of voice interface technology in manufacturing environments. The purpose of this paper was to investigate the performance of leading voice interface technology when subjected to manufacturing environment conditions. This paper established performance data for voice assistant "OK Google". Upon analyzing the performance data, it was established that the voice assistant needed to be optimized to allow for integration into a manufacturing domain. As such, the assistant was optimized using an IoT based platform and web-based server. The dissertation also developed a means of implementation of the voice assistant into a manufacturing environment through the integration with a mobile navigation system design. The mobile navigation system design facilitated the use of a voice assistant in a manufacturing domain.

1.5 Project objectives

The objectives of the research were to:

1. Research and establish a list of the essential areas of concern within manufacturing
2. Research and develop a conceptualized methodological approach for integrating voice assistants into manufacturing environments
3. Generate synthetic voice-assistant data using a simulated manufacturing environment
4. Research and optimize voice-assistant technology, conforming to the fourth industrial revolution
5. Analyze the performance of the optimized system and evaluate applicability of the optimized system.
6. Research, design and implement a suitable mechatronic system integrating voice interfaces with mobile robots. The system must include the development of the electronic hardware, the programming of machine control algorithms, and the creation of a suitable interface for human interaction.

1.6 Scientific contribution of the dissertation

This dissertation aimed to contribute to the scientific community in the following areas;

- The generation of operational benchmark data of a voice interface when subjected to manufacturing conditions
- The design of a flexible mechatronic system capable of carrying out operations based on voice commands
- The generation of a structured methodological approach for the generation, analysis, optimization and implementation of voice-interfaces into manufacturing environments.
- The analysis of the baseline accuracy, phrase error rates (PER) and reliability of voice-assistants when subjected to industrial conditions.
- The optimization of a voice-interface system for manufacturing environments with the optimization approach conforming to the ideals of the fourth industrial revolution; with the analysis of the accuracy, phrase error rates and reliability thereof,
- Developing a conceptualized practical approach for implementation of voice-assistants into the manufacturing domain.

1.7 Research question

The research question addressed in the dissertation related to flexible human-robot communication, the research question addressed is stated below:

How can a mechatronic system be designed for a manufacturing environment that integrates voice interface technology and mobile robotics?

1.8 Research publications

Presented below is the peer-reviewed publication that included the research presented in the dissertation. The undersigned agree that the following submissions were made and accepted as described, and that the content therein is contained in this research.

Publication 1: IEEE Africon – 2019 (Accepted and Published)

Pather, S. Basson, C and Bright, G. “An Elementary Review of Cloud Computing with Implications for Manufacturing Systems”, IEEE Region 8 flagship conference in Africa. Published August 2019, 4 Pages.

Shane Pather was the lead author of this paper and conducted all research the supervision of Professor Glen Bright and Mr. Christian Basson.

1.9 Outline of dissertation

Chapter 1 – Introduction: The introduction chapter provides the reader with key insight in the background of the research dissertation, manufacturing challenges encountered, the motivation for study and lastly, the dissertation objectives.

Chapter 2 – Literature Review: A new approach to literature evaluation was presented into this chapter. Research was conducted to isolate key issues presented with the manufacturing domain and research thereafter focused on highlighting these issues and addressing them.

Chapter 3 – Methodology: The Methodology chapter provides the structured and validation methods initiated throughout the dissertation. It presented the approaches considered and the reason as to why the methods were considered.

Chapter 4 – Electronic Design: This chapter encompasses the design component of the dissertation. Functionality requirements, selection methods, configurations and system assembly can be viewed in this chapter.

Chapter 5 – Data Analysis: This chapter embodies the testing results and evaluation of the data extracted from the designed systems.

Chapter 6 – Discussion: This chapter summarises the findings of the dissertation, highlights on the systems performance and contributions, as well as provides insight into the applications of the designed system.

Chapter 7 – Conclusion: The dissertation concludes with the objectives breakdown and how they were met, and provides insight into the future work to be addressed and considered.

1.10 Chapter summary

Chapter one, the introductory chapter, outlines the rationale for the research conducted so as to justify content validity. Research objectives were outlined in this chapter. Furthermore, motivation for the research was expounded and the issues concerning the implementation of voice technology in a manufacturing environment was discussed. Chapter one additionally provided a short summary of each chapter of the dissertation.

2. LITERATURE REVIEW

2.1 Introduction

The major advent for manufacturing in the 21st century can be confined to something very simple, yet quite complicated. A factor of mild subtlety; though it possesses a high level of complexity. This somewhat unbridled factor that seems to define the future of manufacturing is that of Product Customizability. Customers opt for products that not only suit their needs but also their desires. This, however, introduces a variety of complications for production equipment in manufacturing environments. Delving into the depts of manufacturing literature, the subsequent chapter attempts to create an understanding of the complications faced by Advanced Manufacturing Systems. Thereafter, an attempt is made to provide an approach to solve some of the limitations presented. The Literature review that follows covers the following research areas;

- Reconfigurable Manufacturing Systems,
- Programming of Robotic Systems,
- Communication Networks, and
- Collaborative Robotic Systems

2.2 Reconfigurable Manufacturing Systems

The research that follows in this section was conducted to gain an understanding of the manufacturing systems that currently exist, their working structure and the problems they face.

Manufacturing systems at their core, posses a manifold of inherent limiting factors when contested by dynamic market challenges. In response to these challenges, New Generation Manufacturing Systems (NGMSs) attempt to accentuate on areas such as flexibility, reconfigurability, and artificial intelligence. Additionally, NGMSs need to be highly interoperable. These systems are likely to be comprised of cyber and physical systems, metal and composite materials, and hybrid systems of human and robot collaboration [6]. The conceptualization of the paradigm of Reconfigurable Manufacturing was devised through the collaboration between researchers in Europe and the United States. The definition that followed stated [8] :

" A Reconfigurable Manufacturing System is a system that combines the advantages of Dedicated Manufacturing Systems and Flexible Manufacturing Systems by designing it at the outset for rapid change in structure, as well as in its machines and controls, in order to quickly adjust production capacity and functionality in response to market or product changes."

The aforementioned definition provides a pellucid view of Reconfigurable Manufacturing Systems (RMSs). As the definition states, RMSs embody the advantages of Dedicated Manufacturing Systems (DMSs) and Flexible Manufacturing Systems (FMSs). The efficacy of RMSs can be attributed to its intermediate relationship of product variety and product capacity (Figure 2-1). To fully grasp the functionality of RMSs; DMSs and FMSs were researched. Research was carried out to understand the collaborative union of DMSs and FMSs in RMSs

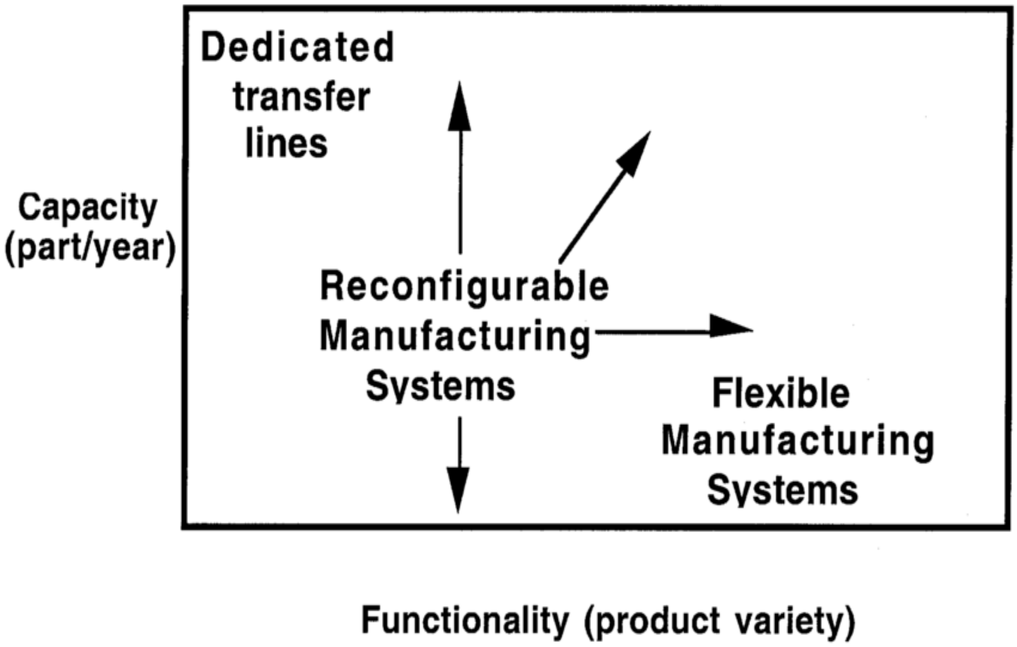


Figure 2-1: Mapping several types of manufacturing systems in capacity-functionality coordinates [9]

2.2.1 Dedicated Manufacturing Systems

Historically, DMSs was the first manufacturing system; introduced at the beginning of the 20th century. DMSs are designed specifically for the production of a single, or a bounded group of products. The dedicated nature of DMSs provides for a significantly lower capital startup as compared to FMSs. DMSs can be affiliated with transfer line technology, fixed automation, robust system performance and dedicated machinery as a whole. Characteristically, DMSs is capable of handling high production rates of core products; though inefficiency does present itself with low levels of flexibility. As such, featured products for DMSs are required to be constant throughout the systems lifecycle. Additionally, product customizability for DMSs invariably requires a significant cost incurred by companies. These costs fall along the lines of machine change-overs, implementation, and adaptability [9].

2.2.2 Flexible Manufacturing Systems

FMSs are systems that consist of computer numeric control (CNC) machine tools. These tools are interconnected by automatic material handling and storage systems and controlled by an integrated computer system. FMSs are very advantageous when viewed from a flexibility perspective, though they do possess their share of limitations. Limitations accompanied by FMSs present themselves in the form of low throughputs – as compared to DMSs, and dedicated equipment fostering increased component costs. Cellular Manufacturing Systems (CMSs) are capable of overcoming some of the limitations of DMSs and FMSs. These systems use multiple independent working cells dedicated to product families with similar processing requirements. However, CMSs are specifically designed for the production of a specific set of products with stable demand levels and long lifecycles [10]. A tabular assessment of the aforementioned manufacturing system can be viewed in Table 2-1. Key attributes are paired against each other and evaluated on an intensity, adaptability and variance scale [11]. The appropriate scales are provided in

Table 2-2, Table 2-3 and Table 2-4.

Table 2-1: Comparison of existing Manufacturing Systems [10]

	DMS	**FMS**	**CMS**	**RMS**
Cost per part	Low	Reasonable	Medium	Medium
Demand	Stable	Variable	Stable	Variable
Flexibility	No	Medium	Medium	Customizable
Machine structure	Fixed	Fixed	Fixed	Changeable
Product family formation	No	No	Yes	Yes
Productivity	Very high	Low	High	High
System structure	Fixed	Changeable	Fixed	Changeable

Variety	No	High	High	High

Table 2-2: Intensity Scale

0	1	2	3	4
Low	Reasonable	Medium	High	Very High

Table 2-3: Adaptability Scale

0	1	2
Fixed	Changeable	Customizable

Table 2-4: Variance Scale

0	1
Variable	Stable

Predominantly, the characteristic traits of RMSs appeal to changing markets. These characteristics have been further analysed below. In due relation to the framework and the predecessors of RMSs, research progresses along the lines of evolving system configurations and layouts [12].

2.2.3 RMSs Features and Market Contribution

Research conducted on RMSs found six defining features forming the core structure of RMS [13]. These features include; Convertibility, Scalability, Modularity, Intergrateability, Customization, and Diagnosability. A brief exposition of these characteristics are provided below.

- *Convertibility*: This is the ability of a system to transform its functionality to suit new requirements of production.

- *Scalability*: This refers to the ability of a system to modify its production capacity. This is achieved by adding or subtracting manufacturing resources or changing the components of a system.
- *Modularity*: This characteristic refers to operational functions being compartmentalized into units. With this being achieved, units can be manipulated between alternate production schemes to allow for the best possible arrangement.
- *Integrability*: Integrability focuses on the rapid and precise integration of modules. This is achieved by facilitating communication and integration of information, mechanical and control interfaces.
- *Customization*: Customization focuses on the customer. It refers to the ability of a system to produce customized products based on customer specification.
- *Diagnosability*: Diagnosability accentuates on monitoring techniques that evaluate the state of a system. This allows for the detection and diagnoses of product defects; providing a platform for effective condition monitoring.

2.2.4 RMS – Areas of Concern

Given the embedded encapsulating characteristics of RMS, it is easily noticeable why cooperative heads and factories of the future would hold RMSs at quite a high predilection. In retrospect, customizable manufacturing should be of utmost importance for any manufacturer seeking progressive growth. Research on RMSs however, did present a variety of system complications [2]. One complication presented itself as being quite ubiquitous. That complication being implementation. Many publications expound the introduction of RMSs into conventional manufacturing systems; though less accentuation is directed towards an actual methodological approach [3]. Additionally, a large contributor against implementation lies in a manufacturers sense of comfortability. Manufacturers are resistant to change, though their resistance is quite understandable. Innovation halts when successful industries thrive on outdated technology – in a technologically advanced age. Though, with the absence of clearly defined methodologies, steep learning curbs of intricate systems, and the absence of backward-compatible knowledge adaptability [4]; resistance can be proved warranted. The proliferating instances of innovation in industry, however, is key. Learning factories have played a significant role in this through a knowledge share between academics and industry. Small scale reconfigurable manufacturing systems, as well as flexible and assembly systems, have been simulated for the purpose of skills development and eventual ease of transition into a more customizable manufacturing environment [5].

2.3 Understanding the programming process of robotic systems

The research that follows in this section was conducted as directed by an industrial programming expert (Appendix A). Suggestions were made to understand general programming procedures for robotic devices and their impact on manufacturing.

Time is an entity that can be considered to govern all and is undoubtedly a critical factor to consider in manufacturing. Programming of robotic systems in Advanced Manufacturing Systems can be viewed as quite a time-consuming process. Working along the lines of customizability, proliferating instances of customizable products would require programming times for reconfigurable systems to be significantly reduced [14]. This would aid in systems being able to adapt more easily to changing market demands. Additionally, large scale reconfigurability of machines is limited by a lack of expertise. In hindsight, these limitations pave the way for the furtherance of an innovative approach to programming reconfigurability. One such solution, as described in literature, promotes the implementation of a network of interoperable robots, with a master-slave interface. This perspective reduces programming time by temporarily cutting out the middle man, i.e. the programmer. It considers the general approach of humans physically programming a machine, and proposes a system of machines programming each other; given that one machine i.e. the master, has already been programmed. With this approach, however, although programming time can be reduced, reconfigurability still requires an expert to be physically present with the robot he wishes to reprogram. This leaves minimal room for adaptation, especially if experts are off-sight, or if the equipment and software used is not native. Self-sustainable robots refer to an area of communication that doesn’t require experts to carry out tasks such as software updates for changing products. If self-sustainability is achieved, critical issues of reconfigurability could be alleviated. These issues include automatic compensation to limiting variables, adaptability to new processes and ease of programming. Human interaction would still be required; however, off-site programming could be achieved through humans and machines communicating over the internet. Additionally, internet platforms could facilitate global robot communication and information sharing between local robotic systems [15].

2.4 Communication Networks

The research that follows in this section was conducted as a branch of the prior research conducted on programming time of robotic systems. The research proceeded to evaluate modern and common communication networking mediums, their general structure, and their influence on Advanced Manufacturing Systems. An attempt was made to gain insight into communication mediums that could possibly facilitate robotic-to-robot and human-to-robot communication.

Networks provide a platform for interoperable, integratable manufacturing systems. Delving into communication and information sharing on networks, a number of networking systems were broadly analyzed. These systems were assessed and viewed in terms of contributors to Advanced Manufacturing Systems.

2.4.1 Cloud Networks

The National Institute of Standards and Technology (NIST) defined Cloud Computing as,

" A model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction ".

Research conducted on Cloud Computing found five core defining characteristics, three service models and four deployment models [16]. Given the appealing characteristics of Cloud Computing, technology advances in adopting Cloud Computing into Manufacturing. Thus, creating a Cloud Manufacturing environment [17]. NIST further proceeds to define the characteristics of Cloud Computing to be; Broad network access, On-demand self-service, Rapid elasticity, Resource pooling, and Measured service. These characteristics are further elaborated on in Table 2-5.

Table 2-5: Cloud Computing Characteristics [18]

Characteristic	**Description**
On-demand self-service	In the absence of a service provider, end users have the ability to acquire computing features such as storage and server time. Prior to Cloud Computing, additional computational features required either additional hardware procurement or manipulating existing servers.
Broad network access	Cloud systems facilitate communication over the Internet. Given this medium, end-users have a variety of access options at their disposal. Much like an email login; given the supporting hardware, software and authorization credentials, end users have the ability to access cloud systems anywhere in the world.

Rapid elasticity	End-user provisioning capabilities tend to be unlimited, with the ability to appropriate such provisions in any quantity, at any time.
Resource pooling	End- users have the ability to be assigned or reassigned physical or virtual resources as per demand. The provider pools computing resources for multiple customers using a multi-tenant model.
Measured service	A combination of infrastructure management, optimization of workloads, self-service management, and monitoring resources provides a reason for cloud adoption. Resource control and optimization are automated by leveraging a metering capability – appropriate to the type of service being used.

2.4.2 Cloud Service Models

Cloud Computing can be viewed as having three service models; that being Platform-as-a-Service (PaaS), Infrastructure-as-a-Service (IaaS) and Software-as-a-Service (SaaS). A brief description of these service models are listed below [19]:

- *Platform as a service:* This model supplies a platform for developers to create applications and software.
- *Infrastructure-as-a-service:* This model supplies computing resources like storage, servers and other peripherals which can be acquired as a service.
- *Software-as-a-service:* In this model, the initial development of applications and software takes place on the platforms provided by the PaaS layer.

The aforementioned models have an associated definite structure. As a whole, these models form a Cloud Computing stack. The Cloud Computing stack is depicted in Figure 2-2.

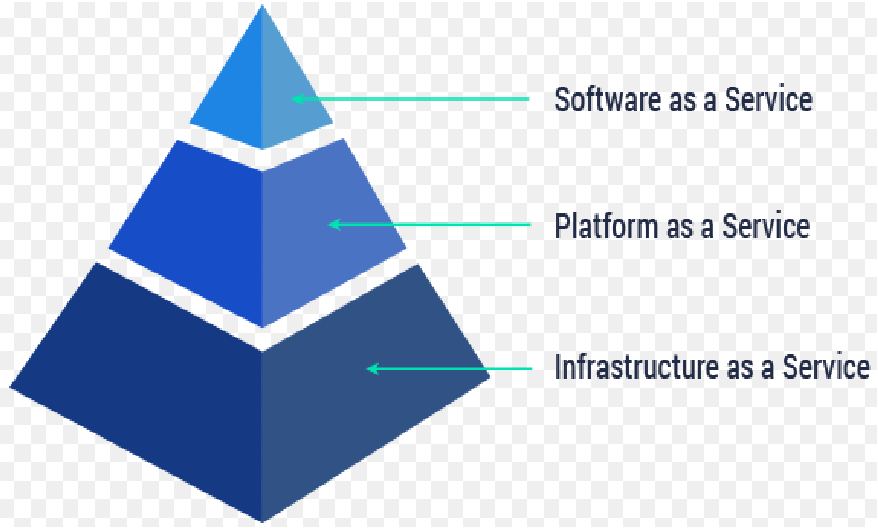


Figure 2-2: Cloud Computing services provided [20]

The service models provided by Cloud Computing are accompanied by a variety of functional providers; though these providers are not consistent for every layer of the cloud. Table 2-6 provides a comparison of the Cloud Computing stack. Each stack layer is compared to the subsequent layer, and likely functional providers are compared across the structure.

Table 2-6: Functionality comparison of Cloud Computing service models [21]

Attributes	**SaaS**	**PaaS**	**IaaS**
Service providers	Google apps, office live	Azure, Netsuite	IBM, Amazon.
Runtime management	By the customers	By the vendor	By the vendor
Data management	By the customers	By the developer	By the vendor
Application management	By the customers	By the developer	By the vendor

Used by	Business Users	Developers and deployers	System manager
Visibility	End users	Application developers	Network architects
Type of services	Dynamic infrastructure service	Integration as a service	Dynamic application services
No of providers	Large numbers of application in the cloud	Few cloud platforms	An elite group of providers
Server management	By the vendor	By the vendor	By the vendor

2.4.3 Cloud Computing Deployment Models

Deployment in this context refers to a means of accessing cloud service. Methods of deployment can be divided into four categories. These methods were found to be client dependent and used based on the need requirements of the client. The deployment methods found in research included; Private, Community, Public, and Hybrid Cloud distributions. An elaboration of these methods are provided below [22].

Private cloud: This model provides for the exclusive use of the cloud infrastructure for a single organization comprising of multiple business units. This type of model may be managed, operated and owned by the organization itself, third-party organizations, or a combination of both. This type of model may, or may not reside on the premises intended for operation.

Examples of Private cloud: VMware and Amazon EC2

Community cloud: This model allows for the exclusive use of the cloud infrastructure by a specific community of consumers. Application of this model is suited for consumers from organizations that have a shared mission, security requirements, policies, and compliance considerations. Ownership, management, and operations are carried out by single or multiple organizations in the community, a third party organization or a combination of the two. Additionally, as was the case for a Private cloud, this model may, or may not reside on the premise intended for operation.

Examples of Community cloud: Google Apps for government and Microsoft Government Community Cloud

Public cloud: This cloud model is for general public use. Ownership, management, and operation are carried out by businesses, organizations, academic institutes, and combinations of these. This model resides on the site of the intended operation.

Examples of Public cloud: Blue Cloud by IBM and Azure services platform by windows

Hybrid cloud: The Hybrid cloud model is composed of two or more distinct cloud infrastructures. These could be Private, Community, or Public infrastructures, with each of the distinct infrastructures still remaining unique entities. However, these infrastructures are bound by proprietary or standardized technology, thus allowing for application and data portability.

Examples: CTERA and Red hat open hybrid cloud

2.4.4 Cloud Computing in Manufacturing

Manufacturing systems designed for high levels of reconfigurability generate high volumes of data. As such, the efficiency of the reconfigurable system largely depends on how its data is processed and analysed. Essentially, Cloud Computing provides for the transition of computation. The transition shifts computation, management, and operation of computing resources (Hardware and software), to renting these resources. Cloud Computing could aid in manufacturing in areas such as;

- Low machinery start-up costs – In the absence of embedded computational requirements of a machine, low-cost systems can be designed to carry out operations alone and not computation.
- Scalability and ease of access – Given the platform of resource pooling of Cloud Computing being over the internet, an increase or decrease of computational needs can be done based on a machine’s needs requirement.
- Lower risk on resource provision – By outsourcing the computational workload, areas of concern such as miscalculations and computational machine shut are eliminated, as these are areas allocated to cloud experts.

Additionally, Cloud Computing has the ability to boost the efficiency of procedures in manufacturing that require a large amount of computational effort, such as modelling during the design process [23]. Adopting Cloud Computing into manufacturing has two approaches. The first being directly offering manufacturing software on the cloud as a service, and secondly; providing manufacturing capabilities as an entirely new type of cloud service [24]. The general approach to data transmission from the manufacturing environment to the cloud can be viewed in Figure 2-3. Sensor data is sent over the internet to the cloud. After processing, recipients of the processed data can be any machine that has access to the same cloud network.

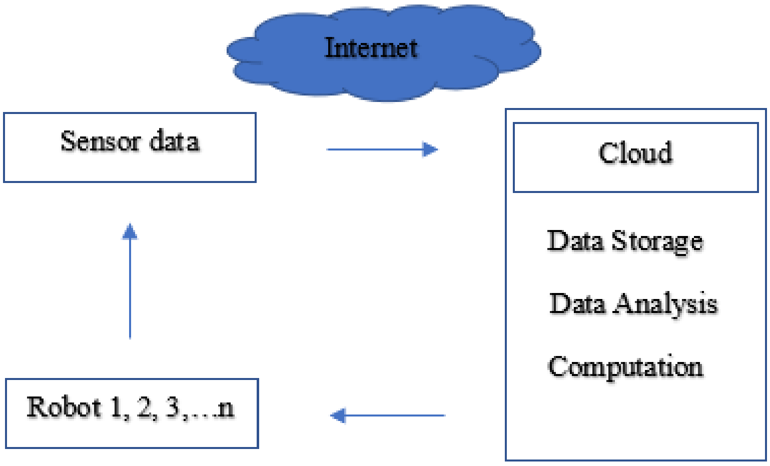


Figure 2-3: Data flow - Cloud Computing to Cloud Manufacturing

2.4.5 Wi-Fi and Bluetooth Communication

Wireless technology is designed at the outset for rapid reduction in implementation time and costs, and reduction in safety hazards created by wired networks. In 1997, Wi-Fi (Wireless fidelity) was introduced by IEEE 802.11 standards. This allowed users to connect to the internet from any geographical location. This service, however, was quite a costly affair until 2002, in 2003, a new 802.11g was released that allowed internet access from wireless access points as we know it to be today [25]. Bluetooth, on the other hand, is a wireless technology standard based on a radio system. It is designed for short-range communication. General data exchange spans a distance 10 – 100m depending on the class of Bluetooth used. Bluetooth specifications are governed by the Bluetooth Special Interest Group (SIG) and is an open specification. Both Wi-Fi and Bluetooth facilitate a platform for wireless communication, the main difference between the two lies in their design purpose. Primarily, Bluetooth is used for connecting devices without the use of cables, whereas Wi-Fi provides devices with access to the internet [26].

2.4.6 Wireless communication for Advanced Manufacturing Systems

The industrial landscape has been reshaped entirely by the Fourth Industrial revolution (Industry 4.0). The exponential rate at which technology advances requires constant change, and adaptability of existing systems and frameworks [1]. One of the major challenges for the future of manufacturing is dealing with collaborative systems of hybrid human and robotic operations, additive and subtractive processes, metal and composite materials, and cyber and physical systems. Focusing on cyber-physical systems, these systems have proven to be efficient in operations such as monitoring, analysing, automation and communication. Understanding this, it is easy to grasp why the issue of wireless connectivity of manufacturing systems needs to be focused on. Wireless communication and sensor-based technology are largely contributing to Industry 4.0 and the cyber-physical environment. This being in an attempt to create an intelligent, smart working environment [6]. Smart working environments, however, associate themselves with a variety of complications. That being, real-time data acquisition, storage, and data analysis using machine learning techniques [27]. As such, many researchers are moving towards cloud management. In turn, this imposes three key problems. That being; implications and costs associated with virtualizing resource management, life cycle management, and scheduling of cloud resources. Solutions to industrial problems of today present development of machinery with relatively high complexity. This essentially contributes to incurring additional costs for implementation, costs associated with training skilled labour, and costs associated with skilled persons carry out maintenance operations when required. Traditional approaches to increase production efficiency need to be re-evaluated from its traditional sense. A likely solution to this problem can be extracted from analysing communication environments. Communication environments can be viewed as having two sub-divisions. Human-centered and cyber-centered. With the increasing complexity requirements of machines, tasks of reconfigurability, adaptability, and flexibility can be allocated to humans to perform. This would essentially reduce the processing requirements and level of complexity required of a machine in carrying out a complex task [28].

2.4.7 Voice interface communication

The field of voice assistants is a relatively new communication paradigm. Understanding the general communication approaches outlined in the aforementioned research, voice interfaces provide a new and vibrant approach to communication mediums. Speech processing robots such as Pepper and voice interfaces such as Amazon Alexa, OK Google, Siri, and Cortana move away from the traditional screens, keyboard, and mouse communication tools. They provide a natural communication medium which has consequently contributed to their rapid rise in global markets. However, due to the fact that these systems are relatively new, they do present a variety of complications [29]. These complications include:

- Voice interfaces requiring highly fluent sentences for efficient processing,
- The inability of voice interfaces to effectively interpret a sentence pause,
- A high level of sensitivity to environments with high levels of noise, and
- The inability of voice interfaces in constructing responses based on ambiguous commands

The aforementioned complications presented are largely attributed to human speech. According to George Mahl, on average, human emit discontinuities in speech every 4.4 seconds. This action is a result of humans occasionally stopping to think, as an example [30]. Research does, however, present methods to facilitate fluent speech transmission. One such method involves voice interface companies incorporating automatic speech recognition and natural language understanding to comprehend if a user command is complete. Figure 2-4, as adopted from Amazon, illustrates state changes by the inclusion of a speech recognizer component. The connectors give an indication of the state transitions and the boxes represent the speech recognizer.

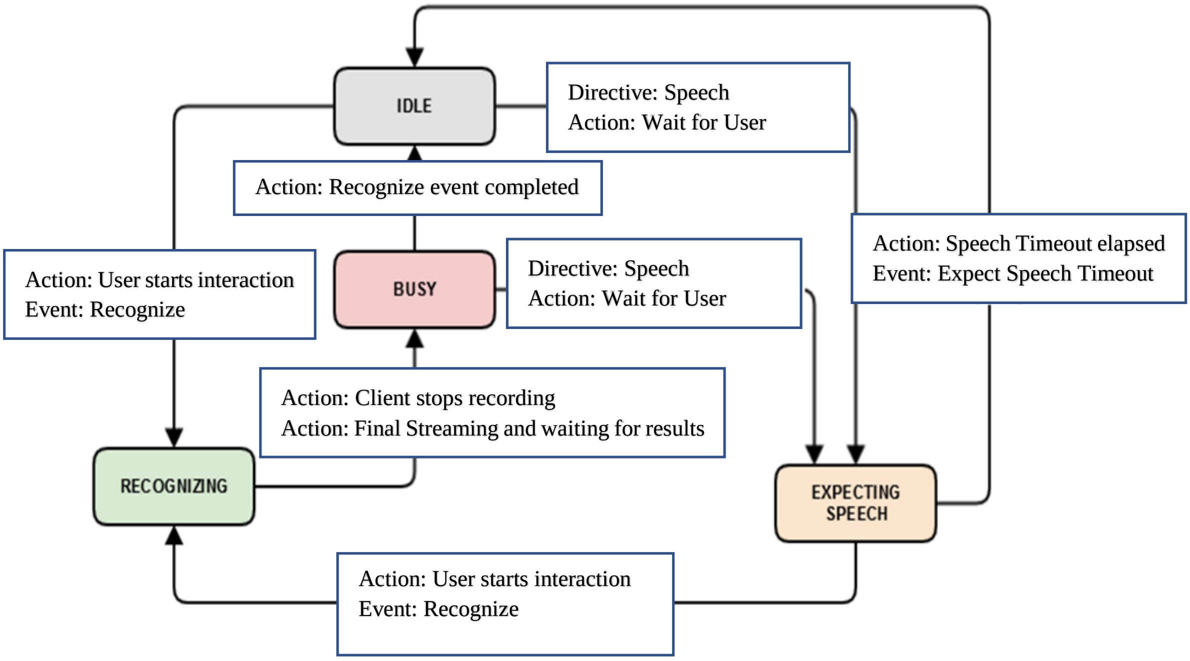


Figure 2-4: Amazons state changes in voice interfaces using speech recognizer [30]

Other systems attempting to close the communication gap between voice interfaces and humans display methods that showcase the relatively new nature of voice interfaces as a communication medium. Such approaches include using a push button as a medium to identify sentence pauses and designing voice interfaces at the outset to output information based on specific words as command inputs, rather than complicated sentences [31].

Applications of Voice Interfaces

Many applications of voice interfaces present themselves outside the manufacturing environment. Though, if implemented in manufacturing, these communication mediums could prove to be highly beneficial. Iqbal et al presents the use of voice interfaces for controlling electrical devices.

This approach focuses on improving the lives of disabled individuals in a household. An overview of the approach is depicted in Figure 2-5 [32].

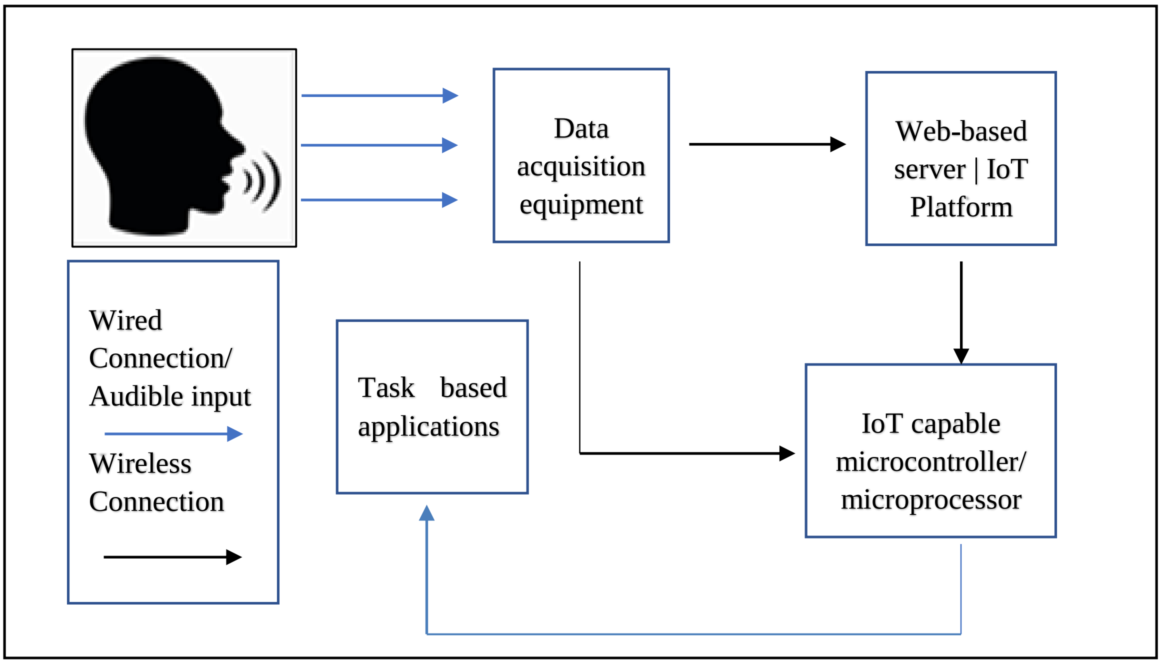


Figure 2-5: Smart household voice interface [32]

Figure 2-5 represents a voice, or optional text command being sent to Amazon Alexa. Thereafter, the service interprets the input voice command and transmits a specific General-Purpose Input/output (GPIO) to the pins on a Raspberry Pi. Appliances in this layout receive commands from the Raspberry Pi wirelessly, alternatively, a wired connection could be made for smaller layouts. Additionally, connectivity between regular appliances and Alexa and formed using Alexa-enabled smart plugs.

Applications of voice interfaces can be viewed as being quite diverse. Kandhari et al presents a paper that accentuates on a cloud-based Speech Recognition System (SRS) for e-commerce applications. The research proposed a taxonomy of SRS and presented a voice-controlled e-commerce application using IBM Watson speech-to-text service. Additionally, a comparative study with other speech-to-text systems such as Google and Amazon were carried out. Essentially, a web-application was presented that receives a voice prompt, the input gets converted to a text, the meaning is extracted from the text, and a wide variety of tasks are performed thereafter (searching, writing text, reading, and browsing). The taxonomy presented by Kandhari et al is depicted in

Figure 2-6. The taxonomy is divided into four categories; type of speech, vocabulary, style of speech and speaker model. As can be viewed in

Figure 2-6, the categories are subcategorized into a set of limiting constraints. The constraints provide a method of extracting sentence fluency from an input [7].

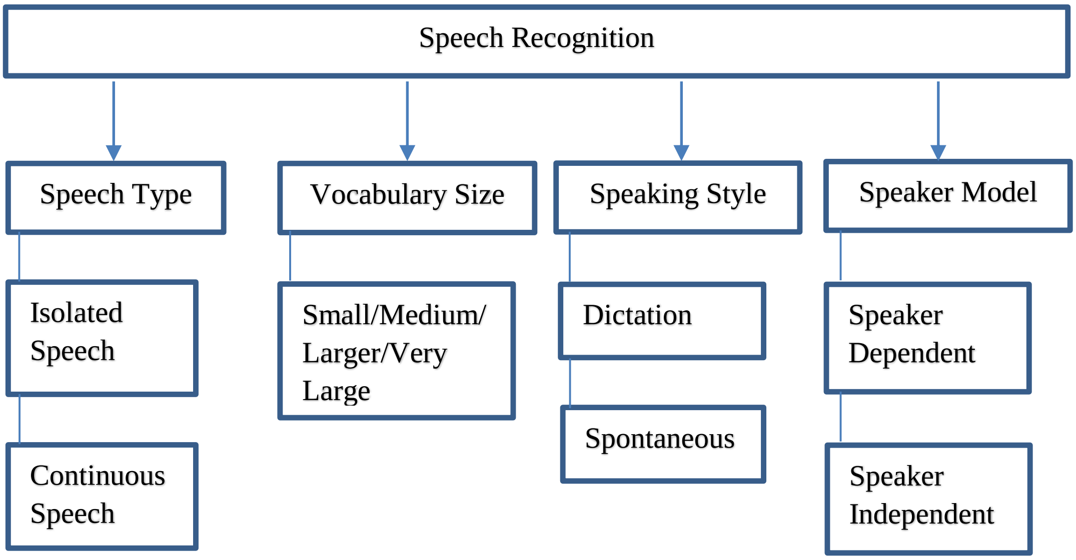


Figure 2-6: Taxonomy of SRS [7]

The quality of results presented was determined using the phrase recognition rate (PRR), as defined by the National Research Council - Equation 1, and the word error rate (WER) using Equation 2.

Equation 1: Phrase recognition rate [33]

$$PRR = \frac{Number\ of\ words(N) - Substitutions(S) - Deletions(D) - Insertions(I)}{Number\ of\ words(N)}$$

Equation 2: Word error rate [34]

$$WER = \frac{Substitutions(S) + Deletions(D) + Insertions(I)}{Number\ of\ words(N)}$$

Figure 2-7 presents the Web Application Architecture for a voice-controlled e-commerce application using IBM Watson speech-to-text service. With the help of microphones in a computer system, voice commands are given to the application. Input signals are streamed through Watson Speech-To-Text (STT). The service then converts the audio signals to text. In the situation of the application not identifying the input words, Watson Text-To-Speech (TTS) service can be used to generate a voice response to inform the user about the error.

Web applications form an essential platform for information sharing and product-customer interactions, i.e Digital Marketing. In understanding this key feature of the modern age, SRS could be used to attract web audiences in the same way online chat programs are used. Though, as displayed in online chat programs, barriers exist for the visually impaired. Additionally, web-accessibility standards created by the World Wide Web Consortium (W3C), Web Content Accessibility Guidelines (WCAG 2.1) are not sufficient to address all accessibility problems (Akram and Bt, 2017). Therefore, SRS could enable people with physical and learning disabilities to interact with web applications at their own pace with comfort and ease.

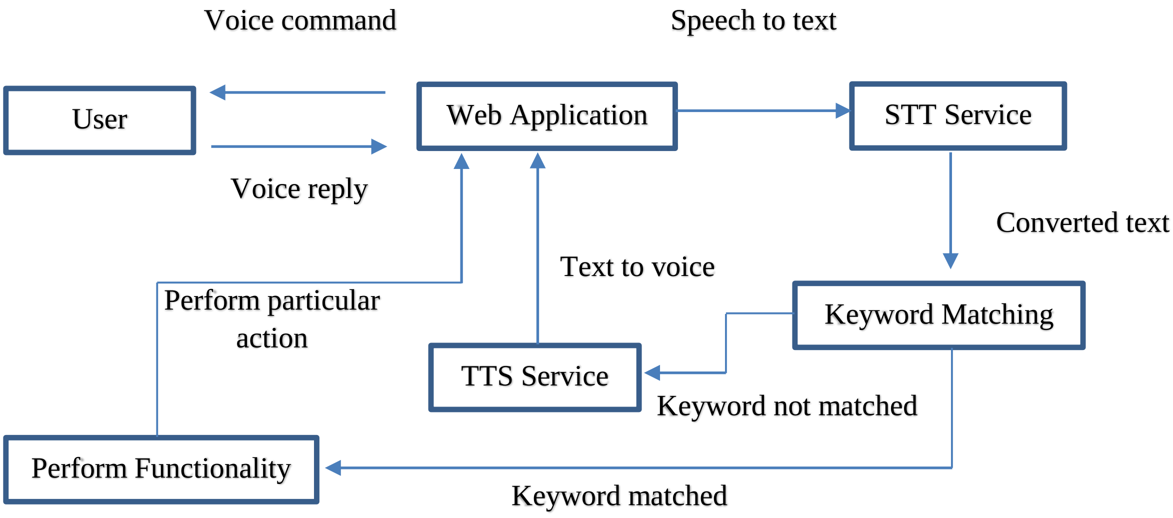


Figure 2-7: Web Application Architecture

2.4.9 The field of Pattern recognition

Pattern recognition is a fast developing, exciting field that acts as a supporting structure to fields such as image processing, text and document analysis, neural networks, computer vision, and image recognition. Additionally, it very closely relates to machine learning and is highly applicable in areas such as bioinformatics, multimedia data analysis, biometrics and as of very recent, data science [35].

Delving into image recognition, in a very simple way, image recognition can be considered as the process of detecting and identifying objects or features from a digital image or video. Applications of image recognition can be found in security surveillance, tool booth monitoring, factory automation and many more. Typically, in the process of identifying images, various algorithms are designed to carry out a variety of recognition operations. These algorithms include:

- Optical character recognition,

- Pattern matching and gradient matching,
- Face recognition,
- License plate matching, and
- Scene identification or scene change detection

When approaching an image recognition problem, machine learning and deep learning prove to be very useful. In due respect to machine learning in image recognition, this method involves the identification and extraction of features from an image as input information for a machine learning model. Whereas for deep learning, this approach uses convolutional neural networks to learn key features from an image automatically, and automatically identify those features when presented with a new image [36].

Facial recognition (FR), as one application of image recognition, has been quite an interesting area of research over the past few decades. Applications of FR can be found in access control systems, surveillance systems, personal authentication, finance and many more. Though, despite the growing adaptations of FR in society, reliability hinders FRs growth and progression into a variety of different ways. The major factors contributing to this hinderance include;

- Degradation of facial image quality – The factors contributing to this being blurriness, misalignment, pose variation and varying illumination conditions, and
- A mismatch between training data obtained as compared to the application data received.

With applications of FR for video sources, poor quality image sequences are responsible for performance degradation [37]. Kim et al proposes a new learning-based face image quality assessment for reliable and robust face recognition. The contribution involved analysis methods through the definition of objective face image quality (OFIQ), and relative face image quality (RFIQ). The OFIQ directly relating to visual qualities in reference to blurriness, misalignment, pose variance and varying illumination conditions. RFIQ, on the other hand, gave relation to mismatch discrepancies [38]. Experimental results showed that with the implementation of the proposed systems such as the OFIQ and RFIQ, identification rates increased significantly. The results were compared to a proposed learning-rank-based quality framework for assessing face image quality (RQS) [39]. Results of the experiment are depicted in Figure 2-8. An additional experiment carried out by Kim et al evaluated the performance of the proposed methods of OFIQ and RFIQ against image acceptance discrepancies. The methods were assessed on making verification assessments. The experimental results, as can be viewed in Figure 2-9, showed a decrease in the error rate using the proposed RQIF and OFIQ methods.

Figure 2-8: Identification rate based on proposed OFIQ and RFIQ [38]

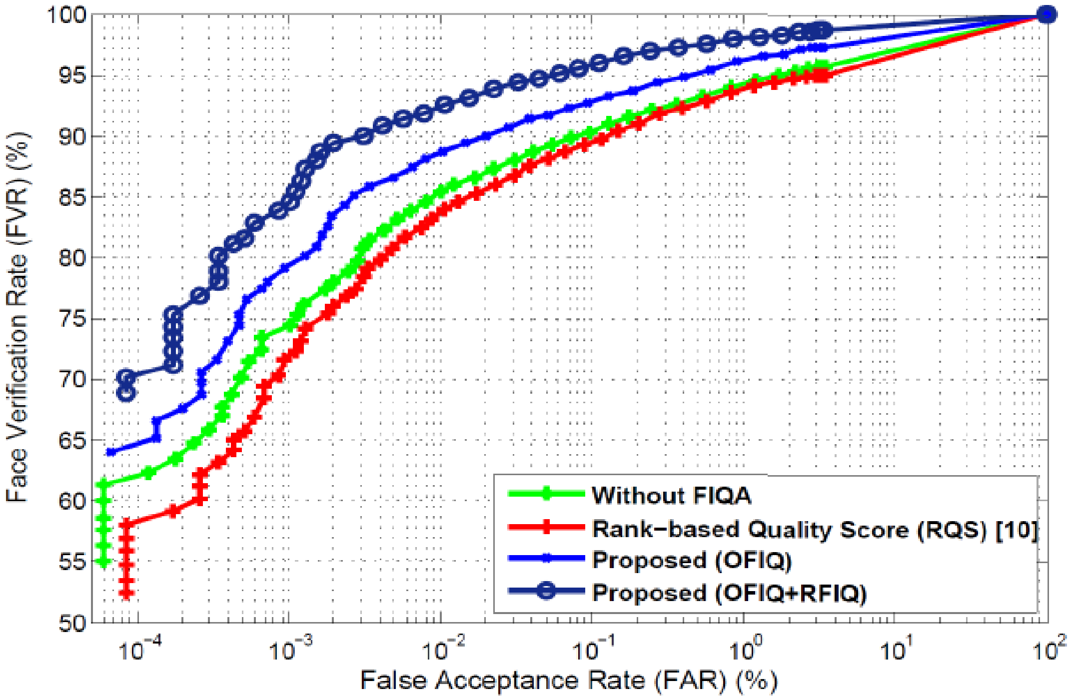


Figure 2-9: Face verification rate vs False acceptance rate [38]

An evaluation of additional research revealed face selection techniques using a patch-based probabilistic image quality assessment, for video input content. Research for this technique focused on introducing a metric for Face image quality (FIQ). This approach enabled the analysis of a variety of quality factors such as sharpness, scale, illumination, variations in pose and alignment. The analysis of which was represented as a probabilistic similarity. Additionally, frameworks for visual quality evaluation exist in a variety of research papers. The frameworks of which, offer learning methods for a ranking-based feature method, essentially providing improved visual quality for image recognition tasks [39].

Further research conducted found that the performance of image recognition depends not only on algorithms but image factors such as image regions and essential image elements. Wang et al presented an essential element-region (EssER) recognition method and carries out a series of experiments to estimate elemental differences which include texture, part, structure, and optical field. Results from the elemental differences experiments showed that although an optical field contributes quite largely to the elemental analysis, the parts of an image plays a very pivotal role in image recognition. The elemental analysis used a Deformable Parts Model (DPM), and a Convolutional Neural Networks (CNN) in the analysis. Results are depicted in Figure 2-10.

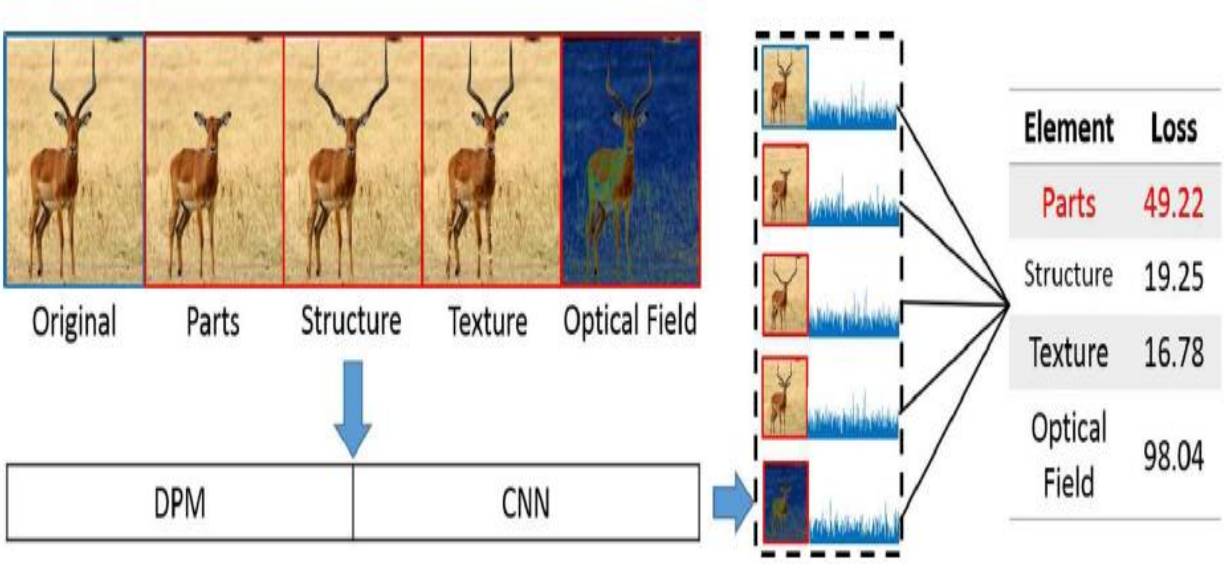


Figure 2-10: Approach to finding essential elements from an image [39]

2.5 Cobots in Manufacturing (Collaborate Robots)

The research in this section was conducted due to the combined knowledge gathered from previous sections. A factor largely contributing to the research carried out in this section related to past research dictating a dependency of progression in manufacturing being human-robot *collaboration. The section that follows evaluates the state of human-robot collaboration and its implications on Advanced Manufacturing systems.*

Cobots (Collaborate robots) is a category of robotics that is designed at the outset to interact and work together with humans. As manufacturing systems grow and develop, it becomes seemingly difficult to incorporate complex flexibility and reconfigurability requirements into manufacturing systems attempting to transition into customizable manufacturing. Combining the strength of robotic systems with the problem solving and dexterity traits of humans, it becomes possible to solve tasks that cannot be fully automated, essentially improving the working conditions, employment opportunities and production quality of manufacturing systems [40].

A key consideration when working with cobots is task distribution, essentially, trying to understand the task allocation between robots and humans. This is an integral part of effective collaboration. The distribution will need to account for the inherent strengths of humans and robots and provide tasks to these members associated with their strengths. One such task planner developed by Roncone et al provided for task allocations and role assignment. When interactions were investigated, it was found that interactions were mainly based on graphical user interfaces and buttons, with general trends considering multi-modal interactions, and gesture expressions being used to counteract the problems presented in noisy environments [41].

One of the main problems presented with cobots is that research is confined to the student and laboratory environment, with less emphasis being placed on introducing people active in the industrial domain. This presents additional problems such as lack of knowledge for implementation in industry and industrial acceptance. This needs to change if cobots are to be introduced into industry. Communication with industrial work-men will provide critical insight into the industrial domain and will undoubtedly provide useful insight into potential human-robot collaborations [42].

2.5.1 Applications of cobots

Incorporating cobots into manufacturing can be highly advantageous. Areas of growth can be seen from the implementation of cobots in Brussels, Belgium. in this particular case, cobots were developed and introduced into an automated assembly line. The assembly line manufactured Audi A1 vehicles. For the production of an Audi A1, 550 industrial robots were used in the body shop, 30 in the paint shop and 6 in the assembly line. Brussels attempted to evaluate the potential of cobots for improving the manufacturing process. The constraining factors presented in the assembly process at Brussels was as follows:

- Products have very high variability, this being due to a variation in product customizability's,
- Materials used for production had a very high sensitivity,
- Automation is challenging due to the complex nature of product assemblies, and
- Injuries and workload challenges associated with an aging workforce.

Audi Brussels tried to focus on using a cobot for applying glue to reinforced plates for supporting the roof racks of their vehicles, with humans focusing on workpiece arrangement. This operation was focused on due to the inconsistencies presented by manual gluing by workmen. This was a serious concern due to products of lower quality being produced. The cobot implemented, made use of gestures for communication due to the noisy environment. Gestures were captured using a RoboSense camera and proved to be easily detectable with consistent performance. Face recognition was also used incorporating neural network and deep learning for operator recognition.

The concluding results showed that cobot Walt contributed to the improved quality of cars produced. With gluing quality being based on flatness parameters, results showed a reduction of 15% for cars not meeting the flatness criteria, additionally, the quantity of overall glue used (as compared to humans carrying out the task), decreased by 20% [43].

Due to the limiting factors such as confinement of cobots to laboratories and students, applications in the industrial environment do not readily present themselves. However, advances are being made in the field of cobots in relation to a more advantageous human-robot working environment. A few examples of these advances that present themselves include;

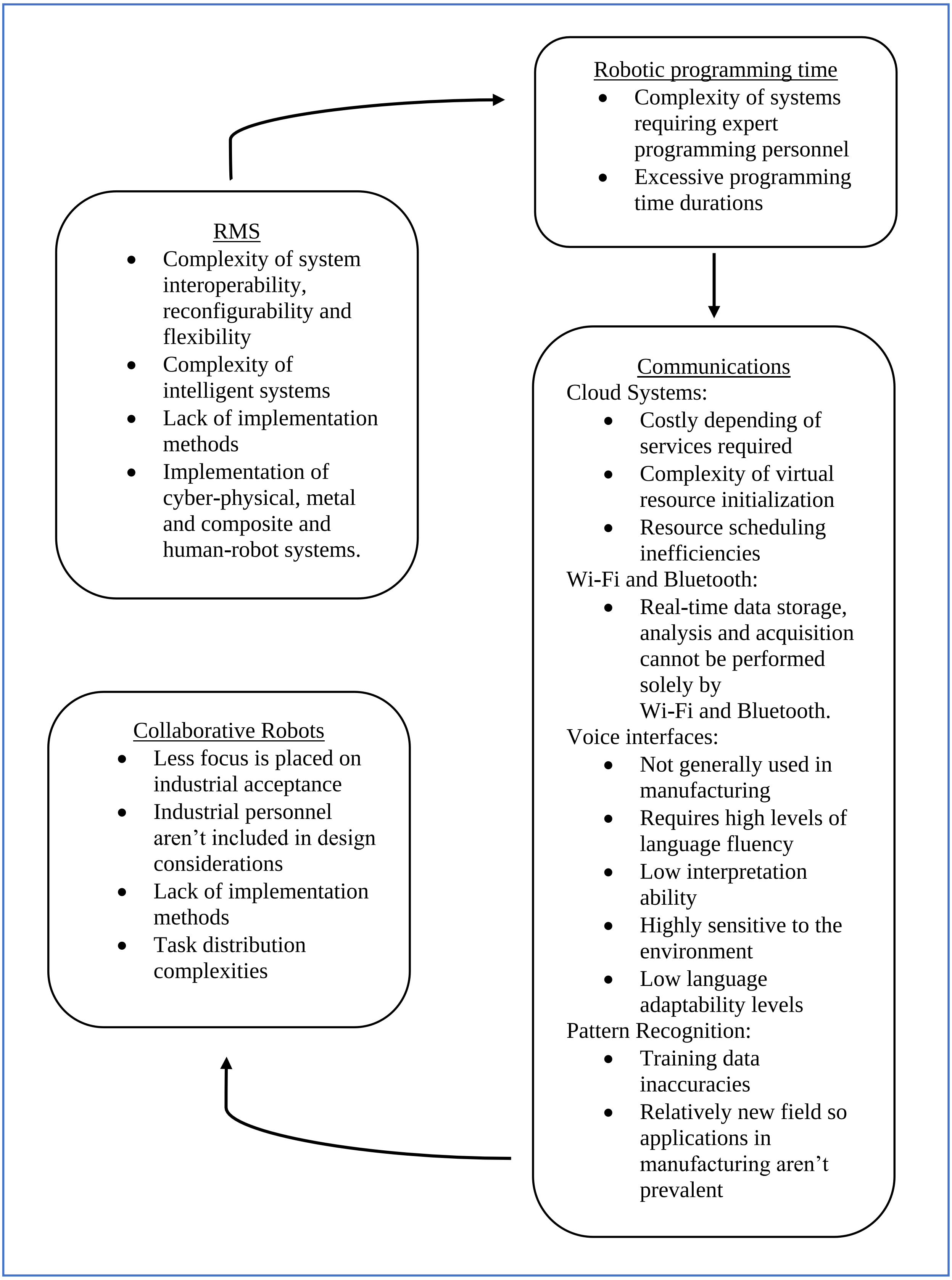
- Development of a safe and performant control system for compliant robotic joint manipulators – In relation to joint manipulators, two problems present themselves, firstly, the elastic elements of the actuators have the potential to store energy that could be potentially dangerous. Secondly, nonlinearities and uncertainties are introduced by the compliance [44].
- The development of new actuator system that allows for haptic interfaces for collaborative robots – In industrial environments, robotic systems are very insensitive to their environment, though, for cobots, close cooperation between humans and robots needs to be established. As such, force sensing and control capabilities for interferences is required for stable interactions between operators, cobots, and the environment [45].

2.6 Chapter conclusion

The Literature Review Chapter summarised the key focus areas from the Advanced Manufacturing Systems. A summarized view of potential areas of growth and improvement for Advanced Manufacturing Systems can be viewed in Figure 2-11.

2.7 Chapter summary

Chapter two displayed the dissertation Literature Review. Four main areas were covered. These areas included; Reconfigurable Manufacturing Systems, Programming of Robotic Systems, Communication Networks and Collaborative Robotic Systems. The Literature Review chapter presented a variety of concerns of voice interface technology, these concerns included; a lack of implementation in manufacturing environments, high levels of language fluency requirements, low interpretability constraints, having a very high sensitivity to the environment and a low adaptability to different languages.



Robotic programming time

- Complexity of systems requiring expert programming personnel
- Excessive programming time durations

Communications

Cloud Systems:

- Costly depending of services required
- Complexity of virtual resource initialization
- Resource scheduling inefficiencies

Wi-Fi and Bluetooth:

- Real-time data storage, analysis and acquisition cannot be performed solely by Wi-Fi and Bluetooth.

Voice interfaces:

- Not generally used in manufacturing
- Requires high levels of language fluency
- Low interpretation ability
- Highly sensitive to the environment
- Low language adaptability levels

Pattern Recognition:

- Training data inaccuracies
- Relatively new field so applications in manufacturing aren’t prevalent

Collaborative Robots

- Less focus is placed on industrial acceptance
- Industrial personnel aren’t included in design considerations
- Lack of implementation methods
- Task distribution complexities

RMS

- Complexity of system interoperability, reconfigurability and flexibility
- Complexity of intelligent systems
- Lack of implementation methods
- Implementation of cyber-physical, metal and composite and human-robot systems.

Figure 2-11: A summary of potential areas of growth and improvement for Advanced Manufacturing Systems

3. METHODOLOGY

3.1 Introduction

The Literature Review chapter presented a variety of problem areas within the Advanced Manufacturing environment. The fields explored included Reconfigurable Manufacturing Systems, Robotic systems programming time, Communication Networks and Collaborative Robots. From the areas of concern presented in Figure 2-11, and considering system requirements of flexibility, adaptability and reconfigurability of machinery for customizable manufacturing, the avenue opted to proceed along was flexibility associated with the fourth industrial revolution. More specifically, the focus was placed on cyber physical systems due to conventional systems being migrated to a digital platform. Flexibility through communication platform was selected due to its ability to provide a conventional shift of tool-centered manufacturing to a new, and more versatile approach. The Literature review chapter presented a variety of concerns of voice interface technology, these concerns included; a lack of implementation in manufacturing environments, high levels of language fluency requirements, low interpretability constraints, having a very high sensitivity to the environment, and a low adaptability to different languages.

The Methodology chapter that follows presents the approach used in an attempt to address the aforementioned concerns. It should be noted that the approach considered was due to the field of voice interface technology being relatively new, and though continuously advancement persists, data on methods of progressive improvements remain proprietary to the technological institutes facilitating advancement. As such, the methodology chapter that follows gains its structure from a variety of different facets; these include; literature, information gained from enrolment into courses facilitated by industrial experts, attending a workshop facilitated by senior research expert in the field of voice technology, interviews with industrial experts and consultation with industrial engineers. Additionally, the methodology chapter outlines the methods and validation for the testing and procedures carried out in the dissertation.

As a whole, the research attempted to understand the reasons for absence, and implication of voice interface technology in manufacturing environments, with the key purpose being to investigate the performance of leading voice interface technology when subjected to manufacturing environments. Additionally, research progressed with the optimization of leading voice interface technology (“OK Google”), so as to allow for a feasible means of implementation into manufacturing environments, with focus also placed on integration of the interface technology with mobile robotics to facilitate industrial integration.

Research Question: How can a mechatronic system be designed for a manufacturing environment that integrates voice interface technology and mobile robotics?

3.2 Methodological structure - Senior research expert

The first issue sought to address was to establish the state of voice interface technology from the perspective of an academic expert. As such, Dr. Alexandra Birch was approached. Dr. Birch is a Senior Research Fellow from the Institute for Language, Cognition and Computation (ILCC) in the School of Informatics at the University of Edinburgh. A machine learning workshop facilitated by Dr. Birch was attended at the University of KwaZulu-Natal, and following the workshop, a dissertation overview was presented to Dr. Birch, and an interview was carried out. The main areas of the interview can be viewed below [46]:

1. It was suggested, that due to the nature of the field of voice interfaces being relatively new, and major developments and data resources being proprietary, it would be a good first step to identify a benchmark of the current operational capabilities of voice interfaces for manufacturing.
2. It was suggested that synthetic data be created – from a commonly used device facilitating voice interfaces – this statement was made to establish the limitations and areas of potential improvement for voice interfaces, and to conform to the selected area of research being flexibility.
3. For possible optimization of voice interfaces relating to manufacturing environments, it was suggested to simulate a manufacturing environment and subject the synthetic data to manufacturing conditions.
4. It was also highly recommended, that for the issue of adaptability of voice interfaces to different languages, it would be a good idea to look at web-based services.

3.3 Methodological structure - Industrial engineering consultant

Consultation with two Industrial system engineers sought to address the issue of lack of implementation of voice interfaces in manufacturing environments. The first consultant, Mr. V. R. Magan holds a position as a System engineer at Eskom Holdings. He holds the title of a professionally registered ECSA engineer and has ten years of engineering experience in manufacturing. The second consultant, Mr. F. Mathe acts as an ECSA candidate structural engineer at Eskom Holdings, with a rich knowledge of plant-based operations. He possesses two years of plant-based operation training. The main areas of presented by consultants were as follows [47, 48]:

1. For implementation of voice interfaces into manufacturing, focus should be placed on collaborative human-robot systems, rather than the conventional machine-centered system designs.

2. It was suggested to find a means of implementing voice interfaces into the maintenance failure process. It was established, that in a manufacturing environment, when a failure occurs, it can be a very time-consuming process to communicate instructions and equipment between the divisions of operations, maintenance and engineering in manufacturing environments.
3. Data of common manufacturing conditions such as decibel ranges in different plant regions were provided by the consultants (Appendix B: 10.5). This allowed for future subjection of designed systems to manufacturing conditions, as suggested in Section 3.2.

3.4 Methodological structure - Literature

Literature was sought after to try and establish a means of improving on the issue of requirements of low interpretability constraints. The following information proved to be useful:

1. A common reason for low interpretability constraints of voice interfaces can be directed to human speech inaccuracies. According to research, on average, humans emit discontinuities in speech every 4.4 seconds. Research showed that implementation of Web-based servers could increase the level of interpretability of voice interfaces.
2. Literature also provided for a means to address adaptability concerns through accounting for speech inconsistencies in the software design process.
3. One specific research report relating to the use of a Voice Controlled E-Commerce Web Application provided for a benchmark and method for implementation of voice interfaces within a cloud-based recognition system [7]. The report also presented data proving the best voice interface system currently in the market being “OK Google” – this statement was also validated with the interview carried out in Section 3.2.

3.5 Methodological structure – Course enrolment

The following undertaking was carried out to establish a link between voice interfaces and machinery. The main results obtained can be seen below: [Appendix A]

1. There exists a major communication barrier between voice interfaces, electronic equipment and machinery however, coupling Web-based services with IoT platforms using cloud servers provide for a very effective means of facilitating this absent communication barrier.

3.6 Methodological structure – Industrial software expert

To gain an understanding of good automation development practices for industrial machines, an expert software developer working in an industrial environment was consulted. The design approach recommended was GAMP®5. GAMP stands for Good Automated Manufacturing Practice. It is a guidance document that relates to a risk-based approach to compliant computerized systems. GAMP is widely accepted approach for automated machines and is widely used in industries in the USA and Europe [49]. The GAMP framework for a risk-based approach to computer system validation can be viewed in

Figure 3-1 [50]. The GAMP development layout includes:

- Defining user requirements,
- Establishing functional specifications,
- Establishing configuration specifications,
- Establishing design specifications,
- Building a system,
- Testing the built system,
- Testing configuration specification,
- Testing functional specifications, and
- Testing user requirements.

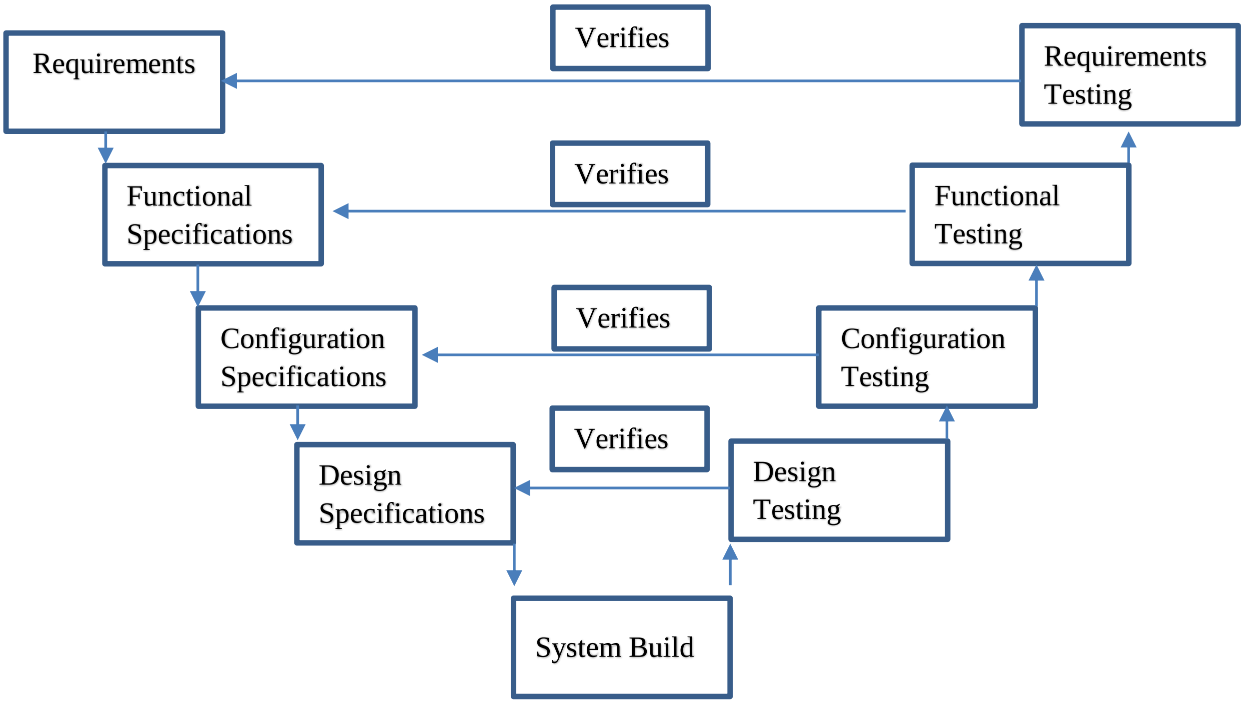


Figure 3-1: GAMP Framework [50]

3.7 System design methodology for implementation

Aligning the design of a prototype to the aforementioned research question, it was proposed to design voice controller mobile navigation robot. The main focus of the design being on optimization of voice interfaces for manufacturing environments, with a means of implementation being facilitated using a mobile robot design. The mobile robot design was considered to first facilitate implementation of voice interfaces into manufacturing, and to also address the issues of less focus being placed on industrial acceptance and less focus being placed on using industrial employees with innovative systems in carrying out designated tasks. This concept stemmed from the considerations taken from literature, Chapter 2 - "Concerns of collaborative robots", and consultation with industrial experts – Section 3.3.

3.3.1 User requirement specifications

To establish, and optimize current voice interface systems alleviate the following issues:

- A lack of implementation in manufacturing environments,
- High levels of language fluency requirements,
- low interpretability constraints,
- having a very high sensitivity to the environment, and
- a low ability to adapt to different languages

With due consideration given to industrial application, and with information provided in Section 3.3, the design of a mobile navigation robot was considered, that capabilities of the robot focused on issues presented in Section 3.3 such as human-robot collaboration and process optimization, the requirements of the mobile robot can be viewed below:

- Interact with humans in the manufacturing environment using a voice interface,
- Interact with the manufacturing environment through a global navigation platform, and
- Is able to assist individuals in manufacturing environments.

3.3.2 Functional specifications

- The mobile robot must have an independent voice interface system, and
- The mobile robot must have an independent navigation system.

3.3.3 Design specifications

- The mobile robot must be integrated with voice interface technology into manufacturing environments, with the ability to act as an assistant to individuals in manufacturing environments.

- The mobile robot must be able to move using four wheels,
- The mobile robot must not exceed the dimensions of 30cm x 30cm x 30cm for testing purposes and,
- The mobile robot must be designed to hold and transport tools/parts.

3.3.4 Configuration specifications

- Design a mobile robot that is able to integrate a voice system and a navigation system.
- Due to the proposal being a conceptual approach, the layout for testing should ideally be the Advanced Manufacturing Laboratory at the University of KwaZulu-Natal.

3.8 Chapter summary

Chapter three displayed the Methodology Chapter of the dissertation. The chapter outlined the approaches adopted to deal with the issues identified in chapter two. The chapter acknowledged that voice interface technology is a relatively new area and that finding solutions to problems identified in chapter two required gaining information and know-how from a variety of sources. These sources included literature, courses facilitated by industrial experts, a workshop facilitated by a research expert in the field of voice technology, interviews with industrial experts and consultations with industrial engineers. Methods and validation for the testing and procedures carried out was also discussed in this chapter. This chapter highlighted on integration of the interface technology with mobile robots to facilitate industrial integration. The technology proposed to implement a voice interface was to design a mobile robot that is able to integrate a voice system and a navigation system. Requirements for such a robot was discussed in this chapter.

4. ELECTRONIC DESIGN

4.1 Introduction

The Electronic Design Chapter focuses on the design of the integrated voice controlled mobile navigation system. The main focus of this chapter was to design three systems. The first system related directly to the generation of operation benchmark data when subject to manufacturing conditions. Secondly, a system was designed to optimize on accuracy and reliability conditions presented by the benchmark data and on conditions presented in Section 3.3.1. Lastly, a mobile navigation robot was designed to create a platform for implementation of voice interfaces into the manufacturing domain. The design of the aforementioned systems can be viewed in the subsequent chapter. The structure of the chapter stemmed from the methodological chapter.

4.2 Voice interface system

The following section focuses on voice communication. User requirements, functional analysis, configurations and designs are outlined and discussed.

4.2.1 User requirements

Wireless technology is integrated into working environments across the world. Technology, such a Wi-Fi promotes globally connected working environments, and allows for ease of communication for individuals in manufacturing environments. The voice interface system design was considered to allow for a flexible communication platform within modern working environments – with emphasis placed on environments that possess well-defined Wi-Fi networks, this being due to the cloud-based IoT platform implementation. The voice interface system looks to communicate voice commands over Wi-Fi networks to the proposed voice controlled mobile robotic device. Figure 4-1 outlines the key requirements of the voice interface system.

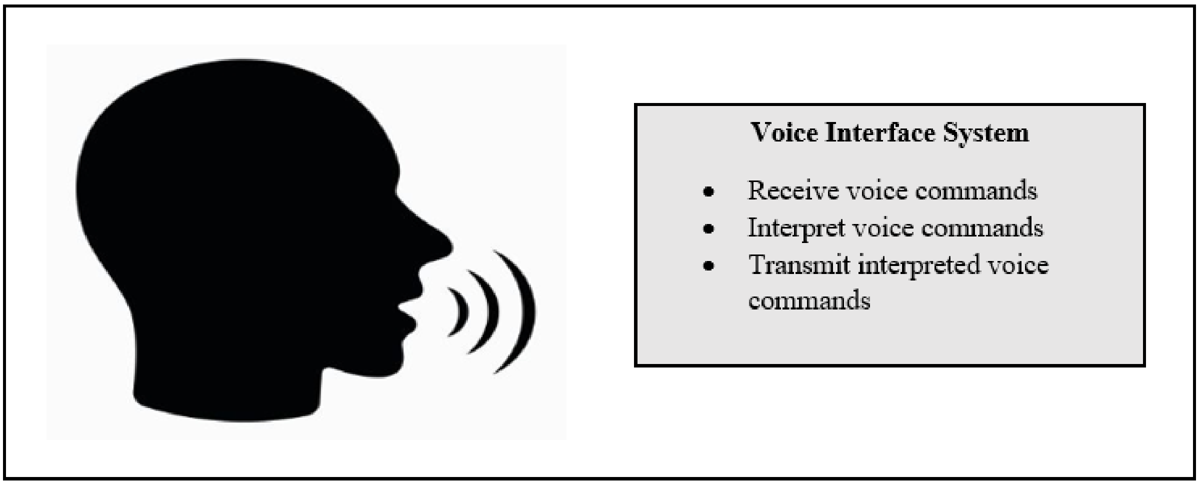


Figure 4-1: User Requirements – Voice Interface System

4.2.2 Component functional specifications

The content that follows gives a brief description the key components that were used for the design of the voice interface System:

Microcontrollers: Design considerations of system control sprouted the idea on Microcontrollers. Microcontrollers are commonly used programmable equipment facilitating a ‘brain’ for electronic systems. They provide for an inexpensive control medium for electronic systems, though programming knowledge is required. Advantages of these systems include being low cost, low power requirements, ease of procurement, and a rich variety of free learning resources. Disadvantages however, include having a low processing ability, low storage ability, and not commonly used in manufacturing environments – due to the lack of implementation methods.

Possible Alternatives: Literature of industrial systems provided insight into Programmable Logic Controllers (PLC’s), and Microcomputer’s such as the Raspberry Pi. PLC’s are very prominent in industrial applications. They are used to control complex machinery in industry and carry a very high cost factor. The design of PLC boards incorporates semi-conductors and depend of thermal characteristics. Additionally, one of the major disadvantages of these systems include expert programmers being required for system implementations. Learning resources are available, though associated with relatively high costs. The Microcomputer considered was the Raspberry Pi. The Raspberry Pi is a single-board computer developed to promote basic computer science and circuit development. The Raspberry Pi runs an independent Linux operating system for software development. It has a variety of free learning resources available online and possess a higher processing ability when compared to the Arduino Microcontroller. It is however, accompanied by additional costs and steep learning curbs to familiarize oneself with its Linux user interface.

Wireless Communication components: Component selection for the communication platform involved two main considerations, these considerations included Bluetooth technology and wireless communication through a portable router. Both Bluetooth and Wi-Fi routers are easily accessible, though, depending on the type of communication speeds, signal strength and range of communication, the price of these systems may vary.

4.2.3 Component selection matrix

The Arduino Microcontroller consists of a wide variety of variances, these include the Arduino Uno, Mega, Duo and Leonardo. These boards are accompanied by varying Digital/Analog pins, memory and processing specifications. The microcontroller selected was one that closely resembles the Arduino microcontroller and is programmed using the Arduino IDE. This microcontroller being the NodeMCU. The NodeMCU (8266) was selected due to the basic serial communication requirements of Arduino compatible wireless modules, as well as its Wi-Fi enabled capabilities. Additionally, the wireless platform selected was Wi-Fi networks through a service provider rather than Bluetooth. Wi-Fi was selected as it allows for a larger number of devices to share a network simultaneously. Bluetooth modules were also compared (HC-05 and HC-06), to allow for ease of testing of voice commands in instances where mobile service providers were absent. Table 4-1 provides a selection matric generated for a comparative approach to component selection based on a weight system. The weights can be viewed below, with a 5 by 5 weight being the highest level.

- 5 = Very high
- 4 = High
- 3 = Medium
- 2 = Low
- 1 = Very low
- 0 = Not an imbedded property

Table 4-1: Selection Matrix- Voice Interface System

	Weight	Node MCU 8266	PLC	Raspberry Pi	HC-05	HC-06
GUI - User friendly	5	5*5	5*1	5*1	5*0	5*0

Cost effectiveness	5	5 *5	5*2	5*4	5*5	5*5
Weight	5	5*5	5*3	5*4	5*5	5*5
Debugging	5	5*5	5*1	5*5	5*3	5*5
Simplicity of coding	5	5*5	5*1	5*5	5*5	5*5
System integrity	5	5*5	5*5	5*5	5*3	5*5
System integration	5	5*4	5*5	5*4	5*5	5*5
Reliability	5	5*2	5*5	5*5	5*3	5*5
Current industrial use	5	5*2	5*5	5*2	5*5	5*5
Free Self-learning resources	5	5*5	5*1	5*5	5*5	5*5
System level of autonomy	5	5*5	5*3	5*5	5*0	5*0
Impact of electronic failure on system	5	5*5	5*5	5*5	5*5	5*5

System capability of autonomous growth	5	5*5	5*5	5*5	5*5	5*5
Total	**325**	**290**	**210**	**275**	**245**	**275**

The component selection matrix validated the use of electronic components required for the design of the voice interface system, however, there are a variety of additional platforms and equipment required for system functionality. These will be discussed further in the Configuration Specifications section that follows.

4.2.4 Configuration specifications

The configuration specifications for the voice interface system was sub-divided into two categories, the first of which being a preliminary environmental configuration and secondly, the electronic design configuration.

Preliminary environmental configuration

As mentioned previously in Section 3.3.4, the region for testing was selected to be the Advanced Manufacturing Laboratory at the University of KwaZulu-Natal. This Lab provided for the perfect environment for testing as its layout was built to simulate an Advanced Manufacturing environment, and its ability to provide for a control testing platform. A graphic representation of the lab can be viewed in Figure 4-2.

Figure 4-2: Advanced Manufacturing Laboratory - University of KwaZulu-Natal

Design and Testing Configuration and Considerations

It was recommended that the creation and analysis of artificial data would be extremely beneficial to the scientific community – as indicated from Section 3.2, 3.3 and from literature. Research presented major advances of voice interfaces by companies such as Google and Amazon. The interface chosen was "OK Google" due to reasons presented in Section 3.2 and 3.4. With the knowledge gained from literature, and from methods indicated in Chapter 3, this propelled the design of the system depicted in Figure 4-3. As a cloud-based IoT platform, Arduino's Blynk Application was chosen as it facilitated data communication across all communication mediums, "OK Google" was selected as the desired voice interface, and the web-based service platform selected was "IFTTT" due to its compatibility with Google's voice application. As can be seen in Figure 4-3 two distinct systems are presented. System "1" being the operational benchmark data extraction unit, with System "2" providing the optimization approach through a cloud-based IoT platform, a web-based server, interference cancelling hardware and software. System "1" provided for the generation and extraction of raw data (operational benchmark data). System "2" attempted to optimize the performance of system "1" using a cloud-based IoT platform, Web based service, interference cancelling devices and imbedded, programmable phrase interpreters. This optimization was done to allow for the implementation of voice interfaces into manufacturing environments.

The first system, depicted as "1" in Figure 4-3, constitutes of the following devices; a microcontroller (NodeMCU), Decibel range meter, Noise simulation application, Bluetooth module (HC-06), Mobile cellular device (Smart phone), and a laptop to facilitate serial communication. Components represented by blue highlight indicate common components of the system integration (The system integration in this case being the benchmark system and optimization system). A noise simulator system was designed to simulate desired noise readings in a control room environment under decibel ranges provided by Section 3.3. The maximum decibel reading for use was 80 decibels, as provided by Section 3.3 [47, 48]. It was considered to start testing the extraction device with readings staring at 30 decibels in increments of 10 decibels to truly gauge the effectiveness of googles voice-assistant under industrial conditions. 30 decibels was chosen due it representing a quiet room environment [51]. The noise metre design and noise simulator designs can be viewed in Appendix B, Figure 10-1 and Figure 10-2. System “2” in Figure 4-3 relates to the “Internet of things” approach to system design. System “2” constitutes of the following devices; a microcontroller (NodeMCU), Google’s voice assistant, a web-based service, IoT platform, Mobile cellular device, interference cancelling hardware, and interference cancelling software. Following the benchmark data obtained from system “1”. An optimization of this data was carried out. Three areas of concern relating to voice interfaces were attempted to be addressed using the cloud based IoT platform and web-based server. The Web-based server was used to address issues such as high levels of language fluency, and low levels of interpretability. The low levels of adaptability were attempted to be addressed using the cloud-based IoT platform. To address the area of low interpretability, it was considered to implement an interpretation aided filter to the software design. Additionally, the only way, found from research to address the issue of environmental sensitivity was to implement a noise cancelling device to a device that facilitates the use of a voice interface. An instruction set for the setup of the cloud-based IoT platform and Web-based server can be viewed in Appendix C. The integration of these systems is also included in the aforementioned Appendix.

From a testing perspective, the following procedures and methods were carried out. The University of KwaZulu-Natal (UKZN – Howard campus) was used as the desired testing facility for the systems designed in this dissertation. The equipment calibration facility used was the Vibration and Research Testing Centre (VRTC), at UKZN – Westville campus. The testing facility at UKZN-Howard was set to mirror that of the VRTC facility. VRTC carries out simulation tests for power utility Eskom, these tests involve the subjection of desired equipment to various interference conditions. Testing carried out was in line with the methods suggested by VRTC, and the methods mentioned in Chapter 3. For system “1” above, the designs for the decibel range metre and noise simulator can be viewed in Figure 10-1 and Figure 10-2 of Appendix B. Due to the nature of the microcontrollers selected for the dissertation, this prompted the design of compatible data extraction units. This meant however, that the designed systems needed to be calibrated to allow for accurate data acquisition. Calibration tests carried out used precision equipment such as National Instruments integrated electronic piezoelectric microphone, TIRA interference control monitor, TIRA interference generator, a CompactDaq module, and LabVIEW software. The calibration equipment can be viewed in Figure 10-4 – Figure 10-7. The calibration testing facility can be viewed in Figure 10-5. The Calibration procedure can be viewed in a structured step by step format in Appendix B - 10.4. Sample calibration data is also provided in Appendix B, Figure 10-8. The data obtained from the calibration tests were integrated into the noise simulator and decibel metre software and hardware designs for accurate data acquisition. The VRTC equipment was not directly used in the systems designed for this dissertation due to their lack of compatibility and mobility constraints. For system “1”, the noise simulator was implemented using a software application channelled to the system designed in Figure 10-2 , of which, desired interference conditions were calibrated, then created. The noise simulator purchased had the ability to create interference conditions similar to test calibration readings at VRTC [52]. Following the generation of the desired noise conditions, recorded voice commands were subjected to system “1” and the data was extracted through the Arduino data monitor. Five different command sentences were subjected to six different interference conditions (30db – 80db). The commands included the industrially provided phrases of [47, 48];

Eight-word phrase:	Spanners are required for the conveyor belt alpha,
Six-word phrase:	Please turn on the orange LED,
Four-word phrase:	Proceed to turbine engineering,
Two-word phrase:	Engage maintenance, and
One-word phrase:	Activate

The data was then captured onto an excel spreadsheet for analysis. For accurate modelling of the data extracted, a python programme was created to model and analyse the behaviour of the data extracted – Appendix B – 10.2. Further analysis was carried out to understand the distribution of the data extracted, as such, all data was subjected to a probability density function. System “2” used the same interference subjection techniques, however, an IoT platform and web-based server was introduced. The methods and steps of the implementation can be viewed in Appendix C. One directional noise cancelling hardware was introduced into the optimization system which can be viewed in Appendix B – Figure 10-11 (Comica VS08). This device was chosen due ease of compatibility with “OK Google” and with the selected microcontroller. Though this hardware was introduced, major optimization was sought to be carried out using the IoT platform and Web-based server, due to their ability to deal with low voice interpretability and adaptability constraints. Additionally, interpretability concerns were addressed through software manipulation techniques presented in Appendix D.

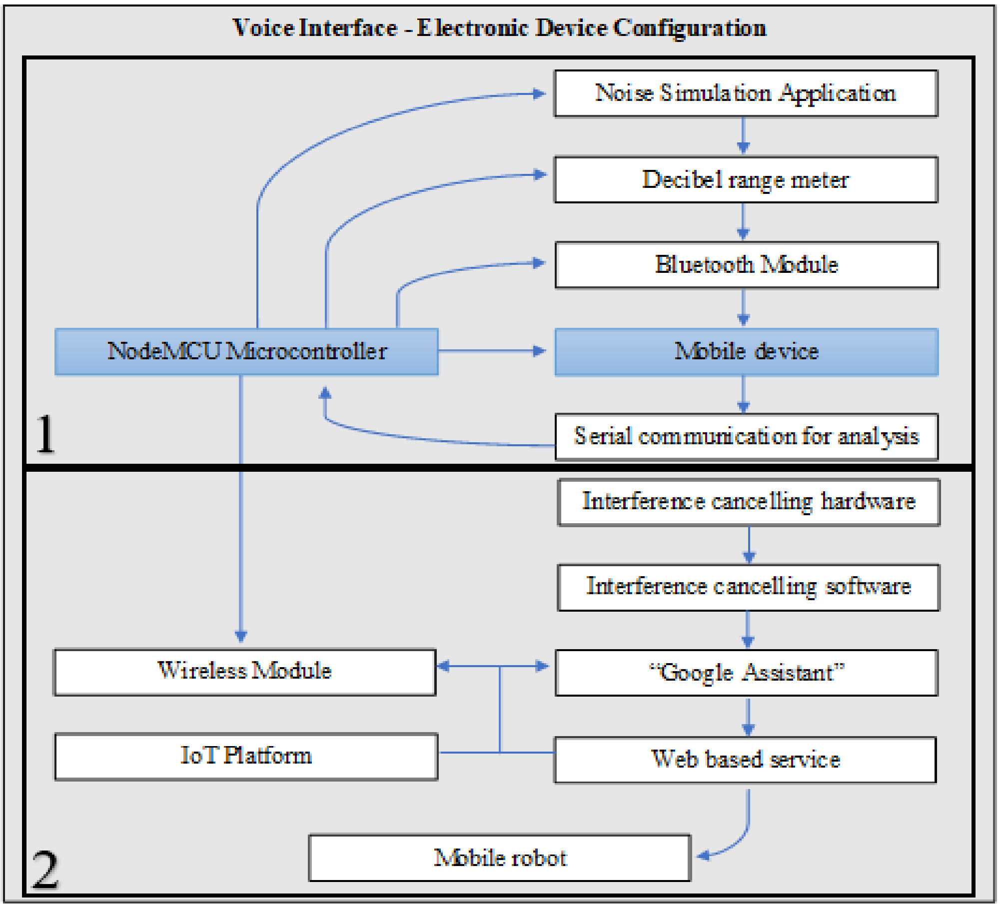


Figure 4-3: Voice Interface System- Device Configuration

4.2.5 System build overview

To gain a good understanding of the schematic representation of the system provided in Figure 4-3, a schematic representation of the system is represented in

Figure 4-4. The figure depicts the overview of the systems integrations.

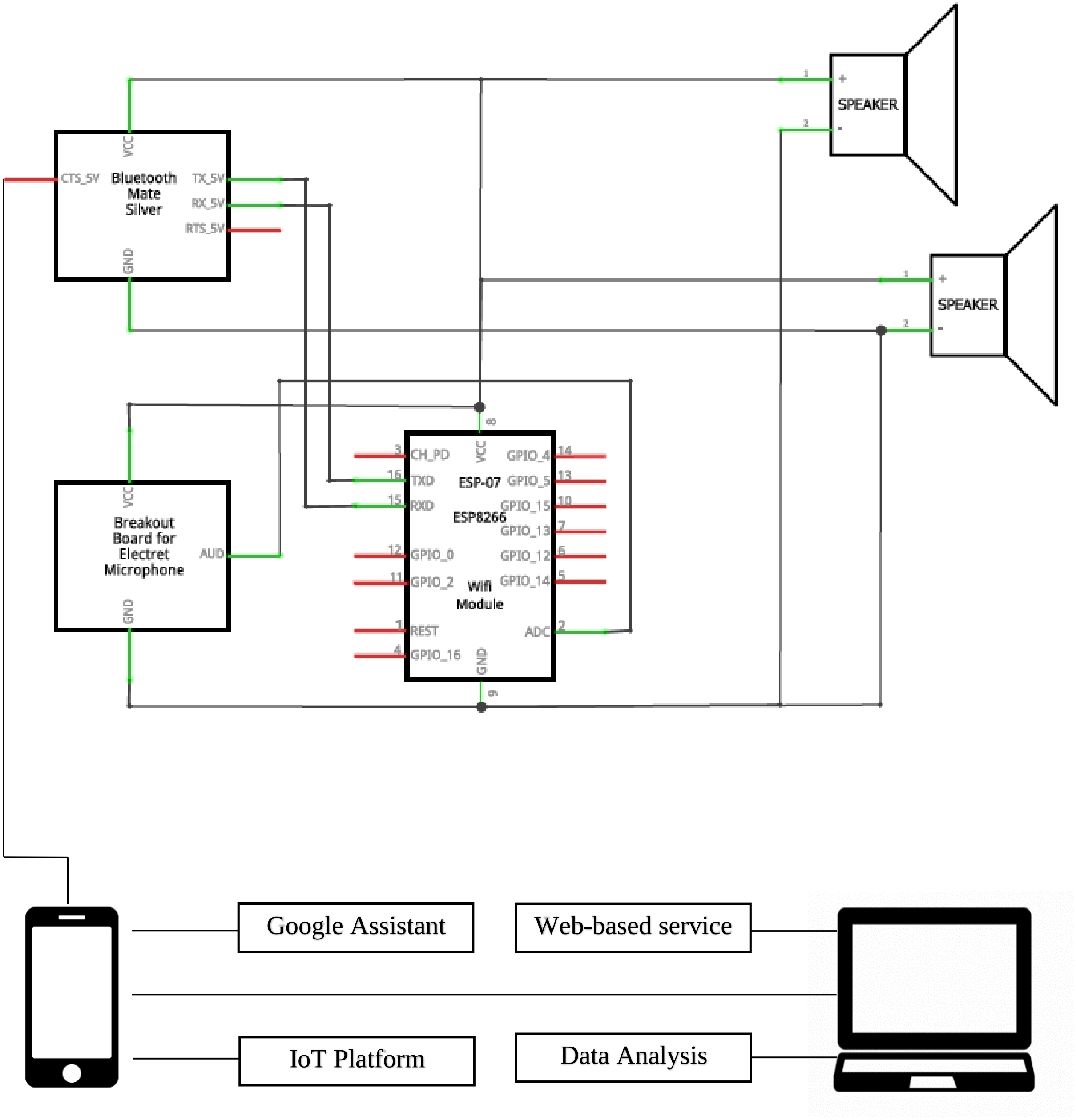


Figure 4-4: Schematic overview - voice system

4.2.6 Software design specifications

The software design section introduces the development of the Algorithm, Instruction Flow and System decomposition design of the voice interface system. These designs were developed to allow for an ease of transition into the software development phase.

System Decomposition Design- Benchmark Data Extraction System

The system decomposition diagram depicted in Figure 4-5 provides insight into the devices facilitating communication as well as the means of communication. The Benchmark Data Extraction Unit embodies six main distinctions, as can be seen below.

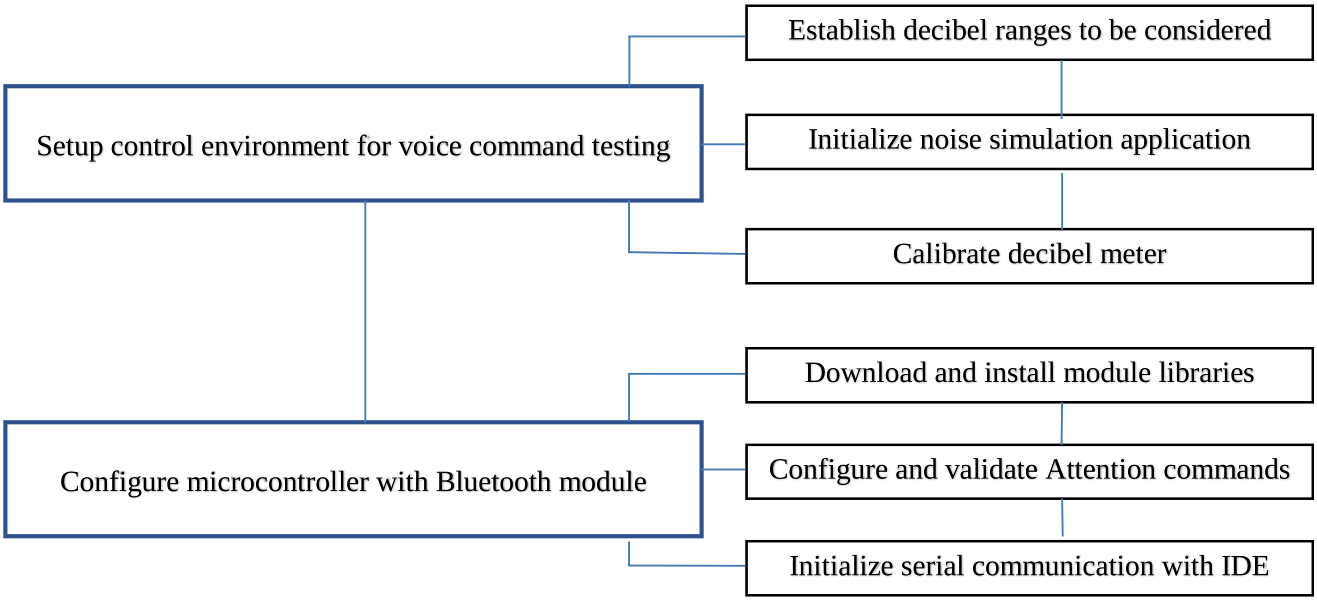


Figure 4-5: System decomposition – Benchmark Data Extraction Unit

Algorithm Design – Benchmark Data Extraction System

The list of instructions provided below provided for a means of problem solving directly relating to the Benchmark Data Extraction Unit. The six steps provided below highlight on the problem-solving software layout introduced.

Step 1: Start

Step 2: Declare variables for Bluetooth communication, set baud rates specific to components being used to facilitate communication. Declare transmission and receiving pins.

Step 3: Set transmission pins on Bluetooth module to feed data to the receiving pin of the microcontroller.

Step 4: If command is received, then transmit to Arduino IDE serial monitor. Set LED pins to HIGH if desired commands are received, and delay response to accommodate LED HIGH

Step 5: Loop Code

Step 6: End

Instruction Flow Design – Benchmark Data Extraction System

The instruction flow design of the Benchmark Data Extraction Unit can be viewed in Figure 4-6. The design represents the logic behind the instruction workflow that was used in the software design for the extraction of voice commands.

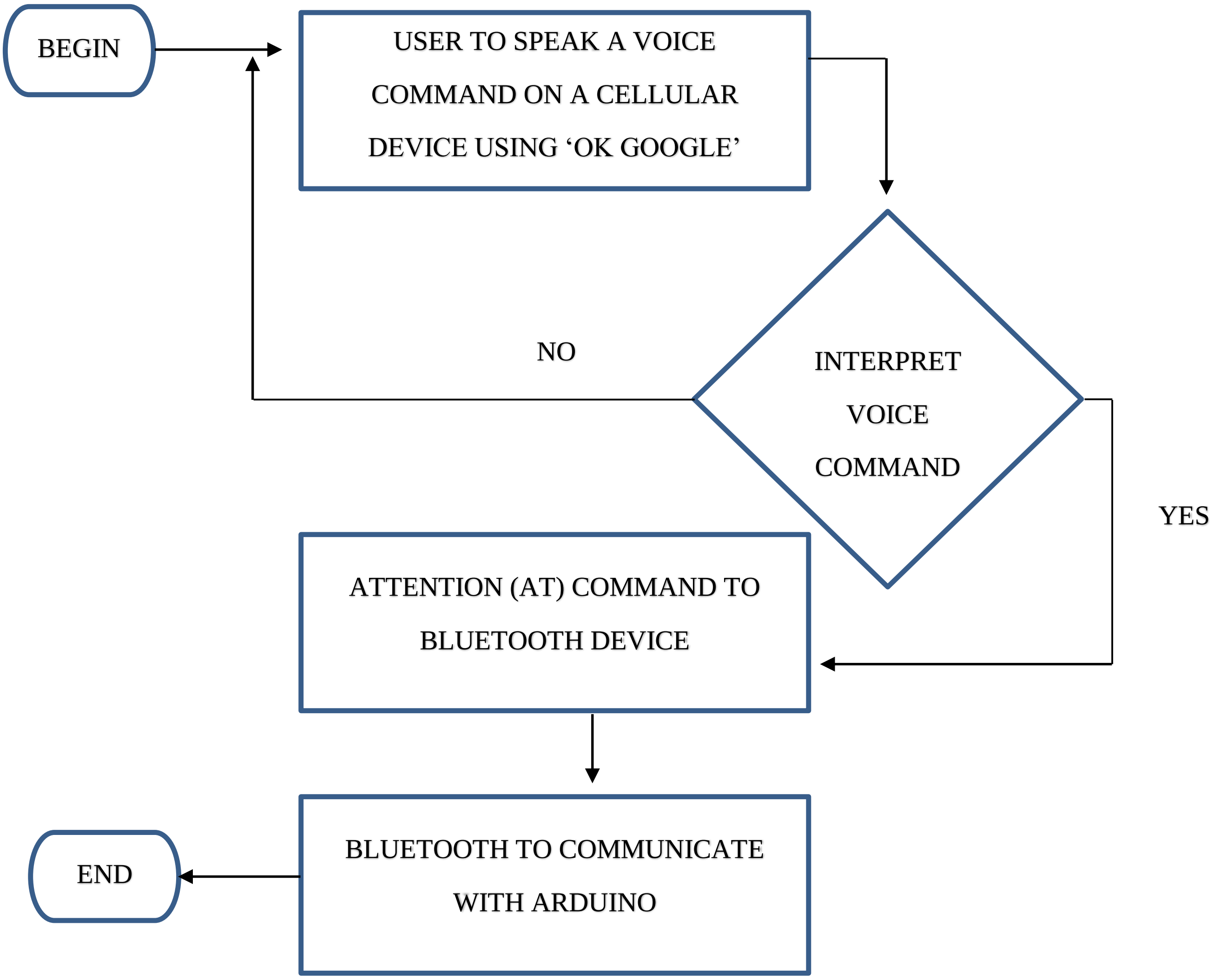
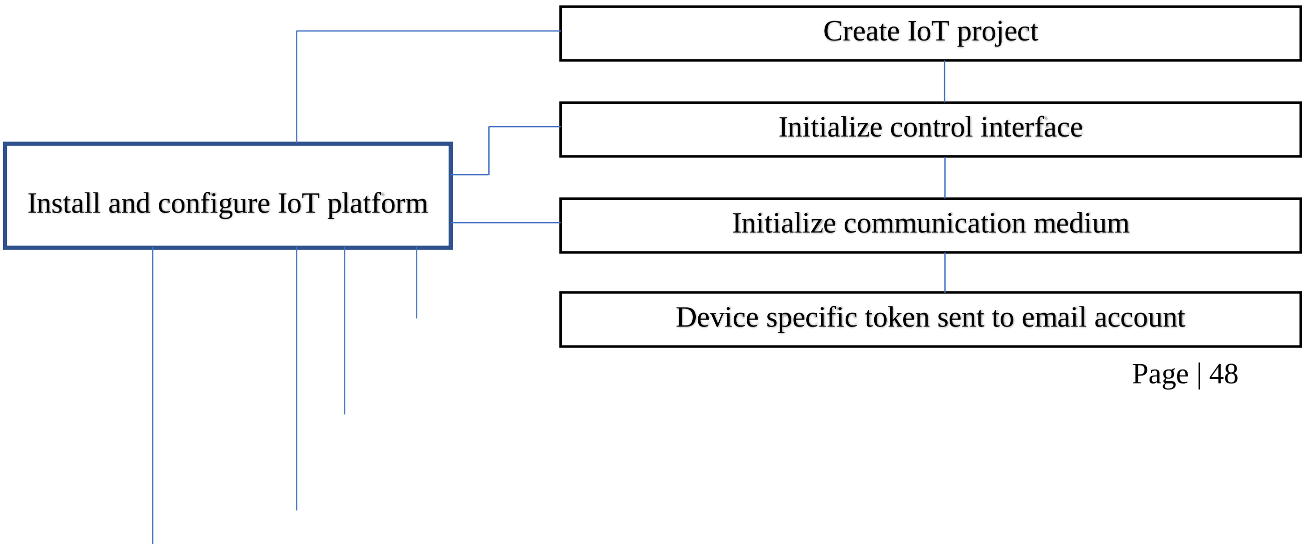


Figure 4-6: Flow design – Benchmark Data Extraction Unit

System Decomposition Design- IoT System

The system decomposition diagram depicted in Figure 4-7 provides insight into the devices facilitating communication as well as the means of communication. The IoT system embodies the distinctions as can be seen in the figure that follows.


Create IoT project
Initialize control interface
Install and configure IoT platform
Initialize communication medium
Device specific token sent to email account

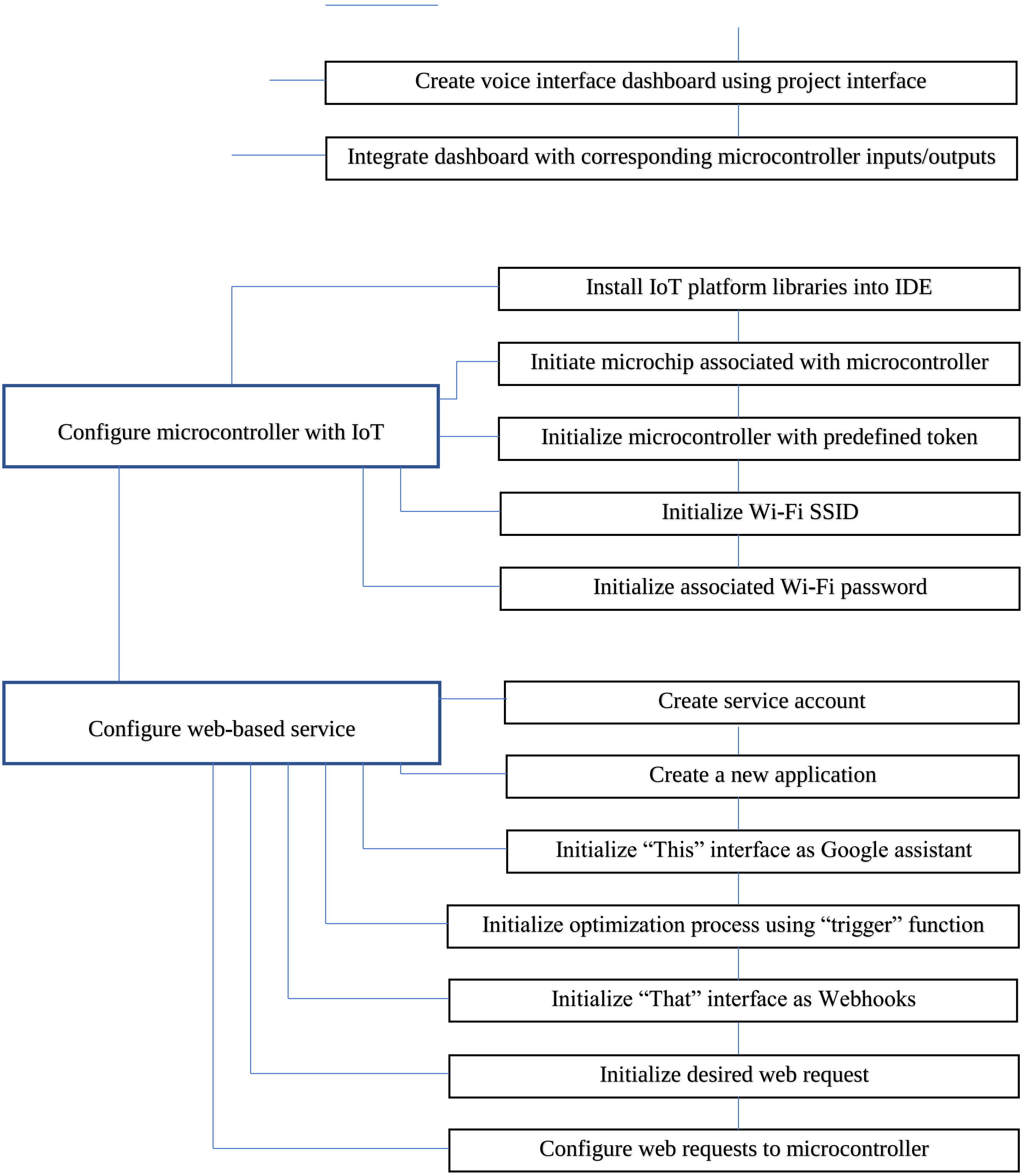


Figure 4-7: System decomposition – IoT system

Algorithm Design – IoT System

The list of instructions provided below provided for a means of problem solving directly relating to the voice interface IoT system. The six steps provided below highlight on the problem-solving software layout introduced.

Step 1: Start

Step 2: Declare variables for wireless communication and, set baud rates specific to components used to facilitate wireless communication. Additionally, declare IoT platform ID token.

Step 3: Configure digital pin to link with IoT dashboard

Step 4: If command is received, communicate command with web based server, send command to IoT platform and then to Arduino IDE

Step 5: Loop Code

Step 6: End

Instruction Flow Design – IoT System

The instruction flow design of the IoT system can be viewed in Figure 4-8. The design represents the logic behind the instruction workflow that was used in the software design for the IoT system.

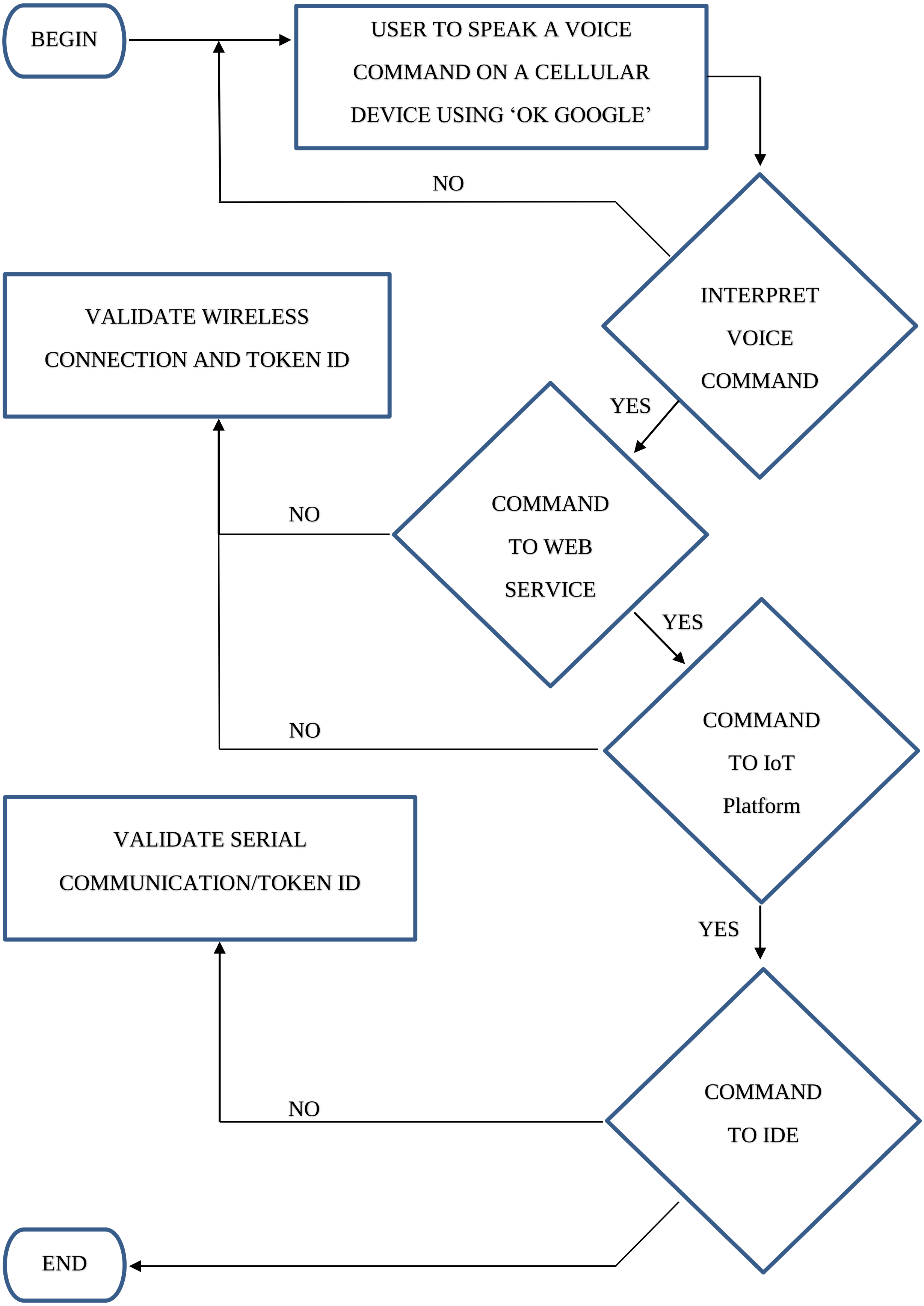


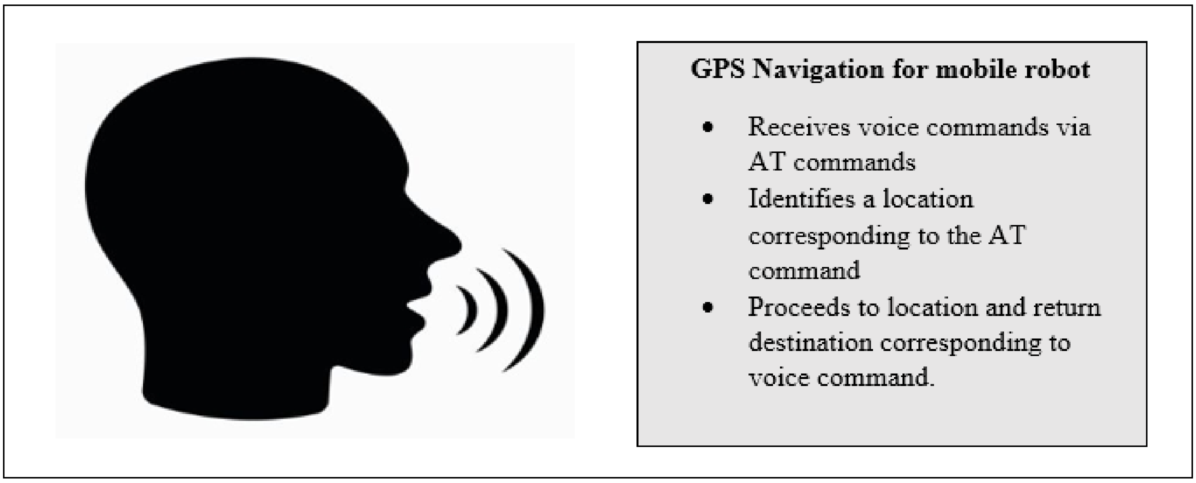
Figure 4-8: Flow design – IoT system

4.3 Navigation system

The mobile navigation robot design was carried out to meet concerns brought up in Section 3.3 and from literature. This system was designed to create a mean of facilitating the integration and implementation of voice interfaces into the manufacturing domain.

4.3.1 User requirements

The navigation system design relies on input signals received from the voice interface system. Given that appropriate commands are received, this system allows for mobility, upon receiving voice commands in AT command format (AT commands), the system was designed to navigate to geographical locations corresponding to the tools/parts required. Figure 4-9 outlines the key requirements of the mobile robot.



GPS Navigation for mobile robot

- Receives voice commands via AT commands
- Identifies a location corresponding to the AT command
- Proceeds to location and return destination corresponding to voice command.

Figure 4-9: Mobile robot system overview

4.3.2 Component functional specifications

As an electronic control device, an Arduino Mega Microcontroller was selected. The reason for which is due to its advanced system requirements and ease of integration to the ESP (8266). Additionally, for the system under consideration, 5V DC motors were selected for mobility applications. The motors were selected due to the limiting voltage output capabilities of an Arduino board. Additionally, low power consumption of low potential motors was a highly contributing factor. Other components considered included DC motor control modules, a magnetometer and a Global Positioning System (GPS) module for positioning requirements.

Motor control module: Initial considerations for a simple motor control module explored the use of two dual channel H-Bridge modules (HG7881), however, upon further research into motor control modules compatible with Arduino, an Arduino motor shield presented itself. The motor shield selected was the L293D motor module. The module allowed for the use of 4 DC motors, and had the ability to expand its operations using an additional two servo motors and two stepper motors.

Magnetometer: The magnetometer module allowed for compass navigation and direction identification. Though many magnetometers were explored, all provided for the same directional capabilities. The magnetometer selected was the LSM 303, due to its rich availability of module resources for the Arduino platform.

GPS Module: The GPS module allowed for ease of connectivity with GPS satellites in an attempt to stream from these systems, GPS coordinates. GPS modules compatible with Arduino presented themselves in a wide variety though, many of these modules did not exist in South Africa. In an attempt to build a system capable of re-development locally, local devices were used. Local GPS systems included the GPS module NEO-6M, and a later version of this module, the NEO-7M. The NEO-6M was chosen due to lower cost and higher application library content.

4.3.3 Component selection matrix

The selection matric of the aforementioned electronic components can be seen in Table 4-2.

Table 4-2: Selection matric – Mobile robot

	Weight	L293D	HG7881	NEO-6M	NEO-7M
Cost effectiveness	5	5*5	5*2	5*5	5*3
Weight	5	5*4	5*5	5*5	5*5
Debugging	5	5*5	5*3	5*5	5*5
Simplicity of coding	5	5*5	5*5	5*5	5*5
System integrity	5	5*5	5*5	5*5	5*5

System integration	5	5*5	5*5	5*5	5*5
Reliability	5	5*5	5*3	5*5	5*5
Current industrial use	5	5*2	5*2	5*2	5*2
Free Self-learning resources	5	5*5	5*5	5*5	5*5
System level of autonomy	5	5*5	5*5	5*5	5*5
Impact of electronic failure on system	5	5*5	5*5	5*5	5*5
System capability of autonomous growth	5	5*5	5*5	5*5	5*5
Total	**300**	**280**	**225**	**285**	**275**

4.3.4 Configuration specifications

The configuration specifications for the Navigation system was sub-divided into two categories, the first of which being a preliminary environmental configuration and secondly, the electronic design configuration.

Preliminary environmental configuration

As mentioned previously in Section 4.1.4, the region for testing was selected to be the Advanced Manufacturing Laboratory at the University of KwaZulu-Natal. This Lab provided for the perfect environment for testing as its layout was built to simulate an Advanced Manufacturing environment. Navigation of the mobile robot based on voice commands was carried out in this environment. Within the testing environment existed a variety of simulated workstations such as a “Fanuc station”, “Festo station”, and a “Parallel kinematic station” to name a few. Clearly defined paths to each work station existed to allow for ease of mobility of the robot between stations. Path accuracy was measured using GPS location data. A graphic representation of the lab can be viewed in Figure 4-2.

Design Configuration and Considerations

The mobile robot design configuration was built around its navigation ability. As such, for reasons discussed in Section 4.2.2, the following components were used; As the brain of the mobile system, an Arduino Mega microcontroller, a L293D motor shield for the control of four DC motors, an LSM303 (Triple-axis accelerometer and magnetometer) for direction control of the mobile robot, and a NEO-6M GPS module to allow for accurate GPS data streaming and positioning. A basic layout of the electronic design configuration can be viewed in Figure 4-10. The figure also displays the integration of the voice system with the mobile robot through the NodeMCU Microcontroller.

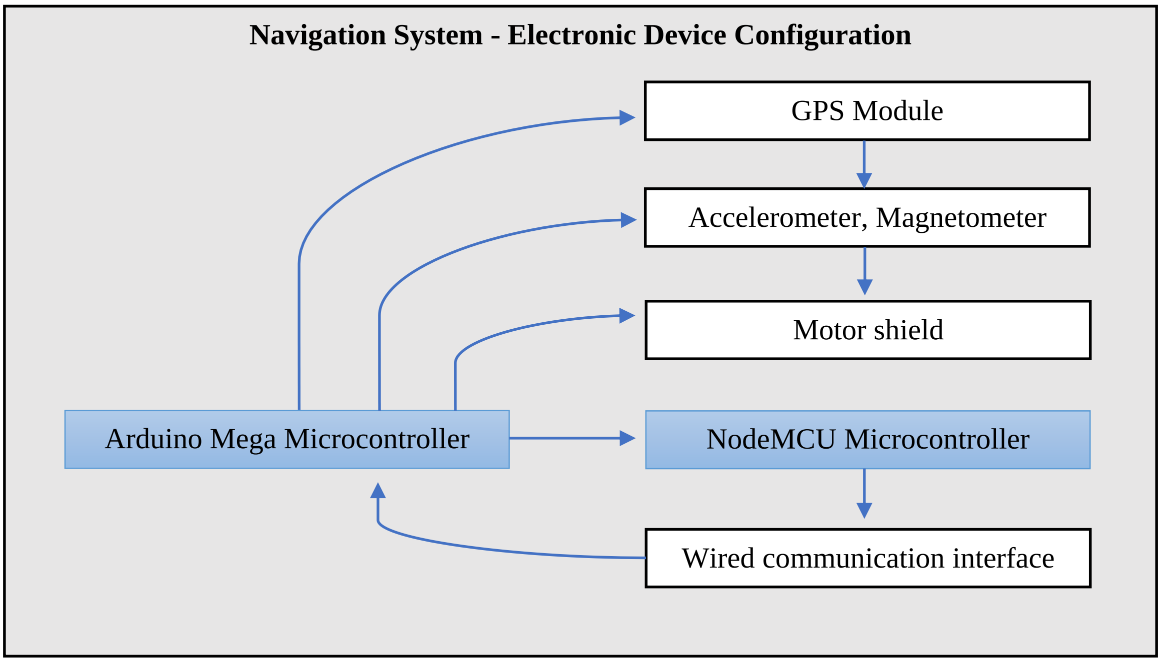


Figure 4-10: Navigation System- Device Configuration

4.3.5 System build

To gain a good understanding of the schematic representation of the system provided in Figure 4-10, a schematic representation of the system is represented in Figure 4-11 . The figure depicts the overview of the system integrations.

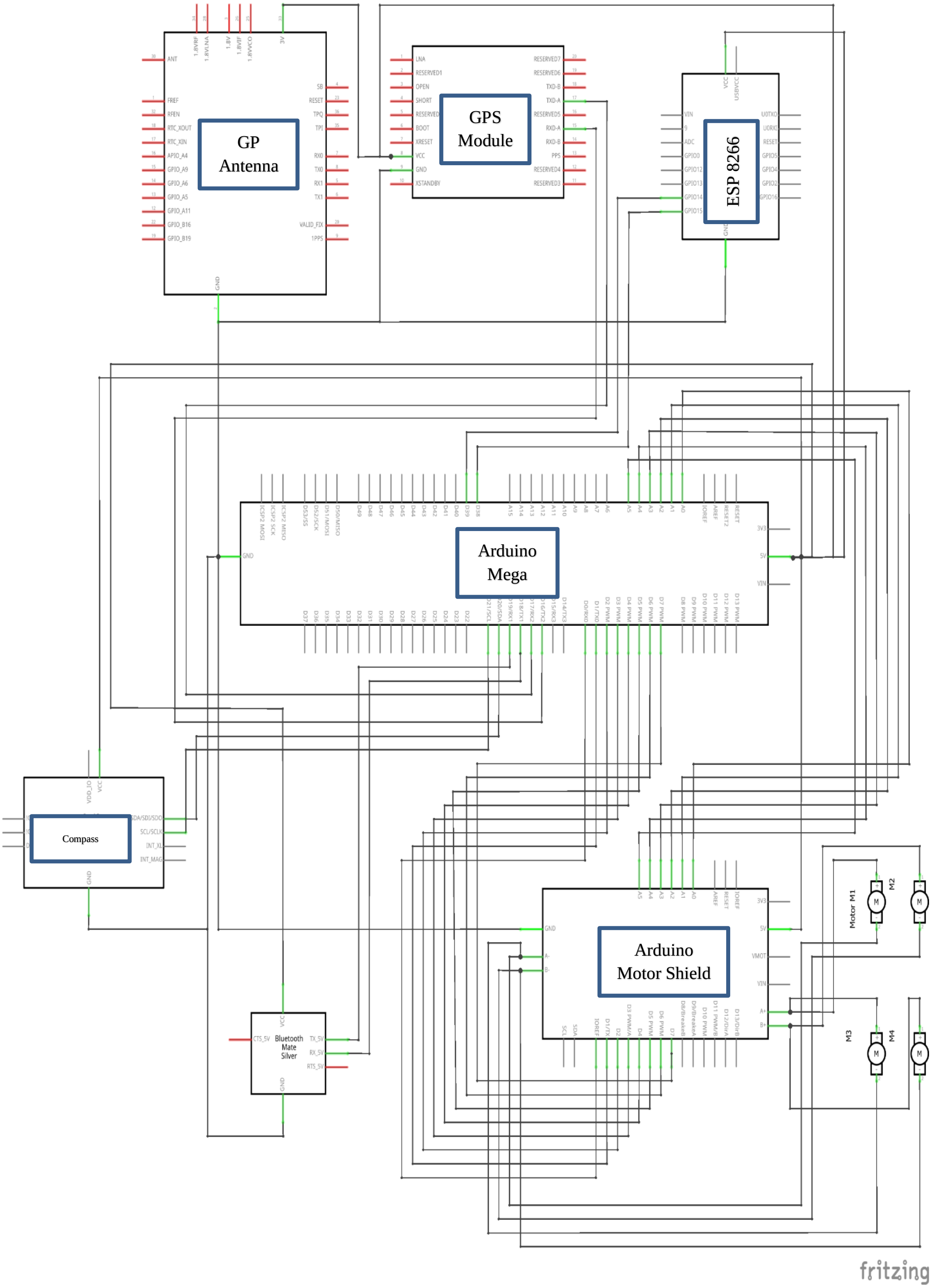


Figure 4-11: Schematic overview - Mobile robot

4.4 Software design specifications

The software used in the dissertation was specific to the Arduino IDE platform. All software used was integrated on this platform and a main class was created to facilitate the integration of sub-classes. The software design used consider the methods mentioned in Chapter 3. The sub-classes controlled by the main class included; a Start-up class, Bluetooth class, GPS and compass class, waypoint class, and steering class. The software design can be viewed in Appendix D.

4.5 Final assembly considerations

The final designs and configurations of the systems mentioned in this chapter can be viewed in Appendix B: 10.5. The interference configuration can be viewed in Figure 10-9. The microphone on the device facilitating “OK Google” was the LG EAB62950801 microphone module, this device was sensitive to a maximum decibel range of 80 dB which was the maximum interference level used for testing in this dissertation. Naturally, operational benchmark values in some instances, would be higher for a more advanced microphone. The device and microphone module can be viewed in Figure 10-10. The device in Figure 10-10 was subjected to the configuration in Figure 10-9. The microphone hardware used for optimization can be viewed Figure 10-11. This device provided for one directional voice interpretation with interference cancelling outside the one directional channel. The mobile robot motor design layout can be viewed in Figure 10-12. The navigation system integration with the motor design can be viewed in Figure 10-13 and Figure 10-14. Figure 10-15 displays the integration of the of the navigation system with the IoT and web-based platform (Through the NodeMCU). Figure 10-16 displays the location of the tool handling pouch and compass integration on the mobile robot. The compass was linked to the navigation system as depicted in Figure 4-11. The full final assembly of the voice controlled mobile navigation system can be viewed in Figure 10-17. Control of the device, as mentioned earlier, could be initiated using the commands mentioned in Section 4.1.4 on any mobile device facilitating “OK Google”.

4.6 Chapter summary

Chapter four displayed the Electronic Design Chapter. The chapter gives details of the design of an integrated voice controlled mobile navigation system, which involves the design of three different systems. The first system related directly to the generation of operation benchmark data when subject to manufacturing conditions. System two was designed to optimize on accuracy and reliability conditions presented by the benchmark data. Lastly, a mobile navigation robot was designed to create a platform for implementation of voice interfaces into the manufacturing domain. Detailed and technical information about these designs were given in this chapter.

5. DATA ANALYSIS

5.1 Introduction

The chapter that follows presents the operational benchmark data extracted from voice-assistant "OK Google" when subjected to industrial noise interference conditions. The range of noise conditions subjected were based on information provided by Section 3.3. To gain a well-established distribution of the data obtained, phrases of varying lengths were subjected to the conditions mentioned in preceding chapters. The testing data obtained can be seen below. The data that follows includes the data of the optimization design as compared to the benchmark data.

5.2 Voice interface - System analysis

The following section provides the data extracted and analysed for the systems designed in Chapter 4.

5.2.1 Voice interface data: Eight-word phrase

The first set of data obtained can be viewed from Figure 5-1 – Figure 5-6, with a results summary being depicted in Figure 5-7. Figure 5-1 – Figure 5-6 represent the data extracted from the subjection of googles voice assistant to conditions highlighted on in Chapter 3 and Chapter 4, this data includes the operational benchmark data, as well as the optimization data, shown as the blue and orange curves respectively (Bench and OPT). As can be seen from Figure 5-1 – Figure 5-6, the data was represented on a probability density function to establish a concise overview of the data, with the ability to additionally identify data inconsistencies. The phrase accuracy raw data extraction curves can be viewed in Appendix E, each of which corresponding to their designated probability density function curves. The phrase accuracy corresponding to Figure 5-1 – Figure 5-6 can be viewed in Figure 13-1 – Figure 13-6. Figure 13-1 – Figure 13-6 displays the operational benchmark data of the eight-word phrase run over 100 constant recorded iterations – to googles voice assistant, to gain a good understanding of the data distribution. Figure 5-7 represents "reliability" and average "phrase accuracy increases" of the eight-word phrase data. The results obtained for the optimization system displayed phrase accuracy increases throughout every instance of the data, with the highest increase being 378% of the original benchmark and a maximum reliability increase of up to 100%. Though not displayed in Figure 5-1 – Figure 5-6, and purely due to the nature of clustered data, the result deviations were not displayed; though key notes will be given in this chapter and further accentuated on in the discussion. As can be seen from Figure 5-1 and Figure 5-2, a clear data concentration shift is noted for the "OPT" curve. Quartiles between one standard deviation of the mean increased by 5.75 % and 3.92 % on their lower and upper limits respectively – for Figure 5-1 , with the average phrase recognition rate in this instance remaining constant at 85.38 %. Figure 5-2 displayed increases of 28.99 %

and 13.37 % in the “OPT” curve for the first standard deviation, with the average phrase recognition rate increasing by 0.65 %. As can be seen from Figure 5-3 – Figure 5-6, a clear, positive horizonal shift of the probability density function curve is noted. This indicated the increase of the operational benchmark values across the distribution. Figure 5-3 showed an increase of 151.04 % and 22.82 % on the limits of the first quartile, with an average operational benchmark increase of 52.21 %. Figure 5-5 showed an increase of 11.23 % and 119.61 % on the limits of the first quartile, with an average operational benchmark increase of 378 %. Figure 5-4 showed an increase of 88.62 % and 100.96 % on the limits of the first quartile, with an average operational benchmark increase of 94.54 %, Lastly Figure 5-6 showed an increase of 3.78 % and 28.78 % on the limits of the first quartile, with an average operational benchmark increase of 12.5 %.

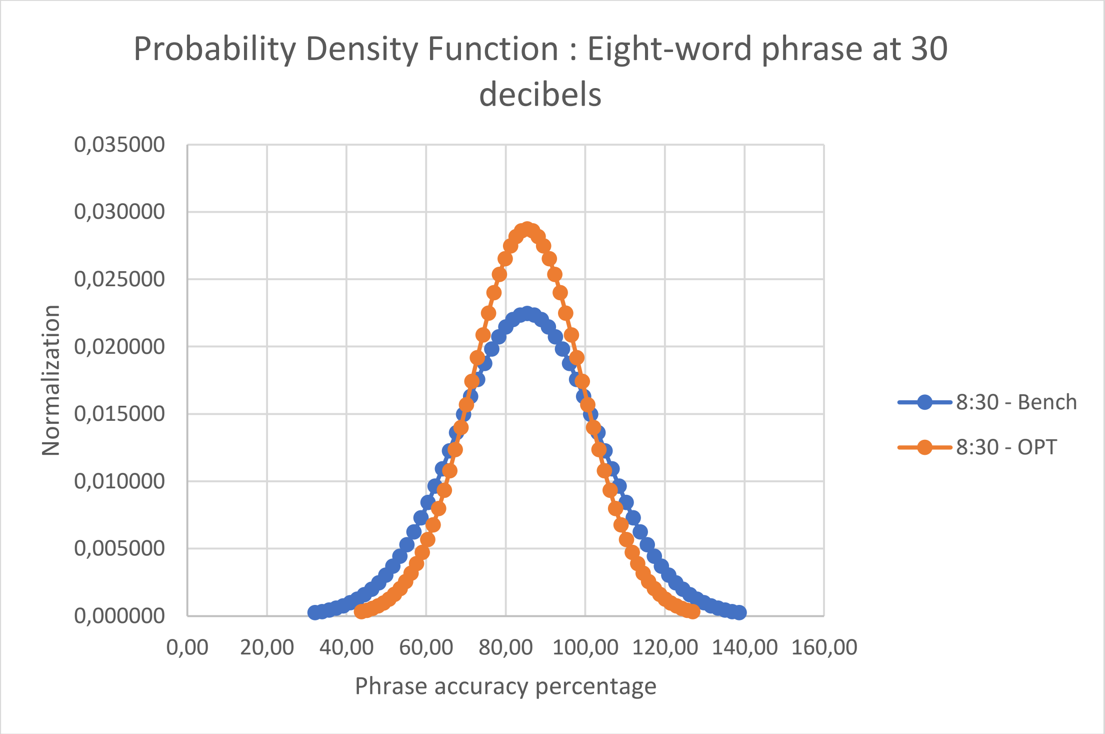


Figure 5-1: Probability Density Function: Eight-word phrase at 30 decibels

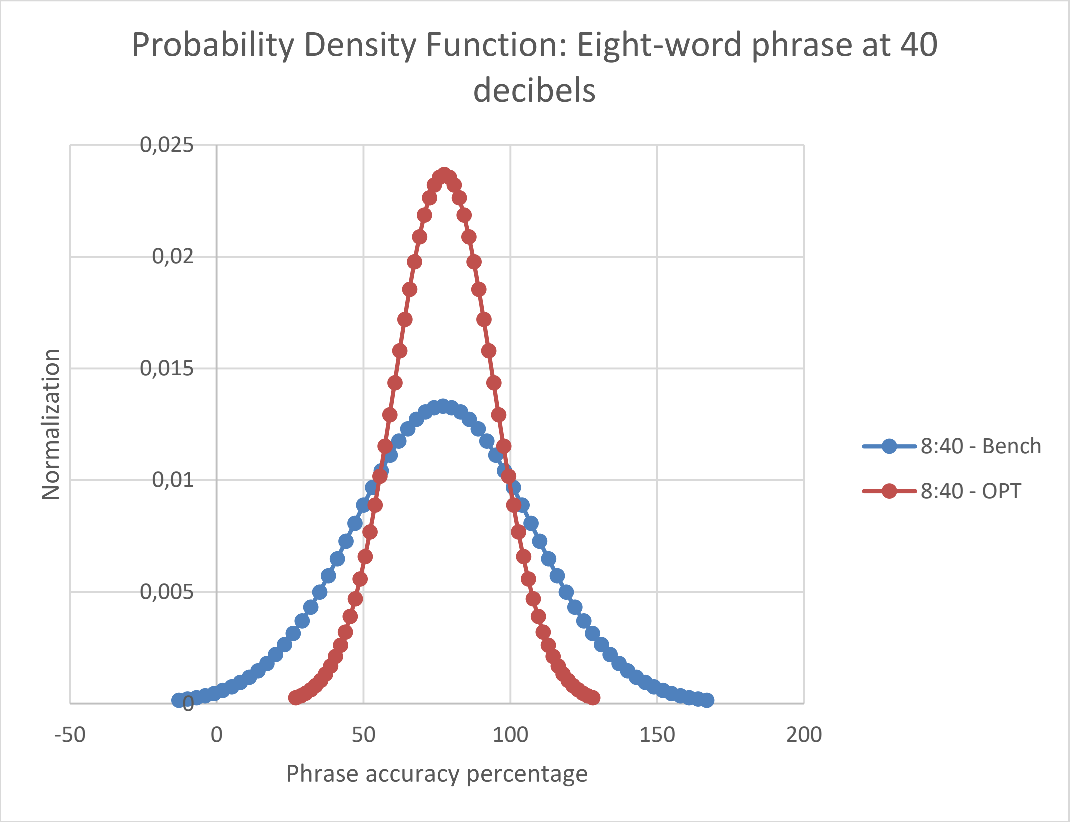


Figure 5-2: Probability Density Function: Eight-word phrase at 40 decibels

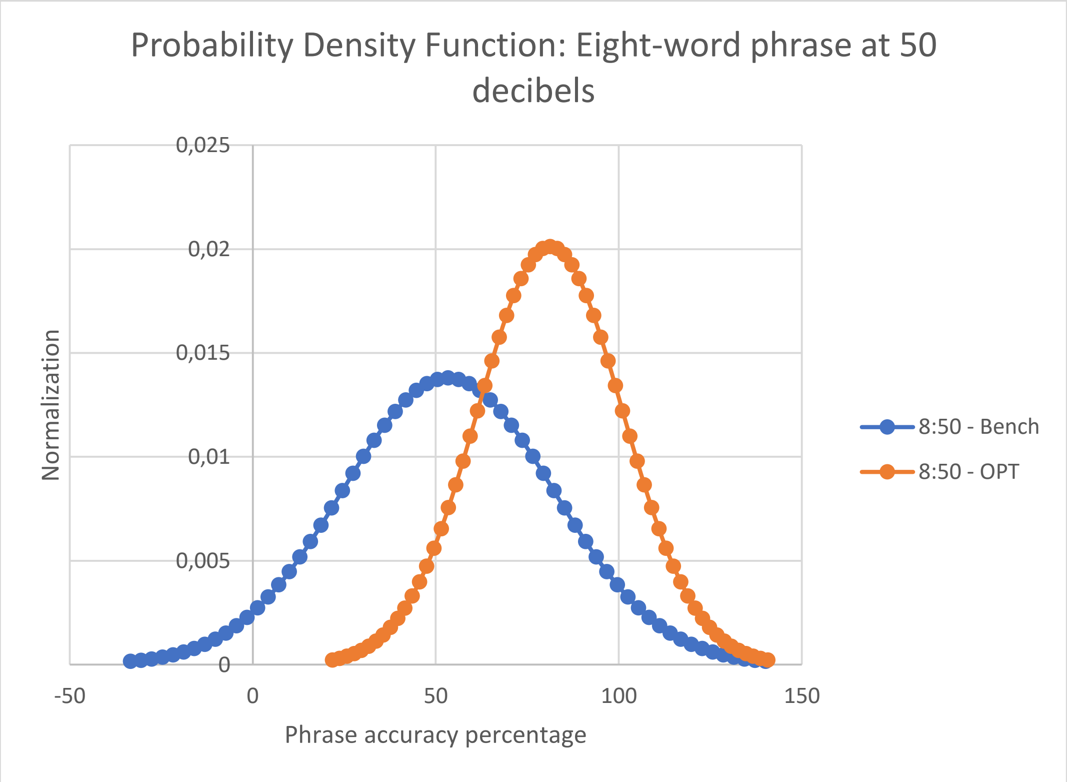


Figure 5-3: Probability Density Function: Eight-word phrase at 50 decibels

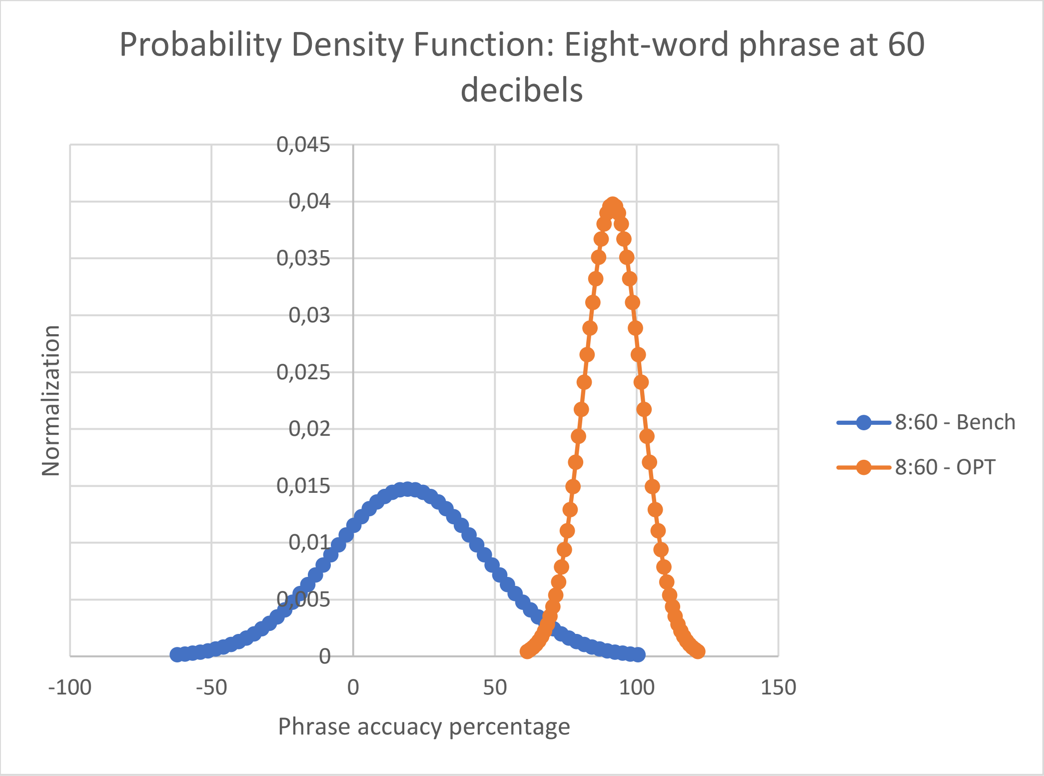


Figure 5-5: Probability Density Function: Eight-word phrase at 60 decibels

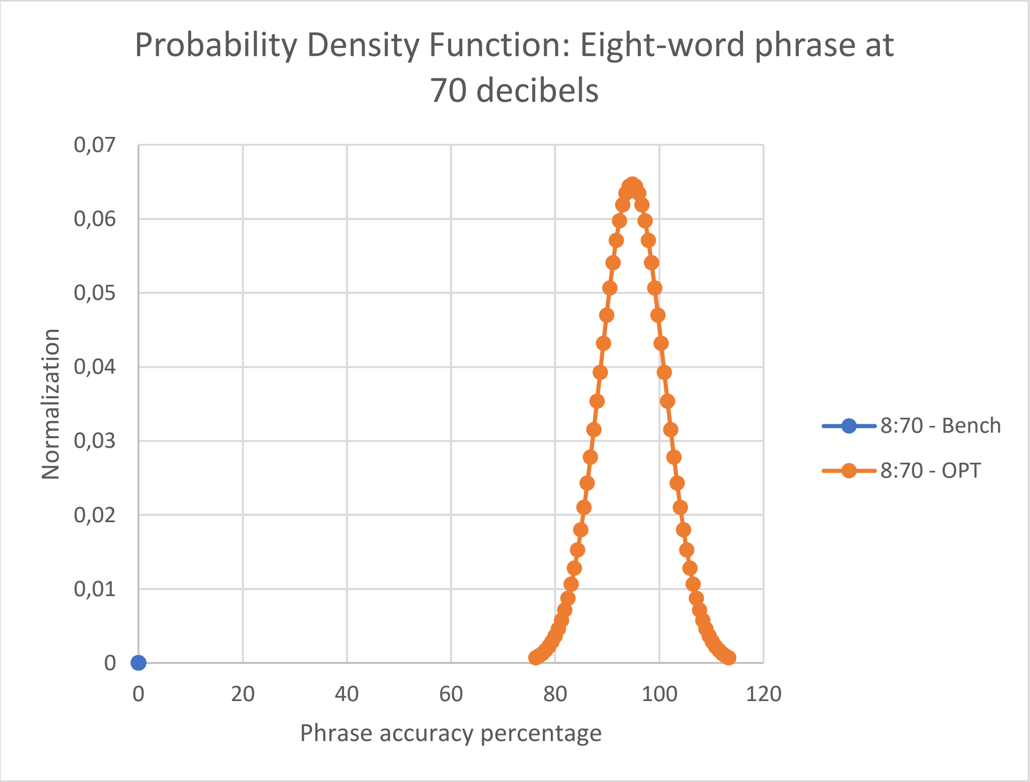


Figure 5-4: Probability Density Function: Eight-word phrase at 70 decibels

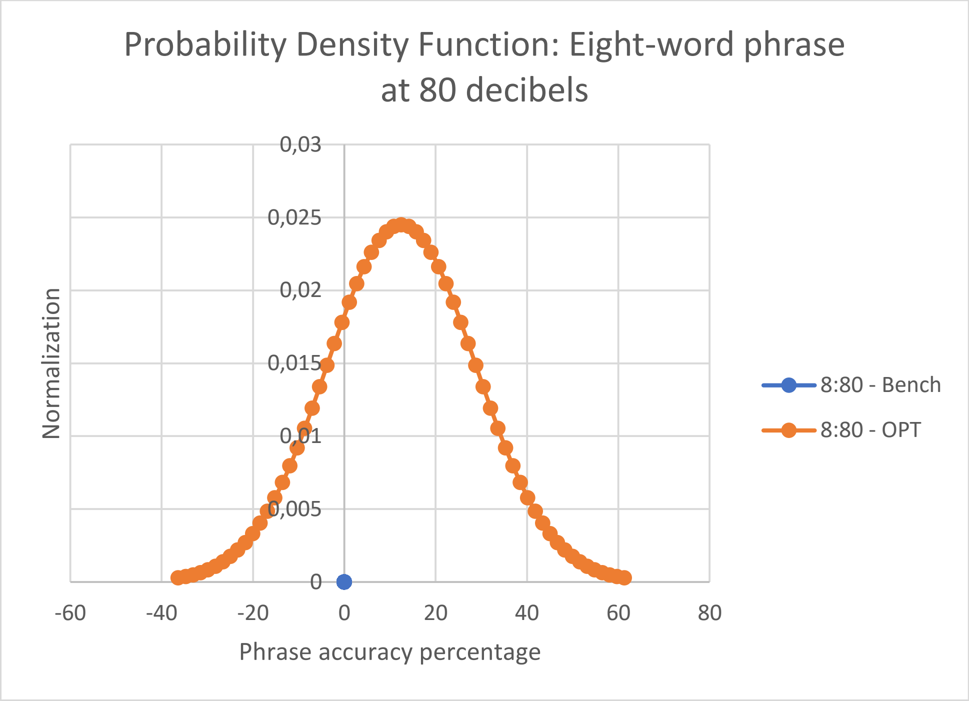


Figure 5-6: Probability Density Function: Eight-word phrase at 80 decibels

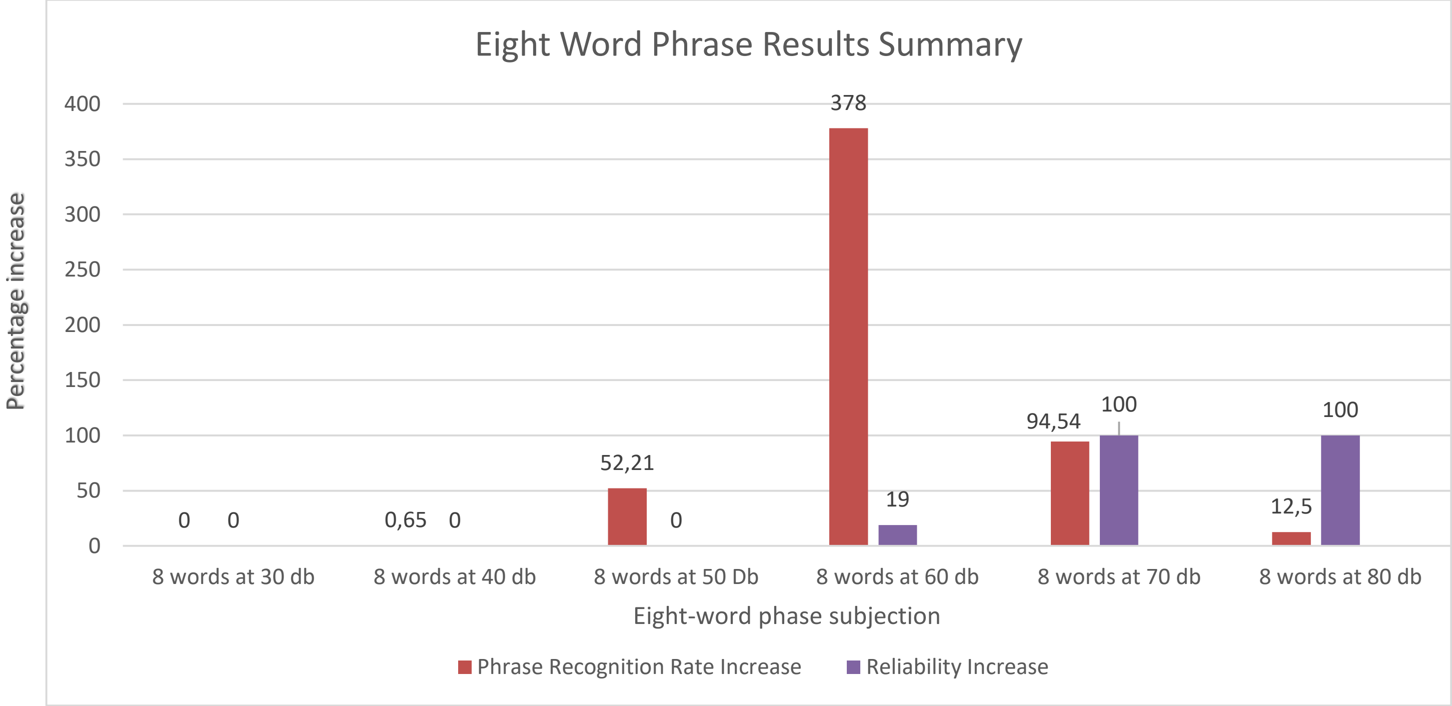


Figure 5-7: Eight-Word Phrase Results Summary

5.2.2 Voice interface data: Six-word phrase

The data analysis for the six-word phrase subject to conditions mentioned in chapter three and chapter four can be seen in the section that follows. Figure 5-8 – Figure 5-13 represent the probability density curves for the data extracted from the voice systems designed previously. The corresponding phrase recognition accuracy corresponding to Figure 5-8 – Figure 5-13 can be viewed in Appendix E in Figure 13-7 – Figure 13-12. Figure 5-14 represents the summary of the “reliability” and average “phrase accuracy increases” for the six-word phrase when compared to the operational benchmark data. Note that Figure 5-14 represents the increases only, if any, of the “reliability” and average “phrase accuracy increases” for the six-word phrase. As can be seen from Figure 5-8 two curves are presented. The probability density function curve displays the spread of the operational benchmark data, and due to the nature of the optimization results, the consistency did not allow for the generation of a probability density function. The probability density function displayed in Figure 5-8 showed a 63.07 % - 108.27% range of the data surrounding the mean at one standard deviation. The optimization results however proved to be very favourable with a consistent 100 % recognition rate. The average operational benchmark increase was shown to be 16.73 %. The probability density function displayed in Figure 5-9 showed a 49.89 % - 104.45 % range of the data surrounding the mean at one standard deviation. The optimization results however proved to be very favourable with a consistent 100 % recognition rate. The average operational benchmark increase was shown to be 29.58 %.

The probability density function displayed in Figure 5-10 showed a 24.79 % - 83.19 % range of the data surrounding the mean at one standard deviation. The optimization results however, proved to be very favourable with a consistent 100 % recognition rate. The average operational benchmark increase was shown to be 85.22 %. The probability density function displayed in Figure 5-11 showed a 5.36 % - 62.78 % range of the data surrounding the mean at one standard deviation. The optimization results however proved to be very favourable with a consistent 100 % recognition rate. The average operational benchmark increase was shown to be 193.51 %. Figure 5-12 showed the inability of the original voice interface to extract data when subjected to the corresponding conditions in Figure 5-12. The “OPT” curve of Figure 5-12 showed a 92.93 % - 103.41 % range of the data surrounding the mean at one standard deviation. The average “OPT” increase was shown to be 98.17 %. Figure 5-13, showed similar conditions to Figure 5-12, with the inability of the original voice interface to extract data when subjected to the corresponding conditions in Figure 5-13. The “OPT” curve of Figure 5-13 showed a 66.51 % - 115.55 % range of the data surrounding the mean at one standard deviation. The average “OPT” increase was shown to be 91.18 %.

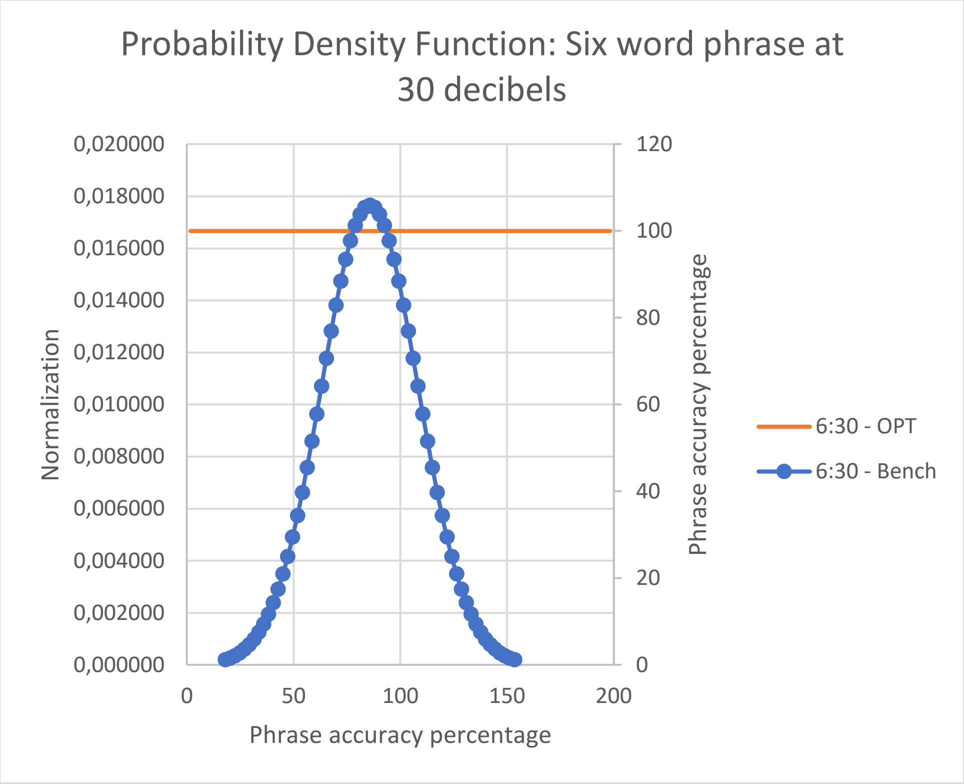


Figure 5-8: Probability Density Function: Six-word phrase at 30 decibels

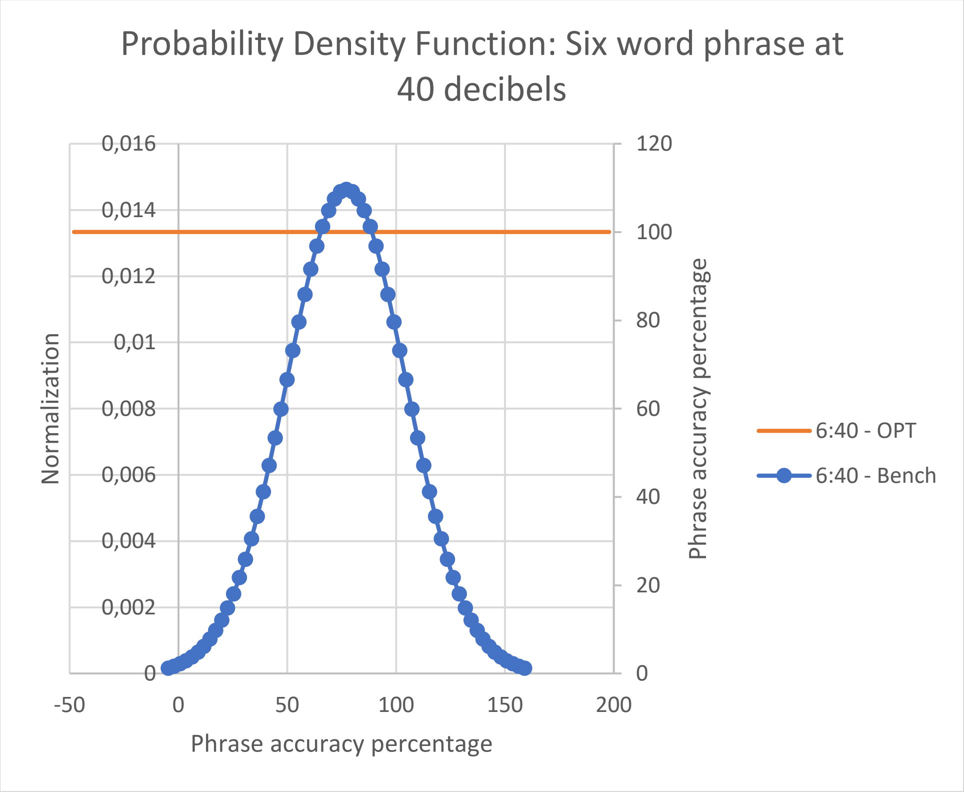


Figure 5-9: Probability Density Function: Six-word phrase at 40 decibels

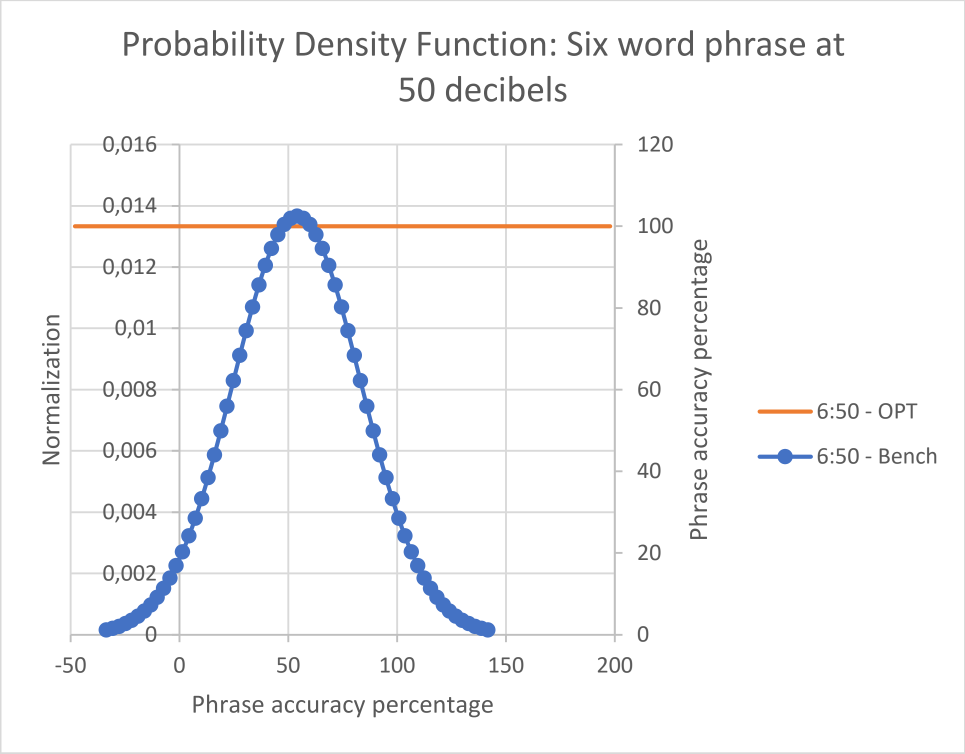


Figure 5-10: Probability Density Function: Six-word phrase at 50 decibels

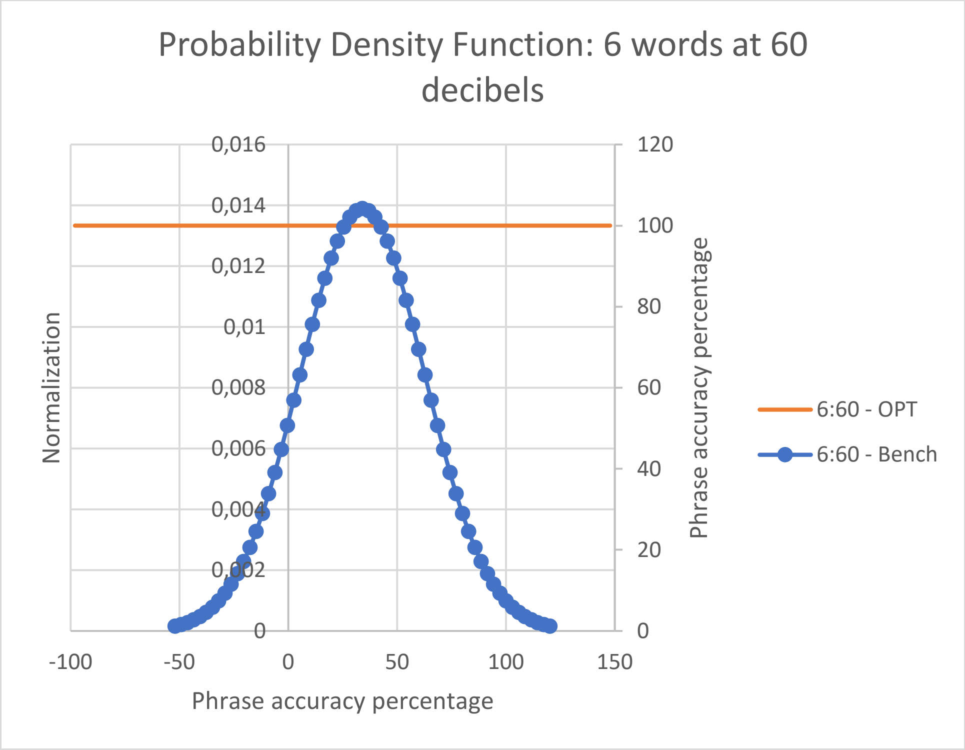


Figure 5-11: Probability Density Function: Six-word phrase at 60 decibels

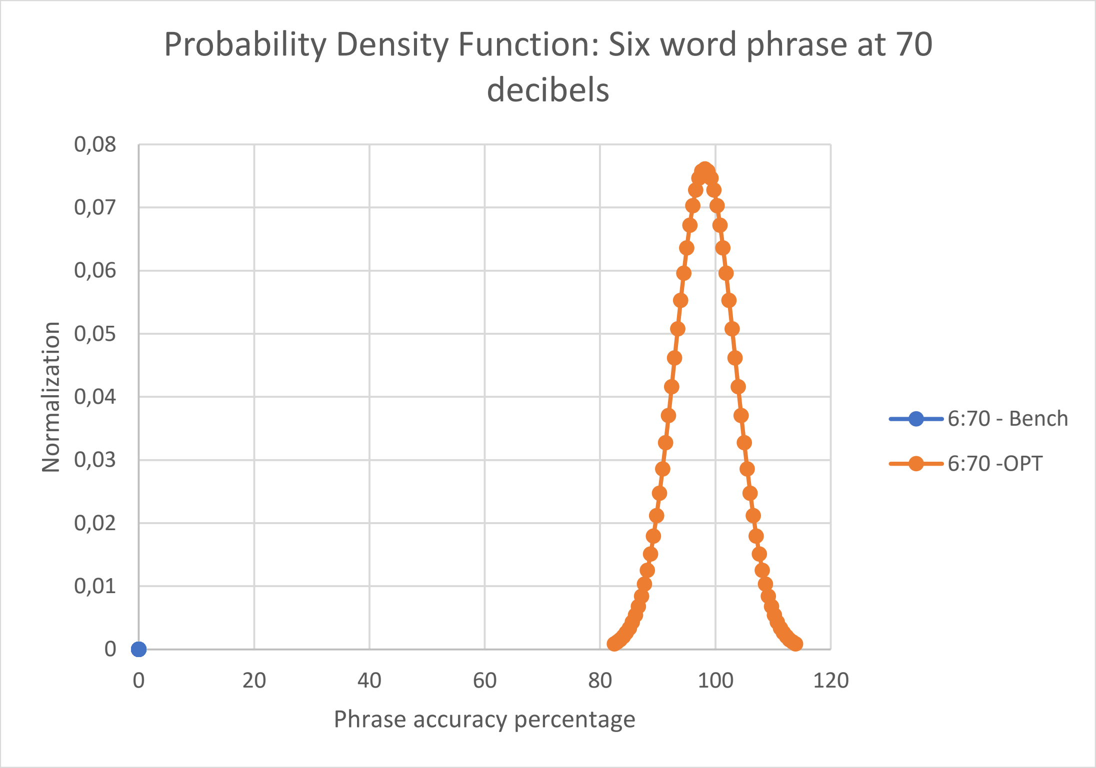


Figure 5-12: Probability Density Function: Six-word phrase at 70 decibels

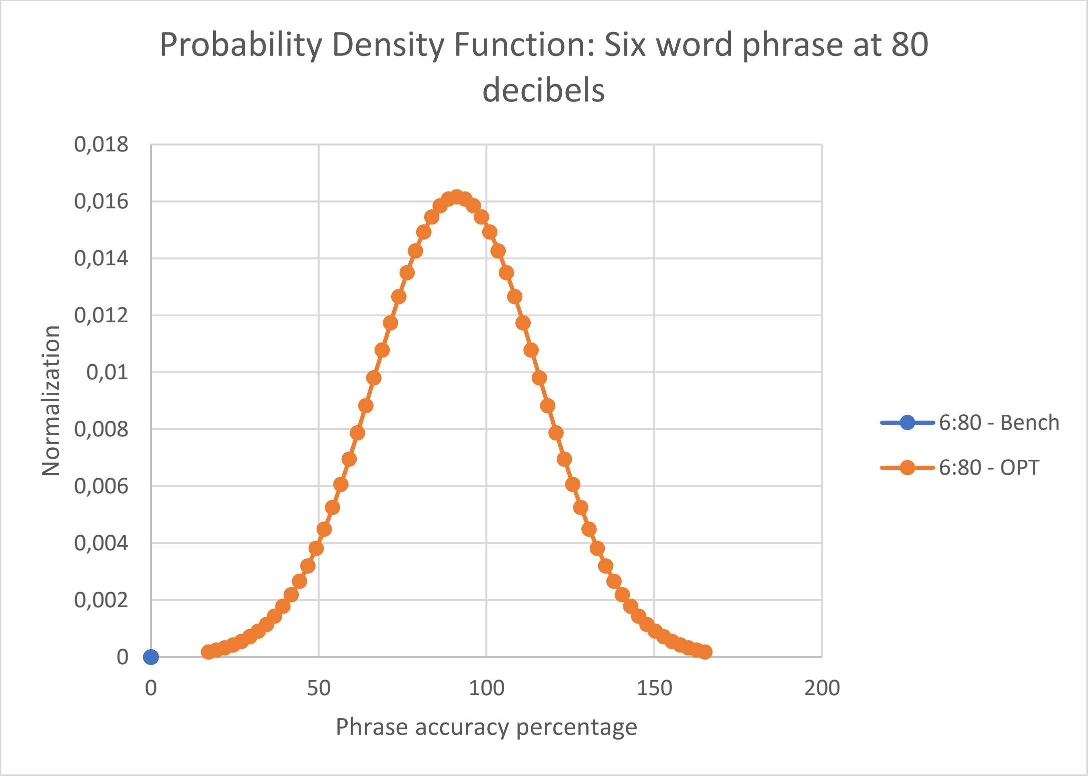


Figure 5-13: Probability Density Function: Six-word phrase at 80 decibels

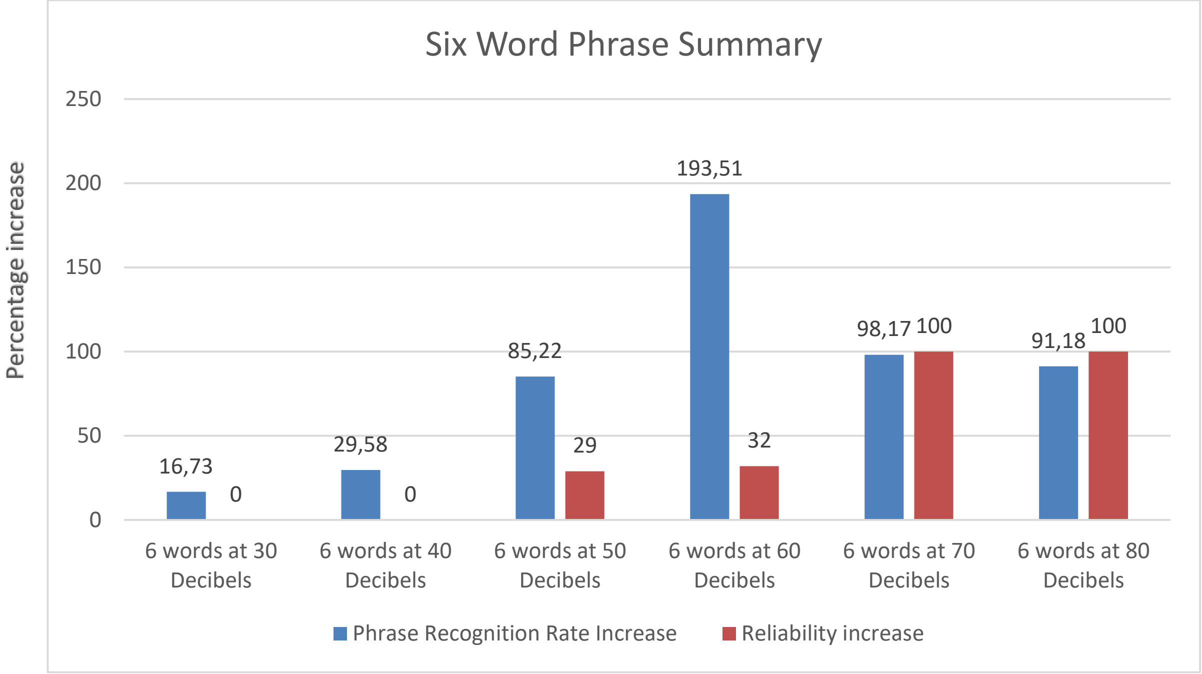


Figure 5-14: Six Word Phrase Summary

5.2.3 Voice interface data: Four-word sentence

The data analysis for the four-word phrase subject to conditions mentioned in chapter three and chapter four can be seen in the section that follows.

Figure 5-15 – Figure 5-20 represent the probability density curves for the data extracted from the voice systems designed previously. The corresponding phrase recognition accuracy corresponding to

Figure 5-15 – Figure 5-20 can be viewed in Appendix E in Figure 13-13 – Figure 13-18. Figure 5-21 represents the summary of the “reliability” and average “phrase accuracy increases” for the four-word phrase when compared to the operational benchmark data. Note that Figure 5-21 represents the increases only, if any, of the “reliability” and average “phrase accuracy increases” for the four-word phrase. As can be seen from

Figure 5-15 two curves are presented. The probability density function curve displays the spreads of the operational benchmark data, and due to the nature of the optimization results, the consistency did not allow for the generation of a probability density function. The probability density function displayed in

Figure 5-15 showed a 62.93 % - 108.57 % range of the data surrounding the mean at one standard deviation. The optimization results however proved to be very favourable with a consistent 100 % recognition rate. The average operational benchmark increase was shown to be 17.65 %. The probability density function displayed in Figure 5-16 showed a 58.22 % - 110.28 % range of the data surrounding the mean at one standard deviation. The optimization results however proved to be very favourable with a consistent 100 % recognition rate. The average operational benchmark increase was shown to be 18.69 %. The probability density function displayed in Figure 5-17 showed a 50.65 % - 72.85 % range of the data surrounding the mean at one standard deviation. The optimization results however proved to be very favourable with a consistent 100 % recognition rate. The average operational benchmark increase was shown to be 61.94 %. The probability density function displayed in Figure 5-18 showed a -9.72 % - 23.36 % range of the data surrounding the mean at one standard deviation. The optimization results however proved to be very favourable with a consistent 100 % recognition rate. The average operational benchmark increase was shown to be 1366.71 %. Figure 5-19 showed the inability of the original voice interface to extract data when subjected to the corresponding conditions in Figure 5-19. The “OPT” curve of Figure 5-19 showed a 72.66 % - 110.68 % range of the data surrounding the mean at one standard deviation. The average “OPT” increase was shown to be 91.67 %. Figure 5-20, showed similar conditions to Figure 5-19, with the inability of the original voice interface to extract data when subjected to the corresponding conditions in Figure 5-20. The “OPT” curve of Figure 5-20 showed a 12.97 % - 81.47 % range of the data surrounding the mean at one standard deviation. The average “OPT” increase was shown to be 47.22 %.

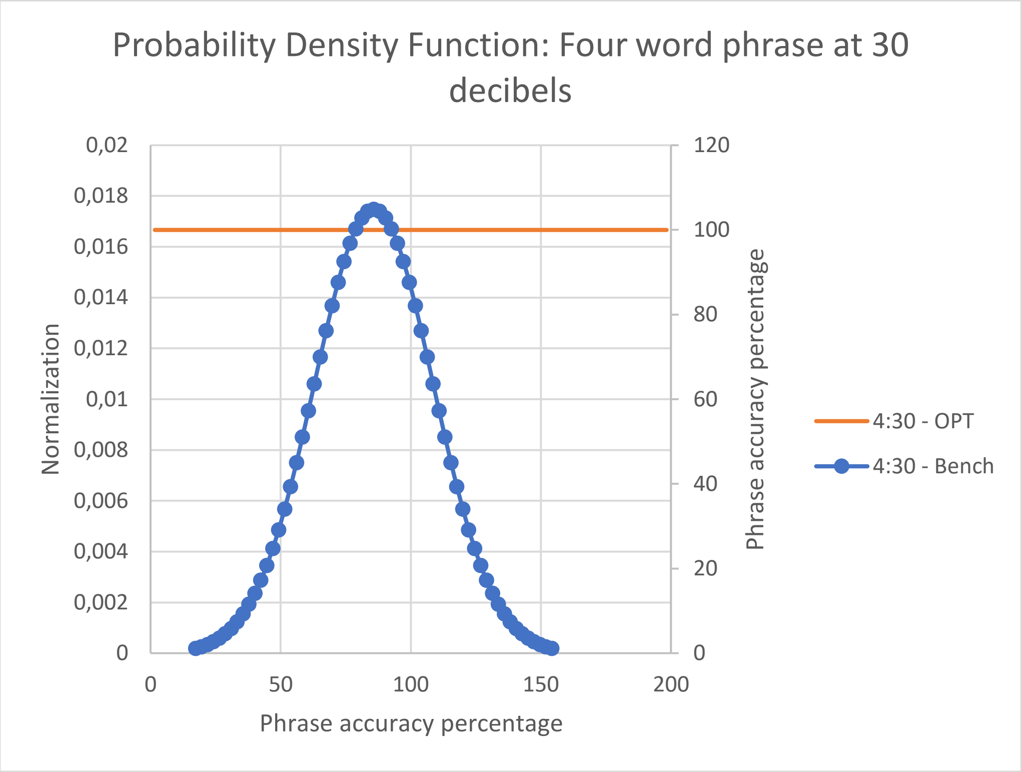


Figure 5-15: Probability Density Function: Four-word phrase at 30 decibels

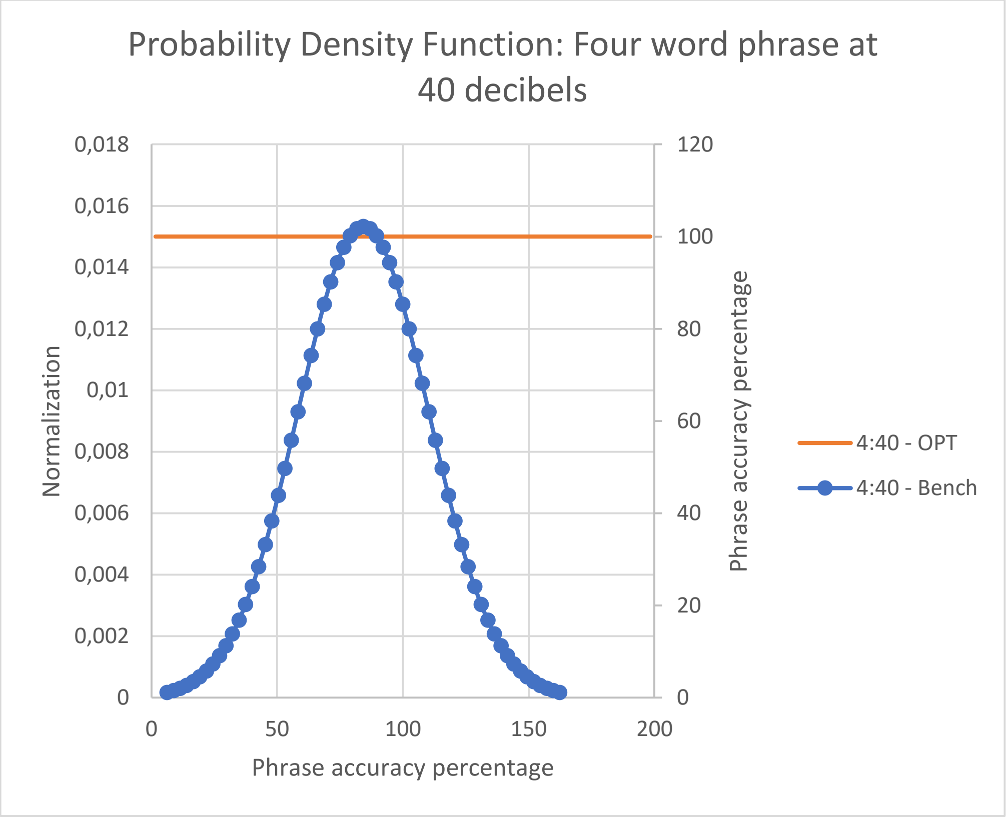


Figure 5-16: Probability Density Function: Four-word phrase at 40 decibels

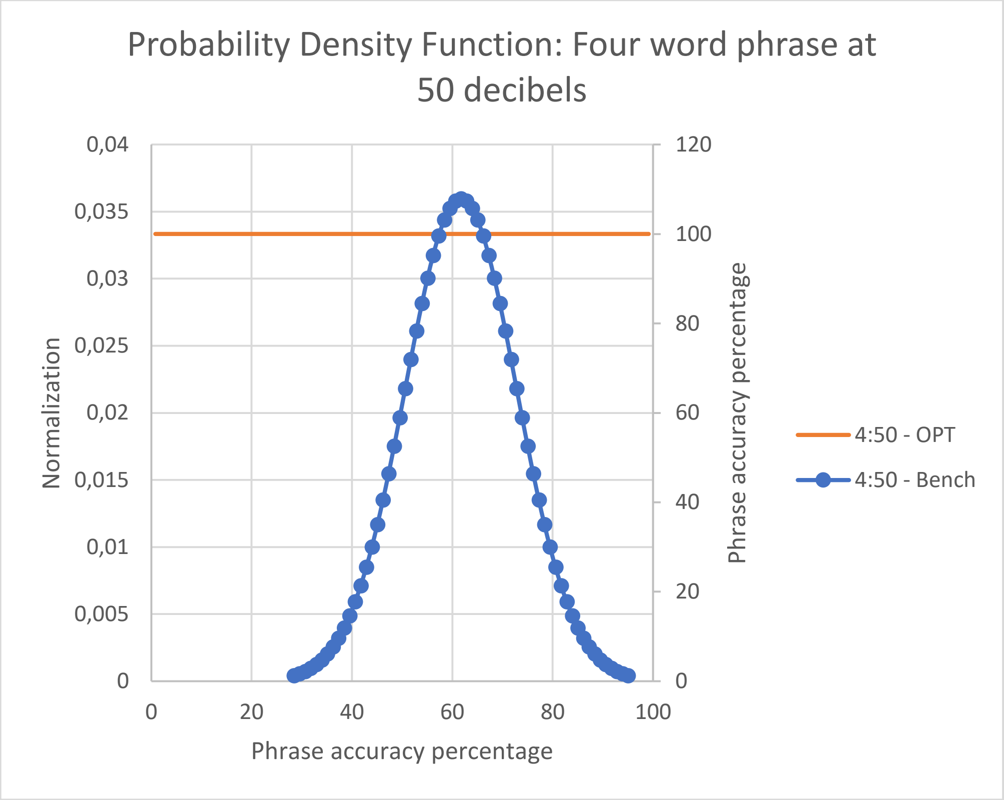


Figure 5-17: Probability Density Function: Four-word phrase at 50 decibels

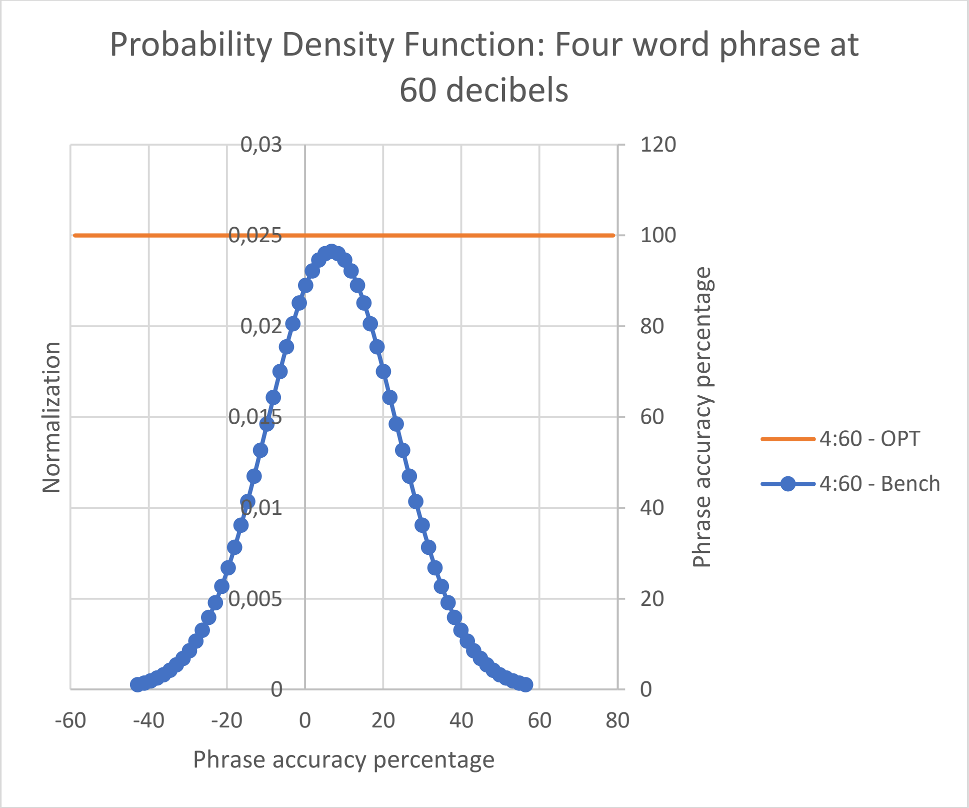


Figure 5-18: Probability Density Function: Four-word phrase at 60 decibels

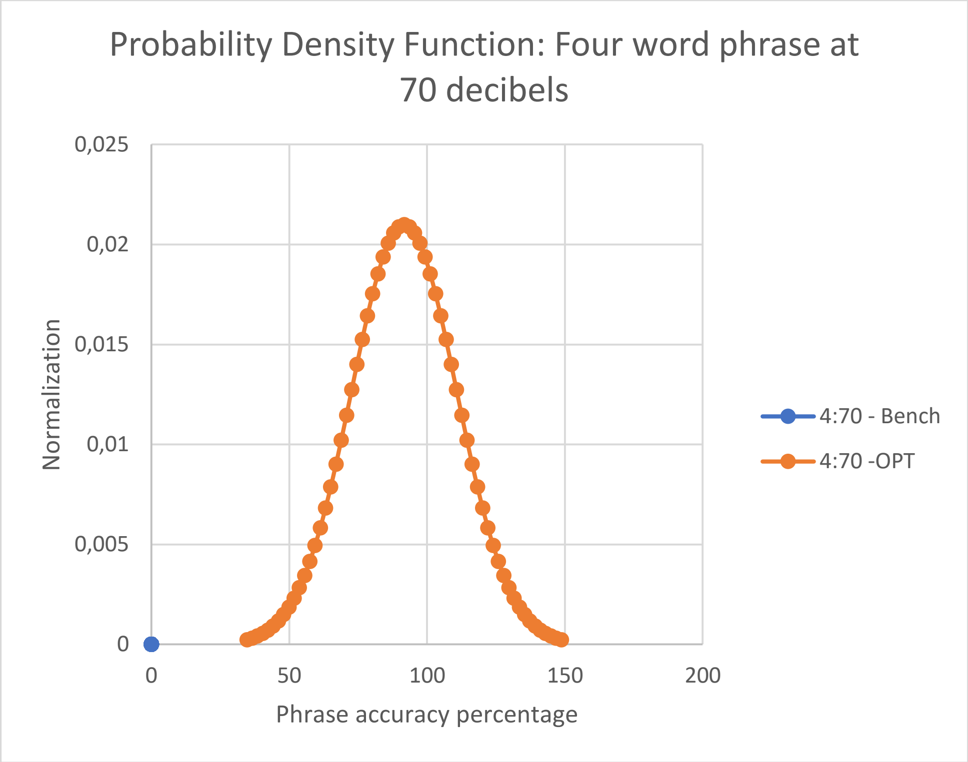


Figure 5-19: Probability Density Function: Four-word phrase at 70 decibels

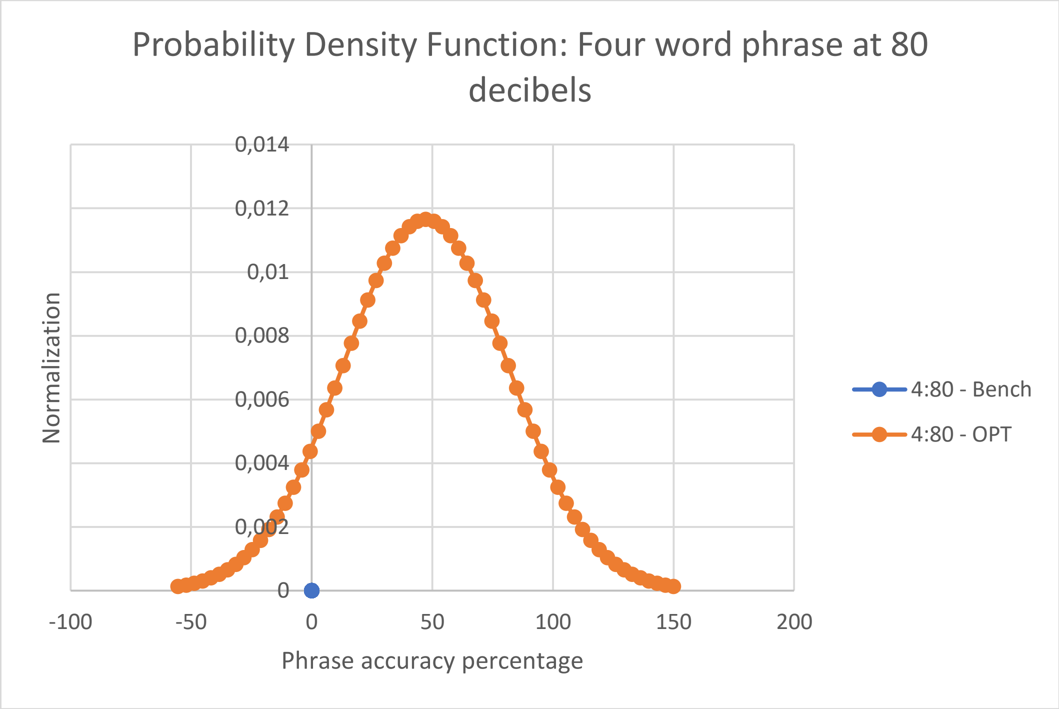


Figure 5-20: Probability Density Function: Four-word phrase at 80 decibels

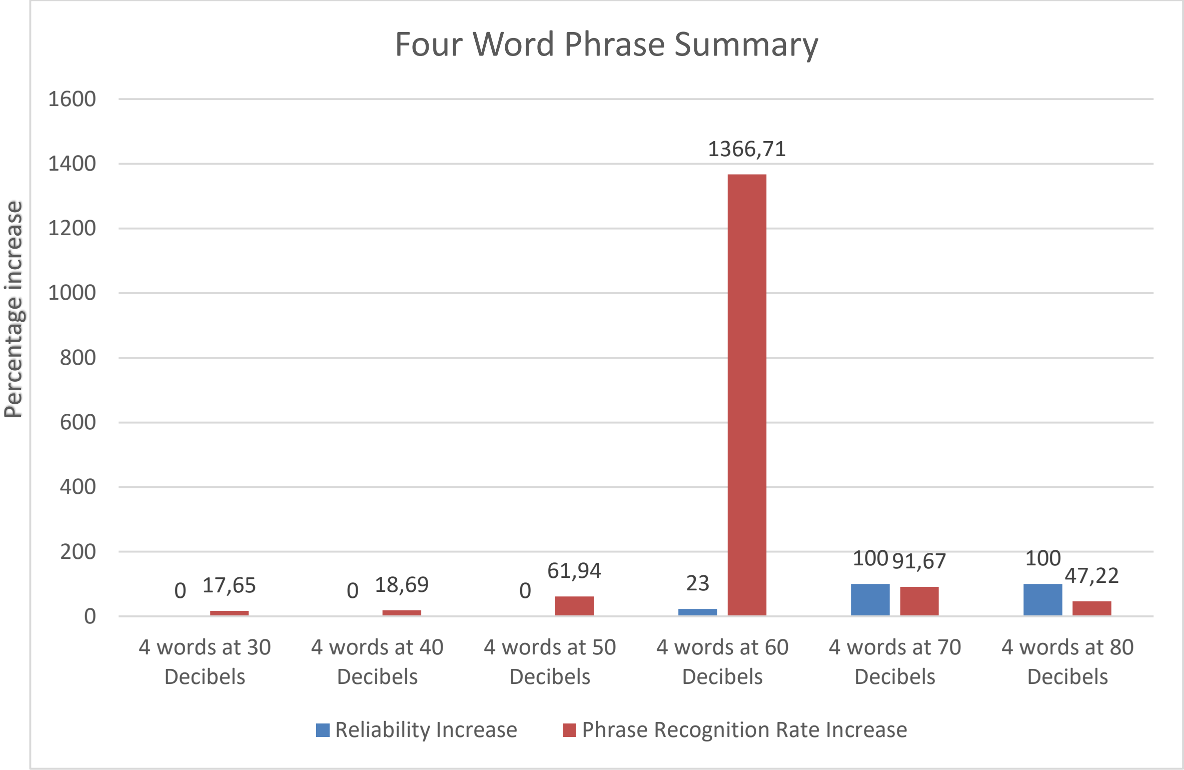


Figure 5-21: Four Word Phrase Summary

5.2.4 Voice interface data: Two-word phrase

The data analysis for the two-word phrase subject to conditions mentioned in chapter three and chapter four can be seen in the section that follows. Figure 5-22 – Figure 5-27 represent the probability density curves for the data extracted from the voice systems designed previously. The corresponding average phrase recognition accuracy corresponding to Figure 5-22 – Figure 5-27 can be viewed in Appendix E in Figure 13-19 – Figure 13-24. Figure 5-28 represents the summary of the “reliability” and average “phrase accuracy increases” for the two-word phrase when compared to the operational benchmark data. Note that Figure 5-28 represents the increases only, if any, of the “reliability” and average “phrase accuracy increases” for the two-word phrase. As can be seen from Figure 5-22 two curves are presented. The probability density function curve displays the spreads of the operational benchmark data, and due to the nature of the optimization results, the consistency did not allow for the generation of a probability density function. The probability density function displayed in Figure 5-22 showed a 75.17 % - 115.83 % range of the data surrounding the mean at one standard deviation. The optimization results however proved to be very favourable with a consistent 100 % recognition rate. The average operational benchmark increase was shown to be 6.95 %. The probability density function displayed in Figure 5-23 showed a 58.03 % - 106.97 % range of the data surrounding the mean at one standard deviation. The optimization results however proved to be very favourable with a consistent 100 % recognition rate. The average operational benchmark increase was shown to be 21.21 %. The probability density function displayed in Figure 5-24 showed a -13.6 % - 77.58 % range of the data surrounding the mean at one standard deviation. The optimization results however proved to be very favourable with a consistent 100 % recognition rate. The average operational benchmark increase was shown to be 209.67 %. The probability density function displayed in Figure 5-25 showed a -13.28 % - 25.78 % range of the data surrounding the mean at one standard deviation. The optimization results however proved to be very favourable with a consistent 100 % recognition rate. The average operational benchmark increase was shown to be 1500 %. Figure 5-26 showed the inability of the original voice interface to extract data when subjected to the corresponding conditions in Figure 5-26. The "OPT" curve of Figure 5-26 showed a 15.65 % - 92.69 % range of the data surrounding the mean at one standard deviation. The average "OPT" increase was shown to be 54.17 %. Figure 5-27, showed similar conditions to Figure 5-26, with the inability of the original voice interface to extract data when subjected to the corresponding conditions in Figure 5-27. The "OPT" curve of Figure 5-27 showed a constant distribution of 50 % accuracy. The average "OPT" increase was shown to be 50 %.

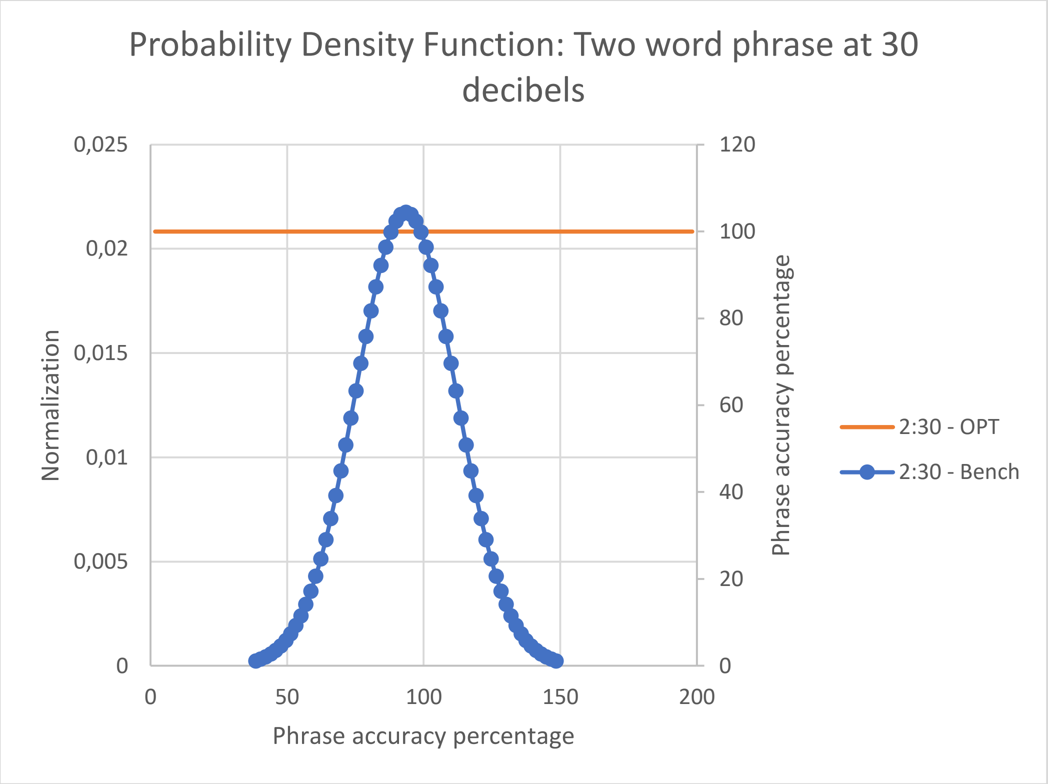


Figure 5-22: Probability Density Function: Two-word phrase at 30 decibels

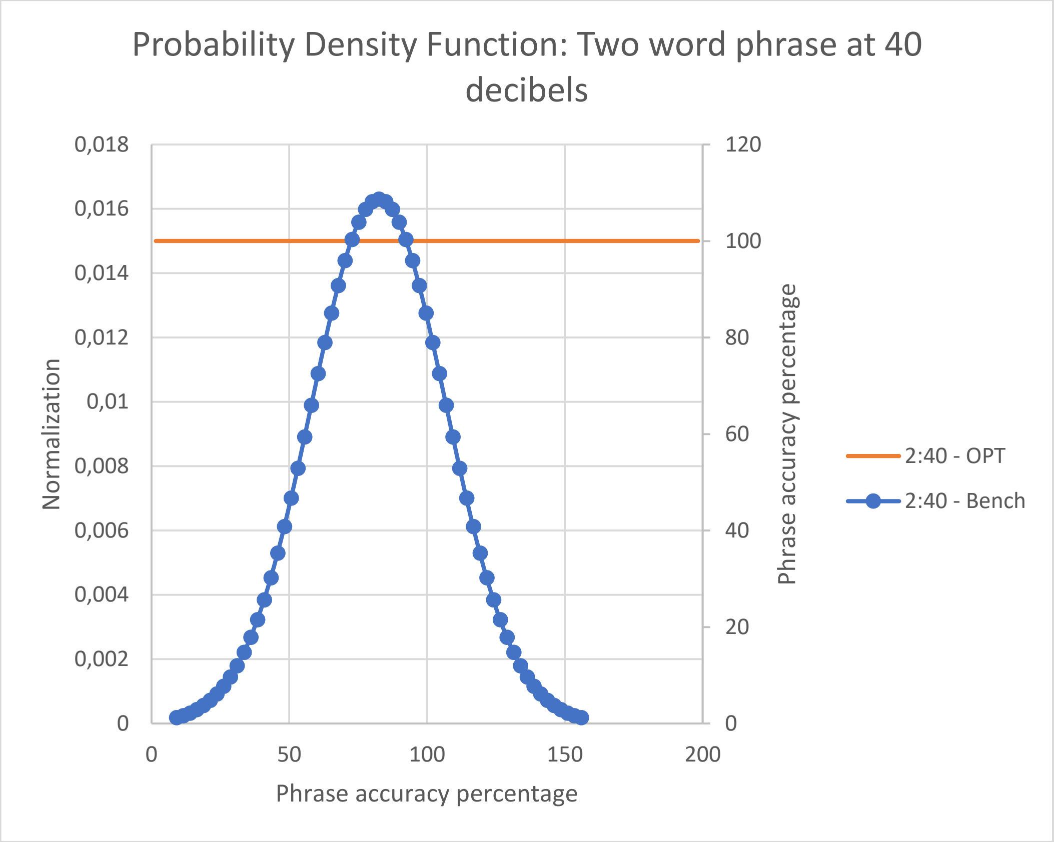


Figure 5-23: Probability Density Function: Two-word phrase at 40 decibels

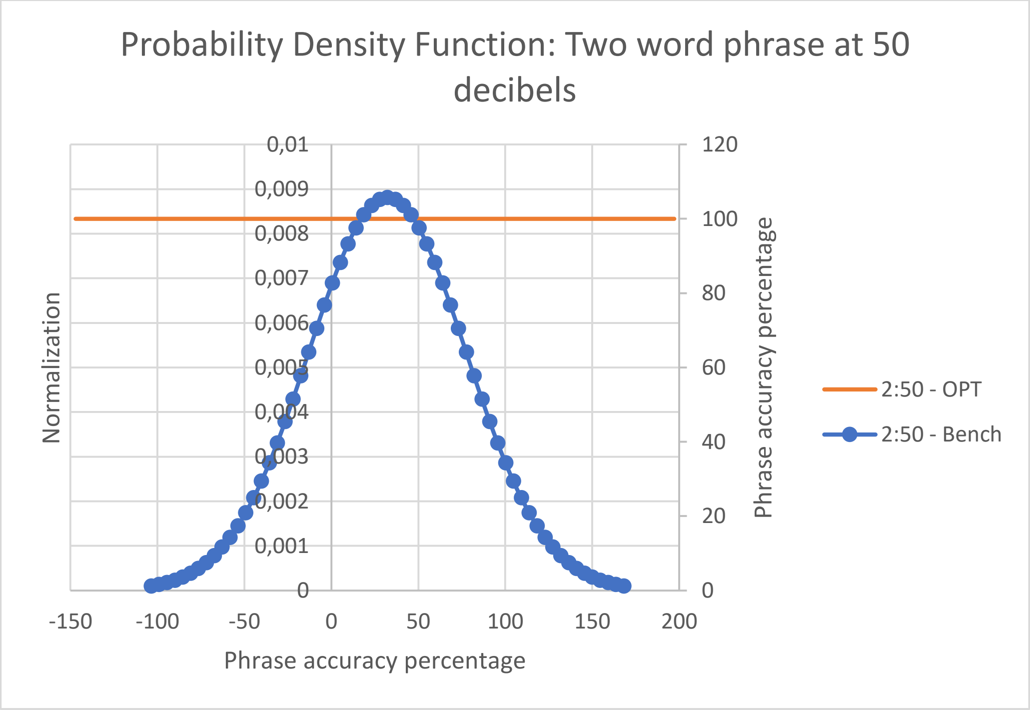


Figure 5-24: Probability Density Function: Two-word phrase at 50 decibels

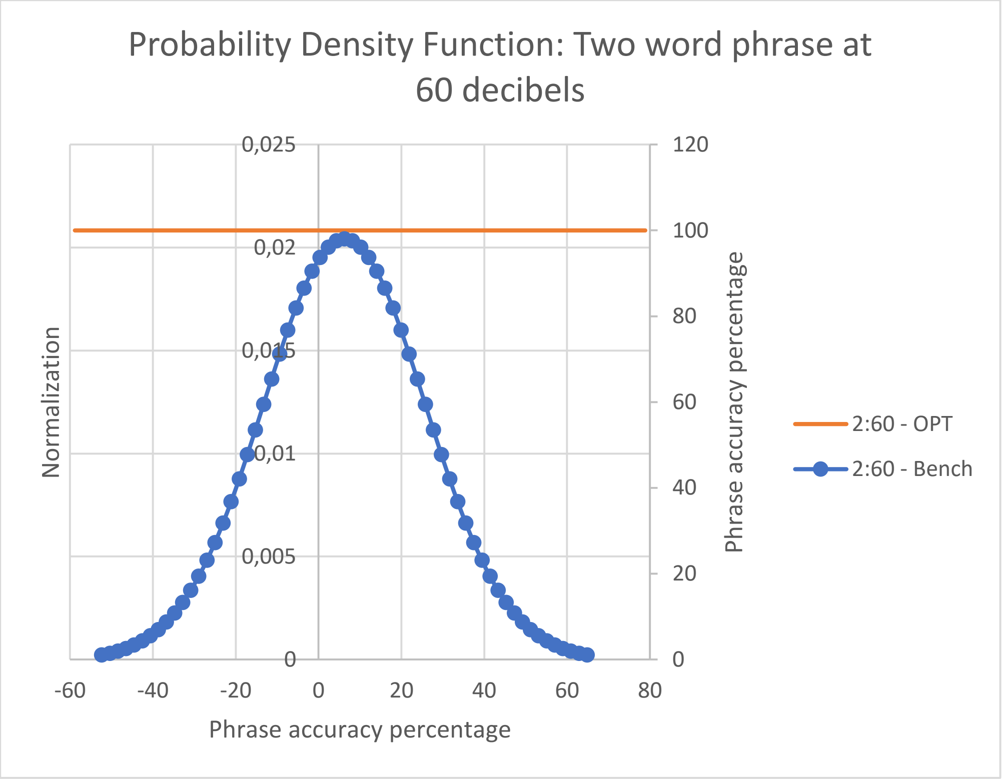


Figure 5-25: Probability Density Function: Two-word phrase at 60 decibels

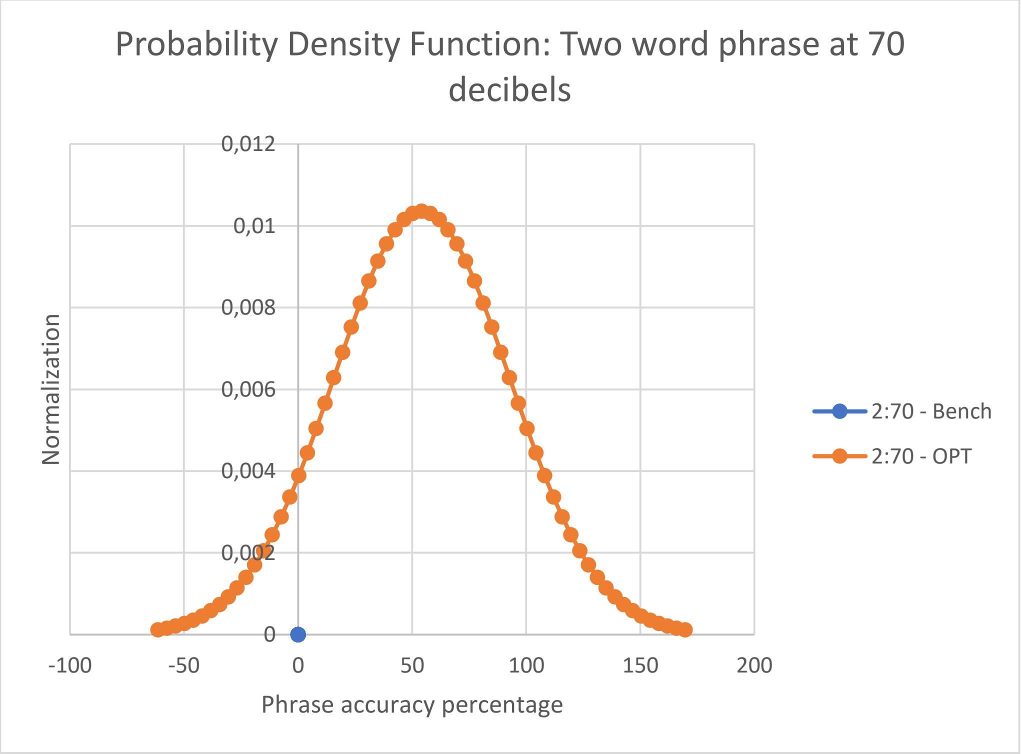


Figure 5-26: Probability Density Function: Two-word phrase at 70 decibels

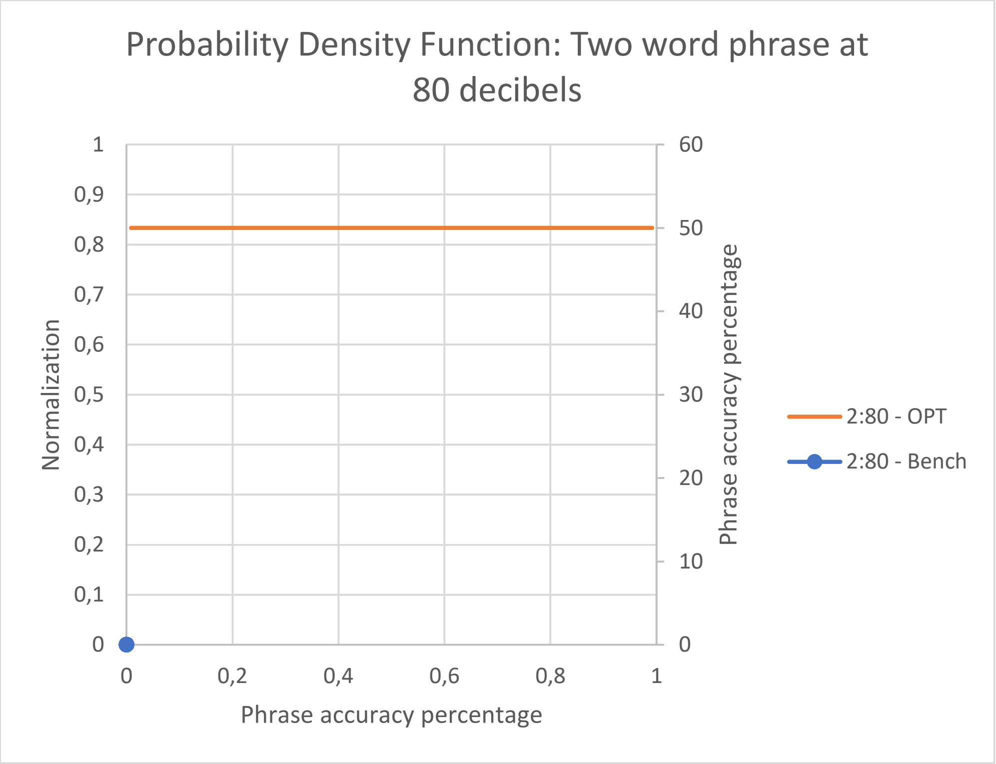


Figure 5-27: Probability Density Function: Two-word phrase at 70 decibels

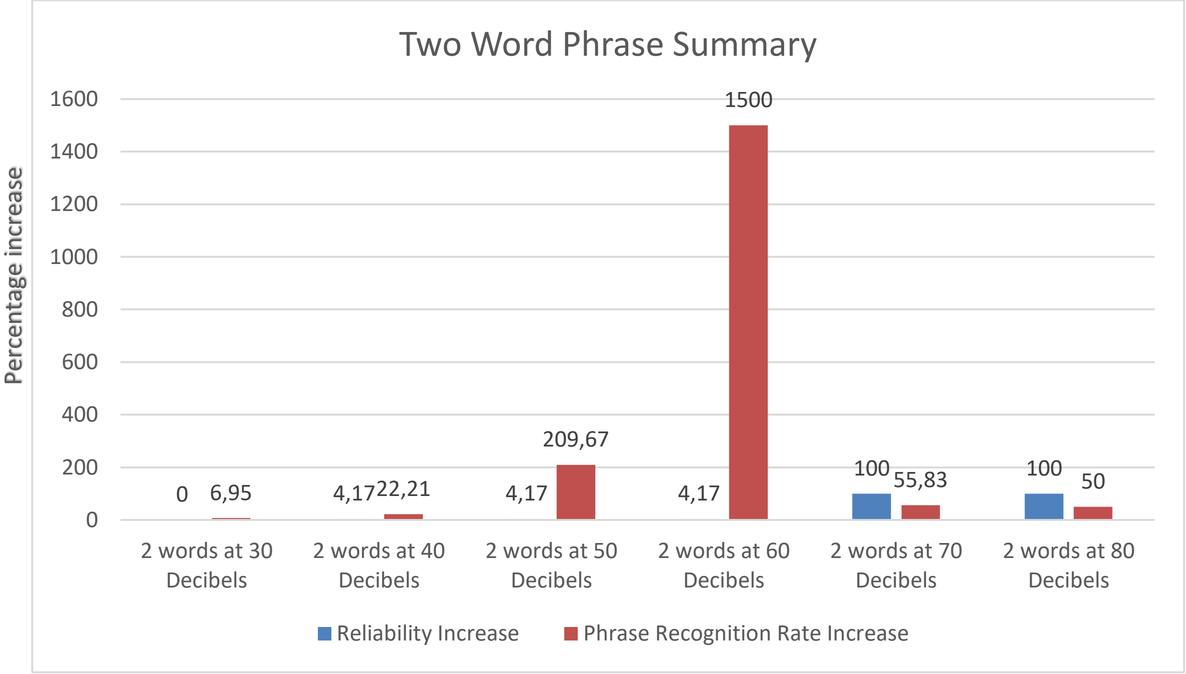


Figure 5-28: Two Word Phrase Summary

5.2.5 Voice interface data: One-word phrase

The data analysis for the One-word phrase subject to conditions mentioned in chapter three and chapter four can be seen in the section that follows. Figure 5-29 – Figure 5-34 represent the probability density curves for the data extracted from the voice systems designed previously. The corresponding average phrase recognition accuracy corresponding to Figure 5-29 – Figure 5-34 can be viewed in Appendix E in Figure 13-25 - Figure 13-30. Figure 5-35 represents the summary of the "reliability" and average "phrase accuracy increases" for the one-word phrase when compared to the operational benchmark data. Note that Figure 5-35 represents the increases only, if any, of the "reliability" and average "phrase accuracy increases" for the one-word phrase. As can be seen from Figure 5-29 two curves are presented. The probability density function curve displays the spreads of the operational benchmark data, and due to the nature of the optimization results, the consistency did not allow for the generation of a probability density function. The probability density function displayed in Figure 5-29 showed a 22.52 % - 115.48 % range of the data surrounding the mean at one standard deviation. The optimization results however proved to be very favourable with a consistent 100 % recognition rate. The average operational benchmark increase was shown to be 44.93 %. The probability density function displayed in Figure 5-30 showed a 17.31 % - 130.61 % range of the data surrounding the mean at one standard deviation. The optimization results however proved to be very favourable with a consistent 100 % recognition rate. The average operational benchmark increase was shown to be 35.21 %. The probability density function displayed in Figure 5-31 showed a 38.66 % - 123.04 % range of the data surrounding the mean at one standard deviation. The optimization results however proved to be very favourable with a consistent 100 % recognition rate. The average operational benchmark increase was shown to be 23.69 %. The probability density function displayed in Figure 5-32 showed a 8.15 % - 133.03 % range of the data surrounding the mean at one standard deviation. The optimization results however proved to be very favourable with a consistent 100 % recognition rate. The average operational benchmark increase was shown to be 41.66 %. Figure 5-33 showed the inability of the original voice interface to extract data when subjected to the corresponding conditions in Figure 5-33. The "OPT" curve of Figure 5-33 showed a constant distribution of 100 % accuracy. The average "OPT" increase was shown to be 100%. Figure 5-34, showed similar conditions to Figure 5-33, with the inability of the original voice interface to extract data when subjected to the corresponding conditions in Figure 5-34. The "OPT" curve of Figure 5-34 showed a -0.2519 % - 100.25 % range of the data surrounding the mean at one standard deviation. The average "OPT" increase was shown to be 50 %.

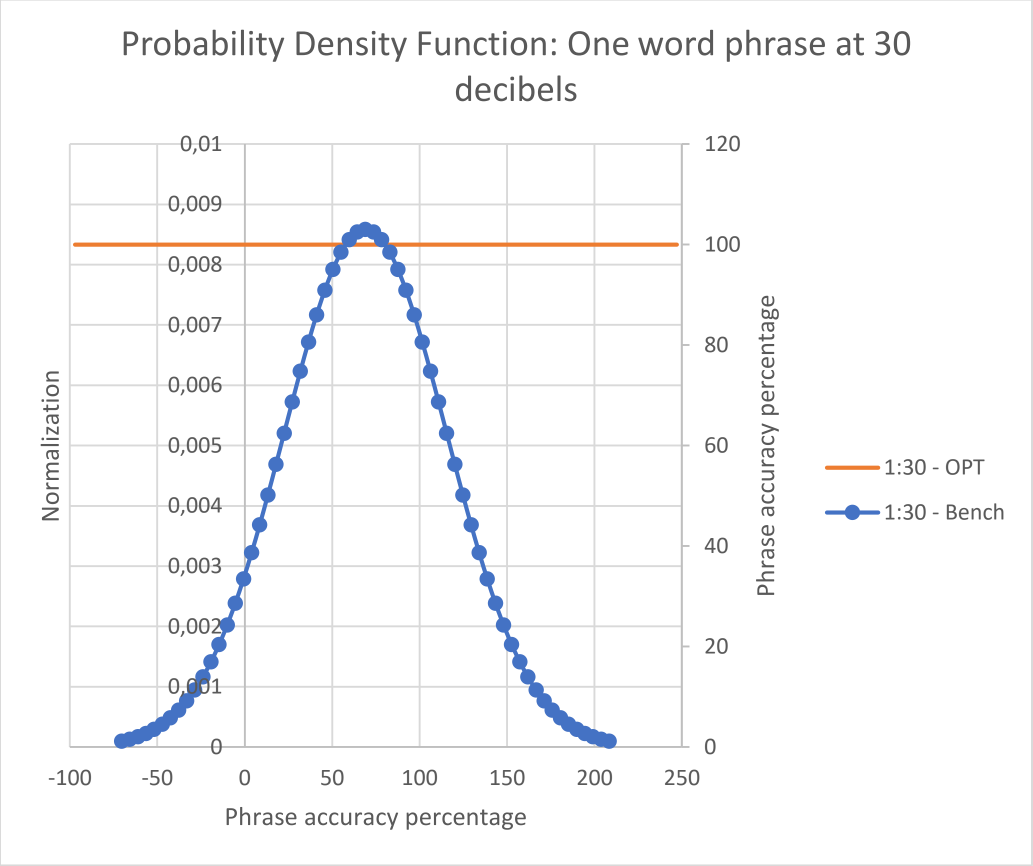


Figure 5-29: Probability Density Function: One-word phrase at 30 decibels

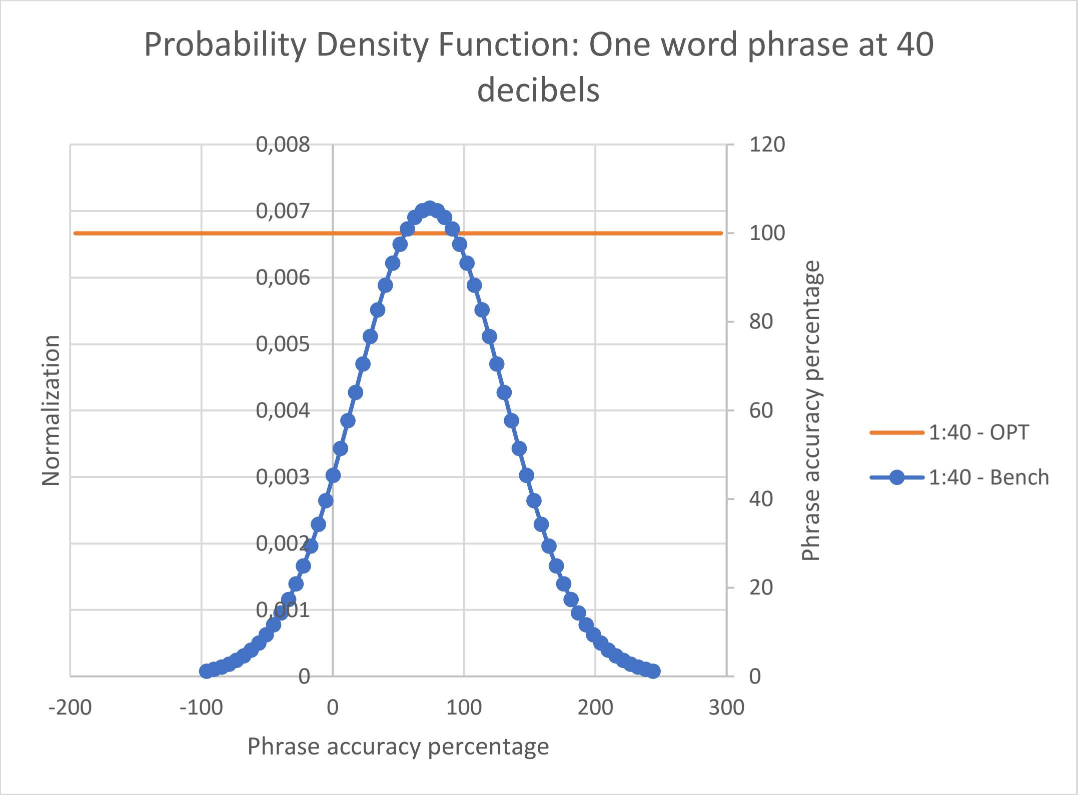


Figure 5-30: Probability Density Function: One-word phrase at 40 decibels

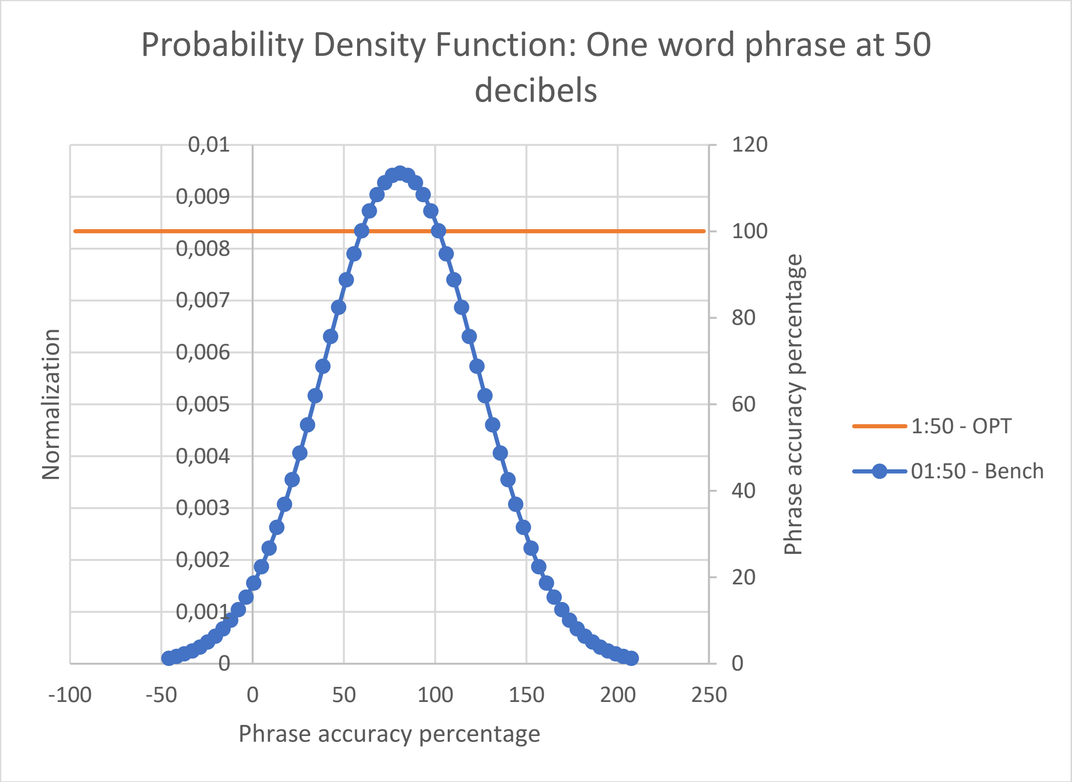


Figure 5-31: Probability Density Function: One-word phrase at 50 decibels

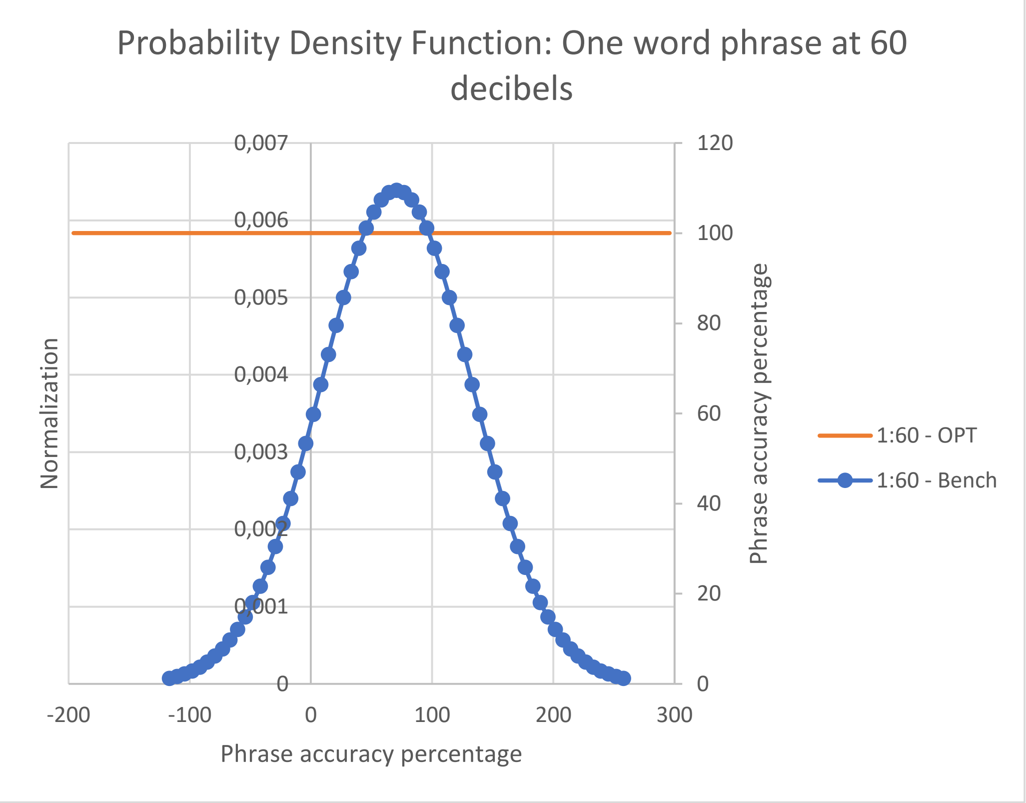


Figure 5-32: Probability Density Function: One-word phrase at 60 decibels

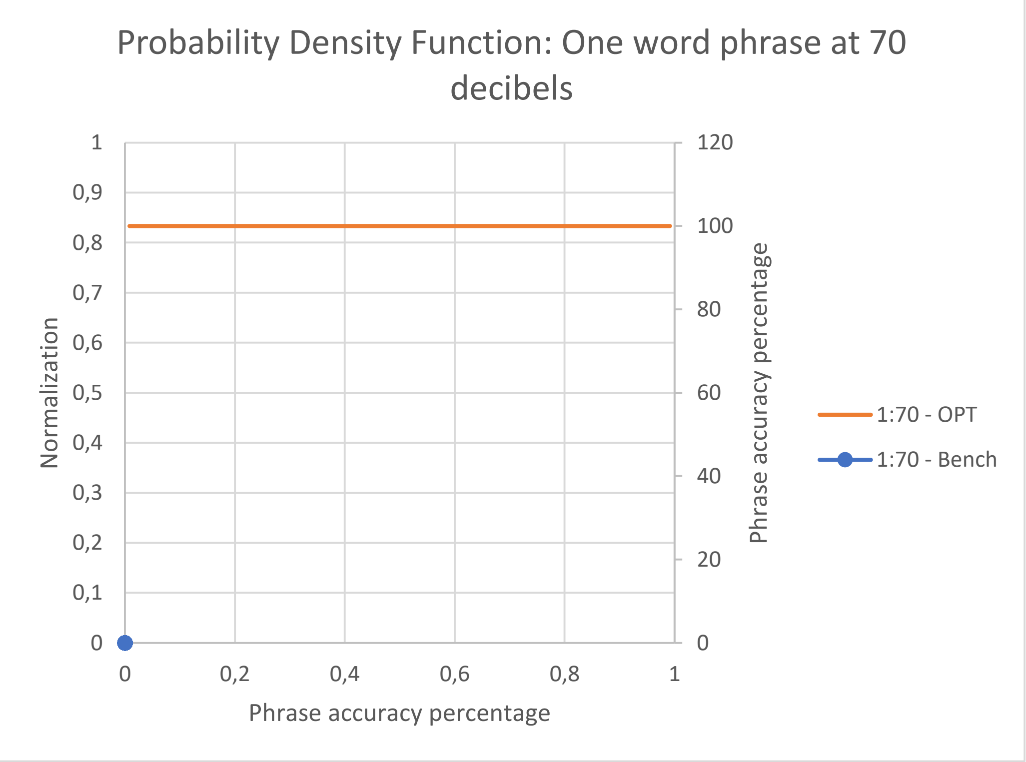


Figure 5-33: Probability Density Function: One-word phrase at 70 decibels

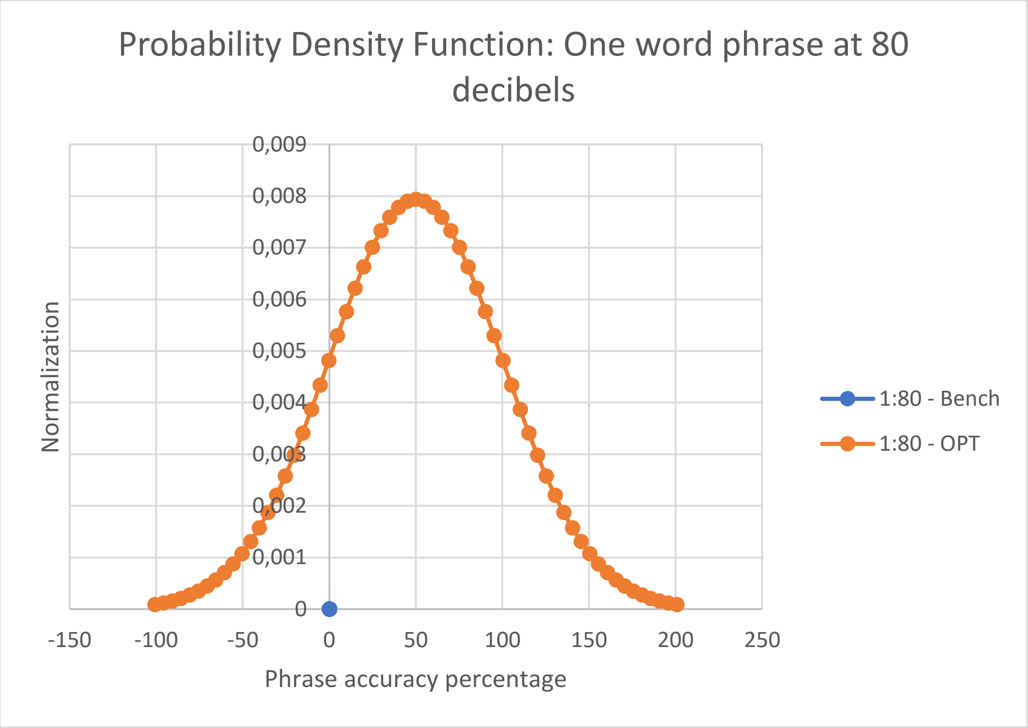


Figure 5-34: Probability Density Function: One-word phrase at 80 decibels

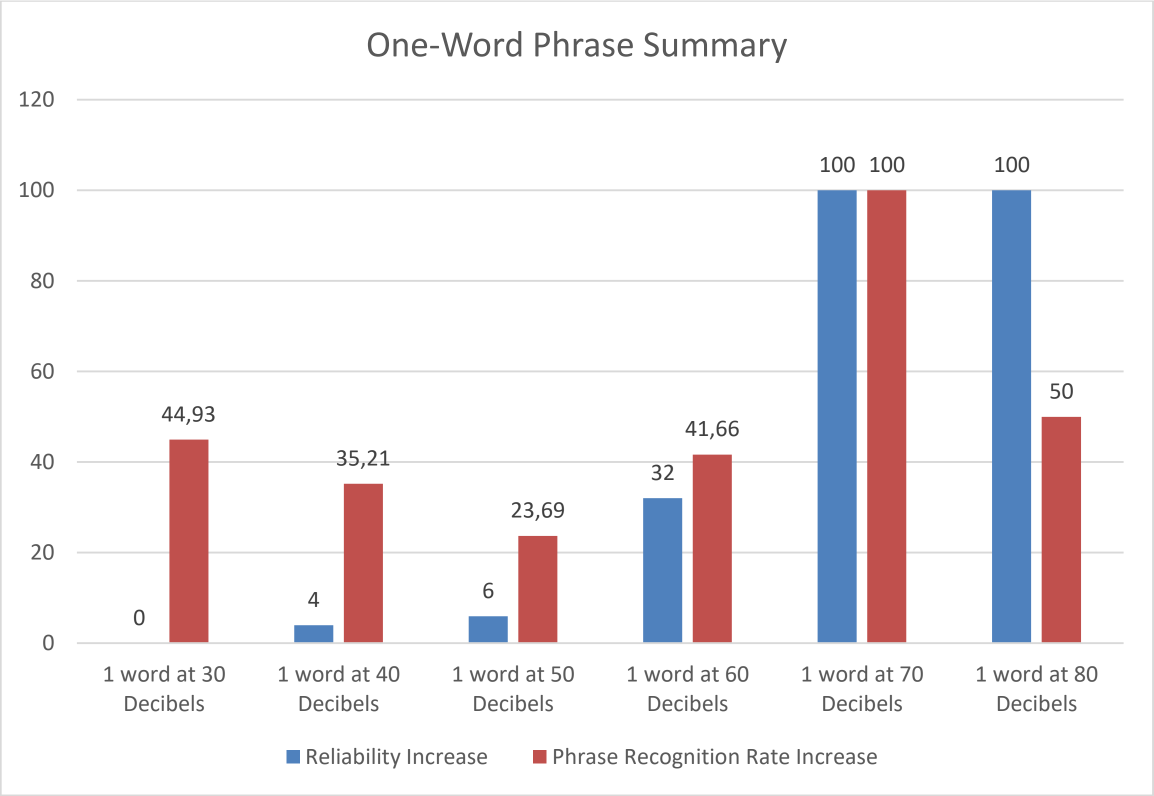


Figure 5-35: One-Word Phrase Summary

5.3 Mobile navigation - System analysis

The data presented in this chapter resembles the accuracy of the mobile navigation robot. Due to the nature of the mobile robot creating a platform for integration of voice interfaces into the manufacturing domain, the key focus of the testing was based on location accuracy testing. The accuracy testing results can be seen in Figure 5-36 - Figure 5-40. Five different instances were tested. The instances included a 3 m, 5 m, 7 m, 10 m and 15 m range test. Each instance ran ten iterations of results to gauge appropriate accuracy results. As can be seen from Figure 5-36 - Figure 5-40, accuracy was limited to a minimum of 111.5 cm and a maximum of 285.5 cm from the desired location. The main reason for the limited accuracy was due to the compass module being very sensitive to the magnetic fields around it. The path accuracy of the design was measured using the GPS location data of the system.

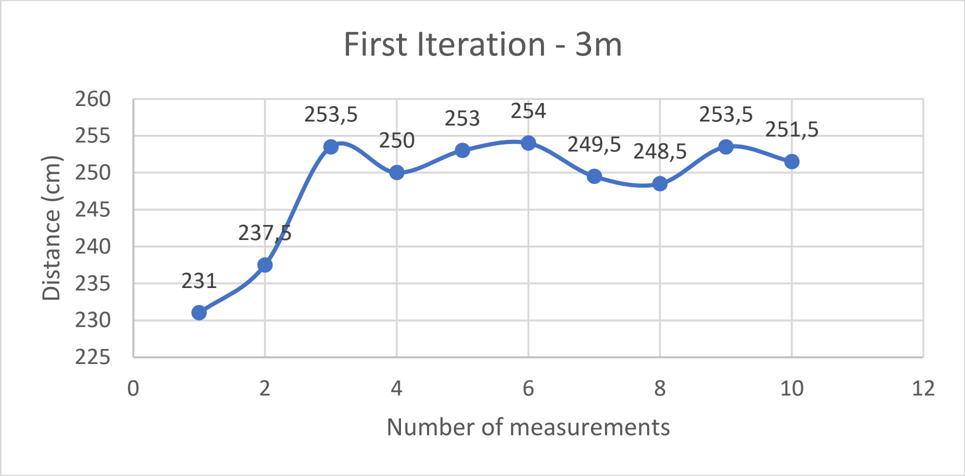


Figure 5-36: First Iteration

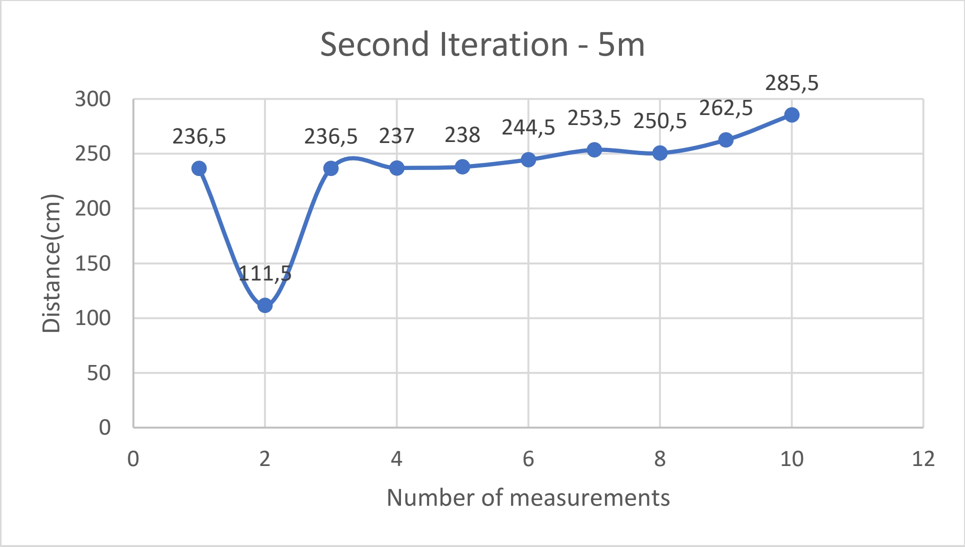


Figure 5-37: Second Iteration

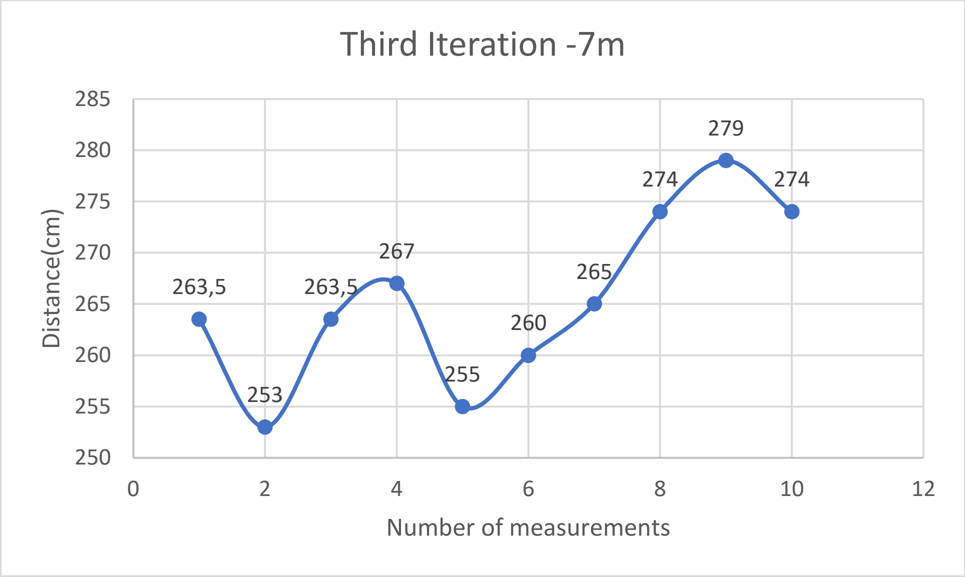


Figure 5-38: Third Iteration

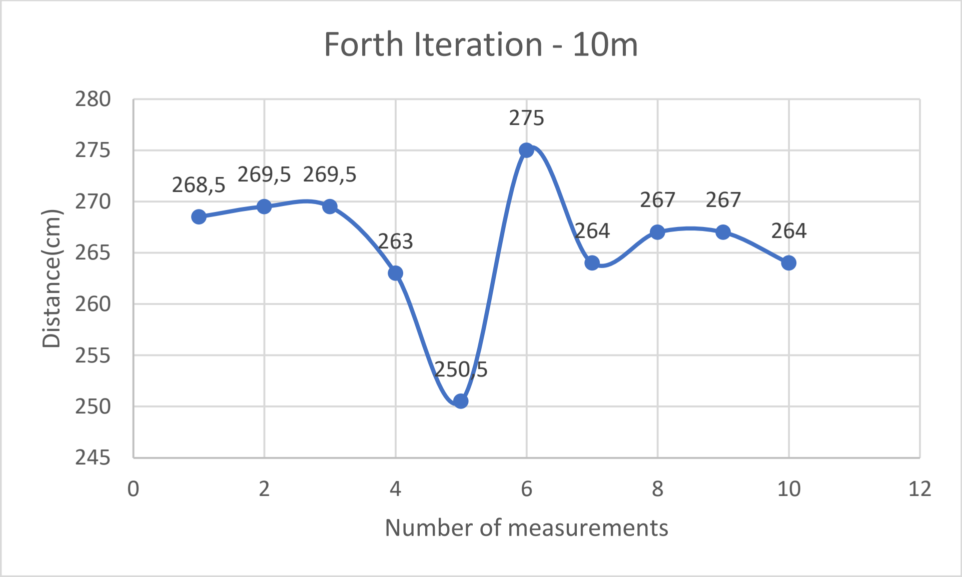


Figure 5-39: Forth Iteration

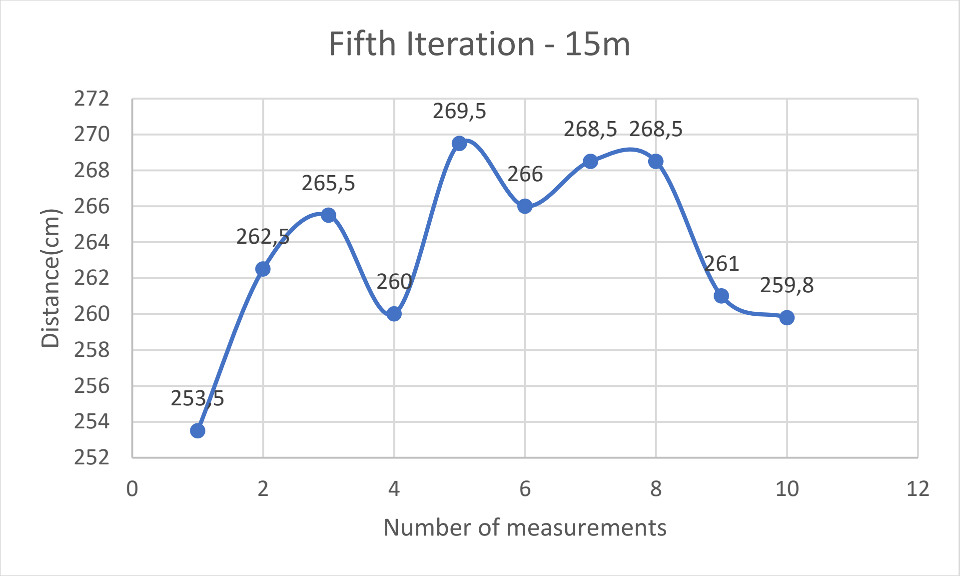


Figure 5-40: Fifth Iteration

5.4 Chapter summary

Chapter five displayed the Data Analysis Chapter. The chapter presented the operational benchmark data obtained when the voice interface system was subjected to industrial noise interference conditions. Additionally, data for optimization of the benchmark system and the accuracy of the navigation system were also presented. An extensive investigation of the voice interface data is presented in this chapter and probabilistic methods were implemented in this chapter.

6. DISCUSSION

The early stages of the dissertation saw the authors attempting to gauge a good understanding of the problem areas within the manufacturing domain. It was established from delving into research, that customizability had essentially created a halt in manufacturing environments. Upon carrying out further research into the field of manufacturing customizability, one such halting factor within customizable manufacturing presented itself as being invariably persistent. That factor being flexibility. In modern times, it was shown that customers opted for products that need not specifically apply to aesthetics, but changing physical characteristics. As such, manufacturing environments needed to not simply be modified, but rather – to evolve. It was found that the reasons for flexibility causing a manufacturing halt were; the development of flexible systems being confined to the academic domain with less focus placed on industrial applications, the absence of a methodological structure for implementing flexible systems into the industrial domain, the absence of due consideration being given to human integration of flexible systems, excessive programming times required for system reconfigurability, high costs associated with the development of a highly flexible manufacturing environment, and high costs associated with adapting pre-existing systems to provide flexibility.

As such, while progression was made during the course of the dissertation, the factors mentioned above were kept very close in mind. It was established that though a system may be improved on through vigorous research, if the factors mentioned above weren't considered during the course of any contribution being made, then the contribution wouldn't really be applicable for implementation, and again, will be confined to the academic domain. The factors mentioned above were not only validated from research, but from consultation with industrial experts as can be seen in Chapter 3. Three main factors were considered from the aforementioned; that being the development of flexible systems being confined to the academic domain with less focus placed on industrial applications, the absence of a methodological structure for implementing flexible systems into the industrial domain, and the absence of due consideration being given to human integration of flexible systems. As such, imbedded in the core structure of the methodology of this dissertation were these factors.

Due to the nature of the dissertation aligning itself to the manufacturing domain, naturally, the fourth industrial revolution was considered. It was found that development within the domain was proliferating with instances of digitalized system, as such, cyber-physical systems were considered and accentuated on. From Chapter 2, literature proved to refine the manufacturing domain through effectively analyzing, monitoring, and creating an interoperable, automated, cyber-machine-centred environment. Further research found that cyber-physical systems focused on data acquisition and sensor-based machine communication for digital platforms. From these areas, flexibility with the communication aspect of cyber-physical systems were investigated. The reason for this choice was taken after research proved to require high levels of intricacy and precision associated with robotic operations, therefore, requiring highly flexible communication interfaces.

As can be seen through the literature review, after due consideration was taken into the problem's areas of manufacturing, an overview of the areas of concern was presented. A short review of programming processes of robotic systems was conducted. This allowed the authors to establish that for future work of any programming related to robotic platforms, low programming times and simple software designs should be considered. Following the literature presented on programming processes of robotic systems, research delved straight into communication networks. The areas considered were; cloud networks, cloud service models, cloud computing deployment models, cloud computing in manufacturing, wi-fi and bluetooth communication, wireless communication for advanced manufacturing systems, voice interface technology and the field of pattern recognition. From these systems, a variety of problem areas were presented and flexible communication through a voice interface platform was selected. This was due to its ability to provide a conventional shift of tool-centered manufacturing to a new, and more versatile approach. Voice assistants also seemed to provide the ability to address the factors contributing to stagnant growth within manufacturing environments.

Research showed proliferating instances of voice interface technology outside the manufacturing domain, and this posed the question: why does the manufacturing domain lack these technologies? It was found that the field of voice interface technology introduced itself as a relatively new communication medium, and though they displayed rapid advancement, methods of development presented themselves as proprietary work of the technological institutes facilitating advancement. Though many instances of voice interfaces did not present themselves in literature, a few areas of concern within voice interface technology was presented, that being; a lack of implementation in manufacturing environments, high levels of language fluency requirements, low interpretability constraints, having a very high sensitivity to the environment, and a low adaptability to different languages. The question as then posed, is possible to introduce voice interfaces into the manufacturing domain? A new approach was considered to answer the unsolved question above. The approach considered to answer this question was developed in the methodology chapter.

The methodology chapter provided for the structure of the dissertation that followed. It considered the opinion, views and information provided by a senior research expert, industrial mechanical engineers, an industrial software engineer, course enrolment, literature and academic staff. This approach was taken to further delve into the field of voice interface technology – to address core manufacturing concerns presented previously. That being the development of flexible systems being confined to the academic domain with less focus placed on industrial applications, the absence of a methodological structure for implementing flexible systems into the industrial domain, and the absence of due consideration being given to human integration of flexible systems.

The first issue sought to address was the state of voice interface technology, from the perspective of an academic expert. Dr. Birch provided knowledge similar to that provided from literature, that being information of voice interface technology and the advancements of these technology being proprietary to the technological institutes facilitating advancement. It was suggested, that due to the nature of the field of voice interfaces being relatively new, and major developments and data resources being proprietary, it would be a good first step to identify a benchmark of the current operational capabilities of voice interfaces for manufacturing. It was suggested that synthetic data be created from a commonly used device facilitating voice interfaces. This statement was made to establish the limitations and areas of potential improvement for voice interfaces, and to conform to the selected area of research being flexibility. Additionally, for any optimization of voice interfaces, it was suggested to simulate a manufacturing environment and subject the extracted synthetic data to manufacturing conditions. It was also highly recommended, that for the issue of adaptability of voice interfaces to different languages, it would be a good idea to look at web-based services. Consultation with two Industrial system engineers sought to address the issue of lack of implementation of voice interfaces in manufacturing environments. From this consultation, information was obtained as to how to proceed with research aligned to industrial applications. The consultants highly recommended to focus on collaborative human-robot systems, rather than the conventional machine-centered system designs, a concern also found to be halting manufacturing improvement from literature. From an application perspective, it was suggested to find a means of implementing voice interfaces into the maintenance failure process. It was established, that in a manufacturing environment, when a failure occurs, it can be a very time-consuming process to communicate instructions and transfer equipment between the divisions of operations, maintenance and engineering in manufacturing environments. As such, a task based mobile navigation robot design was considered to provide for a collaborative human-robot medium, and to allow for basic equipment handling operations to be carried out using a voice controlled mobile navigation robot. Data was provided by the consultants for subjection of the designed system to industrial conditions. Literature provided for a means of approaching the issue of low interpretability constraints and low adaptability within voice interfaces. It was found that web-based servers could alleviate the problem of low interpretability concerns with adaptability concerns having the ability to addressed using appropriate software methods.

Research from literature validated that the best voice interface technology that should be used was googles “OK Google”. The methodological structure additionally included the information provided from the enrolment into web-based and IoT platform short courses. This initiative was taken to establish a link between voice interfaces and machinery. Consultation with an industrial software expert provided vital knowledge of good automation development practices for industrial machines with the design approach recommended being GAMP®5. From a testing point of view, academic staff were consulted. The vibration and research testing centre were consulted due to the nature of their work directly relating to the industrial domain. The centre carries out testing for power utility Eskom. As such, this department was consulted to establish a suitable testing facility for testing, as well as to establish the equipment that would be required for testing of voice interface technology and their associated calibrations. From the information provided in literature and the methodology chapter, the following research question was posed: how can a mechatronic system be designed for a manufacturing environment that integrates voice interface technology and mobile robotics?

The dissertation investigated a conceptualized methodological approach for the implementation of voice interface technology into advanced manufacturing environments. Voice interfaces technology was investigated to develop an operational benchmark – when subjected to simulated manufacturing conditions, and subsequently, the thesis established an optimization for the accuracy and reliability of a voice interface system – with a proposed method of application for manufacturing environments using a mobile navigation robot design.

As a whole, three systems were designed in the thesis to attempt to validate the proposed research question. The first system designed – System 1, constituted the following devices; A microcontroller (NodeMCU), Decibel range meter, Noise simulation application, Bluetooth module (HC-06), Mobile cellular device (Smart phone), and a laptop to facilitate serial communication. This system was designed to establish an operation benchmark for googles voice interface – “OK Google”, when subjected to industrial conditions. Following the extraction of the operational benchmark data from System 1, it was found that an optimization of the voice interface needed to be carried out for it to be implemented into the manufacturing domain. As such, a second system – System 2, was designed, the system constituted of; a microcontroller (NodeMCU), Google’s voice assistant, a web-based service, IoT platform, mobile cellular device, interference cancelling hardware, and interference cancelling software. Following the optimization approach to System 1, a third system – System 3 was designed. That being the design of a mobile navigation robot to create a platform that facilitated the integration of voice interfaces in the manufacturing domain.

To allow for good testing practices, a simulation arena similar to that presented by the Vibration Research and Testing Centre (VRTC) was created. Taking a look at System 1, the Decibel range meter and noise simulation application were designed. In order to calibrate these devices, calibration testing was carried out at the VRTC lab. The operational benchmark system was subjected to recorded instances of voice phrases to establish the phrase accuracies and system reliability. The optimization system used IFTTT software as a web-based server, Arduino's Blynk application as an IoT platform, a unidirectional interference cancelling microphone, and integrated based on benchmark readings, adaptability considerations within the Arduino IDE code. Testing results proved to be very favourable in all instances of testing however, isolated instances did provide for a small degree of inconsistency.

Testing of the voice interface system was divided into five main categories. The categories included 8-word phrases, 6-word phrases, 4-word phrases, 2-word phrases and a 1-word phrase. The recorded phrases were subjected to the operational benchmark system. Following the establishment of an operational benchmark, the same phrases were subjected to the optimized voice interface system. Each category of the phrases mentioned above was subjected to noise interference readings at a constant frequency. Each phrase category was subjected to noise interference levels starting at 30 decibels in increments of 10 decibels, till a maximum of 80 decibels for the benchmark and optimization system. Within each decibel category, 100 iterations were taken for each phrase category. As such, each phrase category presented itself with 1200 instances of voice readings for analysis, with the total analysis accounting for 6000 voice interface readings. Following the data extraction of the benchmark and optimization systems, the data was subjected to a probability distribution analysis, namely, the probability density function, as a means of identifying the validity of results and means of identifying inconsistencies.

Taking a look at the 8-word phrase dataset analyzed. Results for this category can be viewed in Chapter 5.2.1, and Appendix E. Results showed that as a whole, when subjected to decibel ranges of 30 decibels to 80 decibels, the benchmark system was able to detect phrases up until a limiting 60 decibels. When the benchmark system was subjected to interference of 70 decibels and 80 decibels, the benchmark system was incapable of detecting any form of phrase data. From a reliability perspective, for the benchmark system, a 100% detection rate was achieved for 30-50 decibels. This meant that although readings detected may not have been 100 % accurate, the benchmark system was still able to detect some form of a phrase data. Reliability for the benchmark system decreased to 81 % at a 60 decibel subjection and persisted with a 0 % detection for subjections of 70 decibels and 80 decibels for the benchmark system. When the optimized system was considered, the system outperformed the benchmark system by having a 100 % reliability for subjections of 30-80 decibels of an 8-word phrase. Taking a look at the average operational benchmark data, this showed average recognition readings of 85.375 % , 77 %, 53.375 %, 19.136 %, 0% and 0%, for 30-80 decibel subjections respectively. The optimization system, when compared to the benchmark system - proved that under a 30 decibel subjection, the data remained fairly constant – for the average phrase recognition rate, with no increased being noticeable. This ideally meant that for an 8-word phrase being subjected to a 30-decibel interference, the IoT system performed just as well as a non-IoT system. For the 40-decibel subjection, a noticeable increase of 0.65 % was noted, for the 50 decibel subjection, a noticeable increase of 52.21 % was noted, for the 60-decibel subjection, a noticeable increase of 378 % was noted, and for the subjections of 70 and 80 decibels, with a 0 % recognition rate initially being noted, a new reignition of 94.54 % and 12.5 % was noted with a persistent 100 % reliability throughout the subjections. Looking at the data distribution to try and establish the true nature of the data obtained, it was found that for the 30 decibel subjection, for data within one and two standard deviations of the mean, that there was a distinct increase of lower limits and decrease of the upper limits – for the 8-word phrase. With this key increase and decrease on the respective limits being noticed, this proved to be highly favourable as this meant that the data for the optimized system was more highly centered and concentrated around the mean. The increases and decreases noted, for the first and second deviations, of an 8-word phrase were noted to be (Note the first number within {" "," "} refers to the lower limit, with the second number corresponding to the upper limit of the data.); {5.75 %, 3.92 %} and {15.61 %, 6.87 %} for 30 decibels, {28.99 %, 13.37 %} and {156.98 %, 23.15%} for 40 decibels, {151.04 %, 22.82 %} and {0 %, 8.71 %} for 50 decibels, and {11.23 % , 119.61 %} and {0 %, 52.15 %} for 60 decibels. Due to the non-detection of the benchmark system, the limits for one and two standard deviations of 70-80 decibel subjections were {88.62 %, 100.96 %}, {82.46 %, 107.12 %} and {-3.78 %, 28.78 %}, {-20.07 %, 45.07 %}, respectively.

Taking a look at the 6-word phrase dataset analyzed. Results for this category can be viewed in Chapter 5.2.2 and Appendix E. Note that in the data that is presented, the first number within {"a ", "b "} refers to the lower limit – "a", with the second number corresponding to the upper limit of the data – "b". Results showed that as a whole, when subjected to decibel ranges of 30 decibels to 80 decibels, the benchmark system was able to detect phrases up until a limiting 60 decibels. When the benchmark system was subjected to interference of 70 decibels and 80 decibels, the benchmark system was incapable of detecting any form of phrase data. From a reliability perspective, for the benchmark system, a 100% detection rate was achieved for 30-40 decibels. This meant that although readings detected may not have been 100 % accurate, the benchmark system was still able to detect some form of a 6-word phrase. Reliability for the benchmark system decreased to 71 % at a 50 decibel subjection, 68 % at a 60-decibel subjection and persisted with a 0 % detection for subjections of 70 decibels and 80 decibels for the benchmark system. When the optimized system was considered, the system outperformed the benchmark system by having a 100 % reliability for subjections of 30-80 decibels of a 6-word phrase. Taking a look at the average operational benchmark data, this showed recognition readings of 85.67 % , 77.17 %, 53.99 %, 34.07 %, 0 % and 0 %, for 30-80 decibel subjections respectively. The optimization system, when compared to the benchmark system - proved that under a 30 decibel subjection, the average phrase recognition rate increased by noticeable 16.73 %. For the 40 decibel subjection, a noticeable increase of 29.58 % was noted, for the 50 decibel subjection, a noticeable increase of 85.22 % was noted, for the 60 decibel subjection, a noticeable increase of 193.51 % was noted, and for the subjections of 70 and 80 decibels, with a 0 % recognition rate initially being noted, a new recognition of 98.17 % and 91.18 % was noted with a persistent 100 % reliability throughout the subjections. Looking at the data distribution to try and establish the true nature of the data obtained, it was found that for the 30 decibel subjection, for data within one and two standard deviations of the mean, that there was a distinct increase of lower limits and decrease of the upper limits – for the 6-word phrase. With this key increase and decrease on the respective limits being noticed, this proved to be highly favourable as this meant that the data for the optimized system was more highly centered and concentrated around the mean. The increases and decreases noted, for the first and second deviations, of a 6-word phrase were noted to be; {58.55 %, 8.27 %} and {147.10 %, 30.87 %} for 30 decibels, {100 %, 4.45 %} and {342.48 %, 31.74 %} for 40 decibels. For the subjection of 50 decibels, the first and second standard deviation limits of the benchmark system were given by {24.78 %, 83.19 %} and {-4.42 %, 112.4 %}, with increased in both cases to a 100% consistency distribution. For the subjection of 60 decibels, the first and second standard deviation limits of the benchmark system were given by {5.36 %, 62.78 %} and {-23.36 %, 91.5%}, with increased in both cases to a 100 % consistency distribution. Due to the non-detection of the benchmark system, the limits for one and two standard deviations of 70-80 decibel subjections were {92.93 %, 103.41 %}, {87.69 %, 108.65 %} and {66.51 %, 115.55 %}, {41.84 %, 140.52 %}, respectively.

Taking a look at the 4-word phrase dataset analyzed. Results for this category can be viewed in Chapter 5.2.3, and Appendix E. Note that in the data that is presented, the first number within {“a ”, ”b ”} refers to the lower limit – “a”, with the second number corresponding to the upper limit of the data – “b”. Results showed that as a whole, when subjected to decibel ranges of 30 decibels to 80 decibels, the benchmark system was able to detect phrases up until a limiting 60 decibels. When the benchmark system was subjected to interference of 70 decibels and 80 decibels, the benchmark system was incapable of detecting any form of phrase data. From a reliability perspective, for the benchmark system, a 100% detection rate was achieved for 30-50 decibels. This meant that although readings detected may not have been 100 % accurate, the benchmark system was still able to detect some form of a 4-word phrase. Reliability for the benchmark system decreased to 77 % at a 60-decibel subjection and persisted with a 0% detection for subjections of 70 decibels and 80 decibels for the benchmark system. When the optimized system was considered, the system outperformed the benchmark system by having a 100% reliability for subjections of 30-80 decibels of a 4-word phrase. Taking a look at the average operational benchmark data, this showed recognition readings of 85 % ,84.25 %, 61.75 %,6.82 %, 0 % and 0%, for 30-80 decibel subjections respectively. The optimization system, when compared to the benchmark system - proved that under a 30-decibel subjection, the average phrase recognition rate increased by noticeable 17.65 %. For the 40-decibel subjection, a noticeable increase of 18.69 % was noted, for the 50-decibel subjection, a noticeable increase of 61.94 % was noted, for the 60-decibel subjection, a noticeable increase of 1366.71 % was noted, and for the subjections of 70 and 80 decibels, with a 0 % recognition rate initially being noted, a new recognition of 91.67 % and 47.22 % was noted with a persistent 100 % reliability throughout the subjections. Looking at the data distribution to try and establish the true nature of the data obtained, it was found that for the 30 decibel subjection, for data within one and two standard deviations of the mean, that there was a distinct increase of lower limits and decrease of the upper limits – for the 4-word phrase. With this key increase and decrease on the respective limits being noticed, this proved to be highly favourable as this meant that the data for the optimized system was more highly centered and concentrated around the mean. The increases and decreases noted, for the first and second deviations, of a 4-word phrase were noted to be; {58.91 %, 8.57 %} and {149.35 %, 31.4 %} for 30 decibels, {71.76 %, 10.28 %} and {210.62 %, 36.31 %} for 40 decibels, and {97.43 %, 37.27 %} and {152.78 %, 19.13 %} for 50 decibels. For the subjection of 60 decibels, the first and second standard deviation limits of the benchmark system were given by {-9.72 % ,23.36 %} and {-26.27 %, 39.9 %}, with increased in both cases to a 100 % consistency distribution. Due to the non-detection of the benchmark system, the limits for one and two standard deviations of 70-80 decibel subjections were {72.66 %, 110.68 %}, {53.66 %, 129.68 %} and {12.97 %, 81.47 %}, {-21.27 %, 115.71 %}, respectively.

Taking a look at the 2-word phrase dataset analyzed. Results for this category can be viewed in Chapter 5.2.4, and Appendix E. Note that in the data that is presented, the first number within {“a ”, ”b ”} refers to the lower limit – “a”, with the second number corresponding to the upper limit of the data – “b”. Results showed that as a whole, when subjected to decibel ranges of 30 decibels to 80 decibels, the benchmark system was able to detect phrases up until a limiting 60 decibels. When the benchmark system was subjected to interference of 70 decibels and 80 decibels, the benchmark system was incapable of detecting any form of phrase data. From a reliability perspective, for the benchmark system, a 100% detection rate was achieved for 30 decibels only. This meant that although readings detected may not have been 100 % accurate, the benchmark system was still able to detect some form of a 2-word phrase. Reliability for the benchmark system decreased to 96 % at a 40-60 decibel subjection and persisted with a 0% detection for subjections of 70 decibels and 80 decibels for the benchmark system. When the optimized system was considered, the system outperformed the benchmark system by having a 100% reliability for subjections of 30-80 decibels of a 2-word phrase. Taking a look at the average operational benchmark data, this showed recognition readings of 93.5 %, 82.5 %, 32.29 %, 6.25 %, 0 % and 0 %, for 30-80 decibel subjections respectively. The optimization system, when compared to the benchmark system - proved that under a 30 decibel subjection, the average phrase recognition rate increased by noticeable 6.95 %. For the 40 decibel subjection, a noticeable increase of 22.21 % was noted, for the 50-decibel subjection, a noticeable increase of 209.67 % was noted, for the 60 decibel subjection, a noticeable increase of 1500 % was noted, and for the subjections of 70 and 80 decibels, with a 0 % recognition rate initially being noted, a new recognition of 55.83 % and 50 % was noted with a persistent 100 % reliability throughout the subjections. Looking at the data distribution to try and establish the true nature of the data obtained, it was found that for the 30-decibel subjection, for data within one and two standard deviations of the mean, that there was a distinct increase of lower limits and decrease of the upper limits – for the 2-word phrase. With this key increase and decrease on the respective limits being noticed, this proved to be highly favourable as this meant that the data for the optimized system was more highly centered and concentrated around the mean. The increases and decreases noted, for the first and second deviations, of a 2-word phrase were noted to be; {33.03 %, 15.83 %} and {75.96 %, 30.17 %} for 30 decibels, and {72.32 %, 6.97 %} and {197.97 %, 31.44 %} for 40 decibels. For the subjection of 50 decibels, the first and second standard deviation limits of the benchmark system were given by {-13.004 %, 77.58 %} and {-58.30 %, 120.88 %}, with increased in both cases to a 100 % consistency distribution. For the subjection of 60 decibels, the first and second standard deviation limits of the benchmark system were given by {-13.28 %, 25.78 %} and {-32.82 %, 45.32 %}, with increased in both cases to a 100 % consistency distribution. Due to the non-detection of the benchmark system, the limits for one and two standard deviations of 70 decibel subjections were {15.65 %, 92.69 %}, {-22.87 %, 131.21 %}. The subjection to the 80 decibel interference showed a constant 50 % recognition for the optimized system.

Taking a look at the 1-word phrase dataset analyzed. Results for this category can be viewed in Chapter 5.2.5, and Appendix E. Note that in the data that is presented, the first number within {"a ", "b "} refers to the lower limit – "a", with the second number corresponding to the upper limit of the data – "b". Results showed that as a whole, when subjected to decibel ranges of 30 decibels to 80 decibels, the benchmark system was able to detect phrases up until a limiting 60 decibels. When the benchmark system was subjected to interference of 70 decibels and 80 decibels, the benchmark system was incapable of detecting any form of phrase data. From a reliability perspective, for the benchmark system, a 100% detection rate was achieved for 30 decibels only.

This meant that although readings detected may not have been 100 % accurate, the benchmark system was still able to detect some form of a 1-word phrase. Reliability for the benchmark system decreased to 96 % at a 40-decibel subjection, 94 % for the 50 decibel subjection, 68 % for the 60-decibel subjection and persisted with a 0 % detection for subjections of 70 decibels and 80 decibels for the benchmark system. When the optimized system was considered, the system outperformed the benchmark system by having a 100 % reliability for subjections of 30-80 decibels of a 1-word phrase. Taking a look at the average operational benchmark data, this showed recognition readings of 69 %, 73.96 %, 80.85 %, 70.59 %, 0 % and 0 %, for 30-80 decibel subjections respectively. The optimization system, when compared to the benchmark system - proved that under a 30 decibel subjection, the average phrase recognition rate increased by a noticeable 44.93%. For the 40-decibel subjection, a noticeable increase of 35.21 % was noted, for the 50-decibel subjection, a noticeable increase of 23.69 % was noted, for the 60-decibel subjection, a noticeable increase of 41.66 % was noted, and for the subjections of 70 and 80 decibels, with a 0 % recognition rate initially being noted, a new recognition of 100% and 50 % was noted with a persistent 100 % reliability throughout the subjections. Looking at the data distribution to try and establish the true nature of the data obtained, it was found that for the 30-decibel subjection, for data within one and two standard deviations of the mean, that there was a distinct increase of lower limits and decrease of the upper limits – for the 1-word phrase. With this key increase and decrease on the respective limits being noticed, this proved to be highly favourable as this meant that the data for the optimized system was more highly centered and concentrated around the mean. The increases and decreases noted, for the first and second deviations, showed that - for the subjection of 30 decibels, the first and second standard deviation limits of the benchmark system were given by {22.52 %, 115.48 %} and {-23.97 %, 161.69 %}, with increased in both cases to a 100 % consistency distribution. For the subjection of 40 decibels, the first and second standard deviation limits of the benchmark system were given by {17.31 %, 130.61 %} and {-39.34 %, 187.26 %}, with increased in both cases to a 100 % consistency distribution. For the subjection of 50 decibels, the first and second standard deviation limits of the benchmark system were given by {38.66 %, 123.04 %} and {-3.53 %, 123.04 %}, with increased in both cases to a 100 % consistency distribution. For the subjection of 60 decibels, the first and second standard deviation limits of the benchmark system were given by {8.15 %, 133.03 %} and {-54.28 %, 195.46 %}, with increased in both cases to a 100 % consistency distribution. The subjection to the 70-80 decibel interference showed a constant 100 % and 50 % recognition for the optimized system.

For the mobile robot navigation system, the data for this system can be viewed in Section 5.3, the data presented the accuracy of the mobile navigation robot. Five different instances were tested. The instances included a 3 m, 5 m, 7 m, 10 m and 15 m range test. Each instance ran ten iterations of results to gauge appropriate accuracy results. Results showed, that for the tested instances, accuracy was limited from a minimum of 111.5 cm to a maximum of 285.5 cm.

6.1 Application of design

As recommended from research, it was recommended that once such application of the voice controlled mobile navigation robot was for the implemented into the maintenance failure process. It was established, that in a manufacturing environment, when a failure occurs, it can be a very time-consuming process to transfer equipment between the divisions of operations, maintenance and engineering in manufacturing environments. To gain a good understanding of the application that the proposed systems attempts to simulate, a graphic representation of interactions is presented in Figure 6-1. When a failure has occurred in a manufacturing environment, the control room detects this failure. When the nature of the failure is established, communication can be made with the mobile robot to proceed to a tools workshop, for instance. When the mobile robot navigates to the tool workshop, the workman has the ability to place the tools required by the mobile robot. The mobile robot then has to ability to navigate to the region of failure. The robot then waits for an operator to use the tools and equipment already present at the failure area to carry out maintenance.

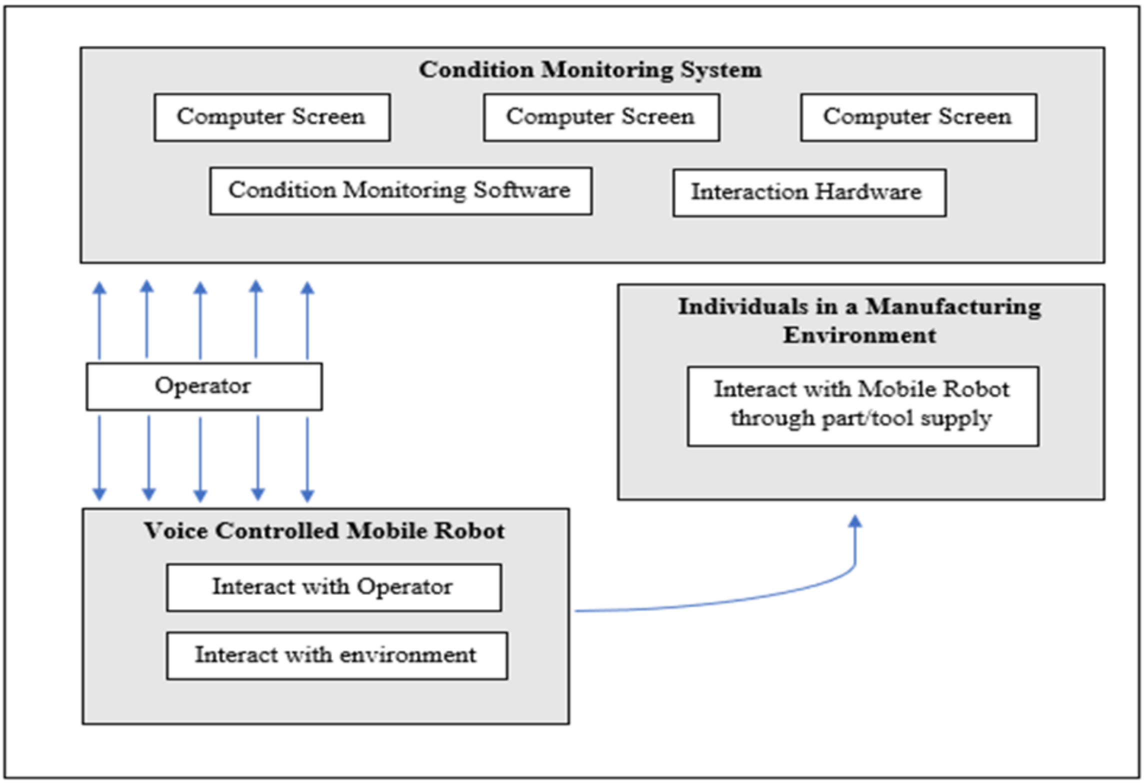


Figure 6-1: Proposed application by Eskom

Another application for disabled individuals in the configuration of Figure 6-1 also presents itself. Given that a physically disabled individual has a fully functional brain, though lack either hand/ leg functionality, they still possess the ability to identify and solve problems the way any other normal able-bodied individual is capable of doing. As presented in Figure 6-1, a disabled individual could perform simple condition monitoring tasks as an operator. When a failure is detected, the disabled individual can relay a voice command to a mobile robot concerning the parts required for repairing the system under failure. The mobile robot then proceeds to the location of the parts required as instructed, receives the parts required and proceeds to the region of failure – where an able-bodied individual carries out maintenance. This system reduces the workload of an able-bodied individual in a manufacturing environment, and introduces disabled individuals into manufacturing.

6.2 Chapter summary

Chapter six displayed the Discussion Chapter. This chapter outlined on pertinent information and results associated with every chapter of the dissertation. Implications of voice interface technology for the fourth industrial revolution were discussed in this chapter, noting that such technology is most likely to be used in advanced manufacturing environments. This chapter noted that the use of mobile navigation robots can facilitate the integration of voice interface technology in manufacturing operations. It was established that one area of application of the technology developed is when a maintenance failure occurs; the process to be followed in the case of such a failure to be the mobile robot design outlined.

7. CONCLUSION

This research was motivated by the need to develop a conceptualized methodological approach for optimization of voice interface technology, with proposed applications to advanced manufacturing environments. Voice interfaces technology was investigated to develop an operational benchmark – when subjected to simulated manufacturing conditions. The study established an optimization for the accuracy and reliability of voice interface systems – with a proposed method of application for manufacturing environments using a mobile robot design. The voice interface optimization system design focused on phrase accuracy and reliability; when subjected to common manufacturing noise conditions. The voice interface optimization system was subjected to a simulated manufacturing environment to create synthetic data, the data was empirically analysed and tested to formulate appropriate conclusions. The mobile robot design paired with the voice interface technology was developed to create a platform for implementation of voice interfaces into manufacturing environments. The mobile robot was subjected to path accuracy testing to establish the validity of its imbedded navigation system. Testing of the mobile robot design was implemented in a simulated manufacturing environment.

The major concerns presented by the research topic was; a lack of an implementation mean for manufacturing environments, high levels of language fluency requirements, low interpretability constraints, having a very high sensitivity to the environment, and a low adaptability to different languages. The first issue sought to address was to establish the state of voice interface technology from the perspective of an academic expert. As such, Dr. Alexandra Birch was approached. Dr. Birch is a Senior Research Fellow from the Institute for Language, Cognition and Computation (ILCC) in the School of Informatics at the University of Edinburgh. A machine learning workshop facilitated by Dr. Birch was attended at the University of KwaZulu-Natal, and following the workshop, a dissertation overview was presented to Dr. Birch, and an interview was carried out. Consultation with two industrial System Engineers sought to address the issue of lack of implementation of voice interfaces in manufacturing environments, literature was sought after to try and establish a means of improving on the issue of requirements of low interpretability constraints. Enrolment into online learning facilities was carried out to establish a link between voice interfaces and machinery – due to the proprietary nature of the information required. To gain an understanding of good automation development practices for industrial machines, an expert software developer working in an industrial environment was consulted. The design approach recommended was GAMP®5. From literature and Dr. Birch, an IoT platform and web-based server provided for a means to address the concerns of high levels of language fluency requirements and limitations placed by sensitivity to the environmental conditions.

The first objective of the dissertation was to research and establish a list of the essential areas of concern within manufacturing. To address this objective, research was conducted and six areas of concern were established within the manufacturing domain. The dissertation incorporated three limiting manufacturing factors into its design process, this was done to create a system that could be implemented outside the academic domain. The factors addressed included; the development of flexible systems being confined to the academic domain with less focus placed on industrial applications, the absence of a methodological structure for implementing flexible systems into the industrial domain, and the absence of due consideration being given to human integration of flexible systems. To address these concerns, the authors of the dissertation integrated these factors into the design methodology. Industrial experts were also consulted to establish a needs analysis and an appropriate means of dissertation progression.

The second objective of the dissertation was to research and develop a conceptualized methodological approach for integrating voice assistants into manufacturing environments. To address this issue, information was gained from industrial experts, literature, online courses and academic experts, to develop the methodological structure for a good automated computerized mechatronic system. The developed system structure integrated googles “OK Google” with a mobile navigation system design to carry out basic navigation and tool/part transportation to designated areas. Objective three of the dissertation was to generate synthetic voice-assistant data using a simulated manufacturing environment. To address objective three, a variety of systems were developed and presented through methods shown in Chapter 3 and Chapter 4. Synthetic data was generated to establish a performance benchmark for voice interfaces when subjected to manufacturing interference, simulation of the system was carried out in the Advanced Manufacturing Laboratory at the University of KwaZulu-Natal. Objective four of the dissertation was to research and optimize voice-assistant technology, conforming to the fourth industrial revolution. In an attempt to address objective 4, additional research was conducted to improve on the voice interface performance benchmark data. This was carried out to allow for integration of voice interfaces into the manufacturing domain. The optimization design encompassed web-based servers, and cloud-based Internet platforms (IoT platforms) – in an attempt to conform to the fourth industrial revolution, and to increase the reputability of voice technology in environments previously deemed unfit for implementation. Objective five of the dissertation was to analyze the performance of the optimized system and evaluate applicability of the optimized system. Objective five was addressed through the designed data extraction systems. The data extracted from the voice systems were subjected to a statistical analysis to validate their performance capability within a manufacturing environment. A performance programme was developed and the voice data was subjected to a probability density analysis. Objective six of the dissertation was to research, design and implement a suitable mechatronic system integrating voice interfaces with mobile robots. The designed system was developed to include the development of electronic hardware, the programming of machine control algorithms, and the creation of a suitable interface for human interaction. Objective six was addressed through the culmination of objectives 1-5, with the integration of these objectives being integral.

Through the fulfilment of the objectives listed from objective one to objective six, this answered the research question: how can a mechatronic system be designed for a manufacturing environment that integrates voice interface technology and mobile robotics? A mechatronic system was designed in this dissertation that had the ability to be used in a manufacturing environment, with the integration of voice interface technology with a mobile robot design.

7.1 Recommendations

Though extensive research was conducted into the voice interface systems mentioned in previous chapters, it is recommended that for future work, the systems designed should be subjected to a larger sample data set of voice readings. Additionally, voice variations should be included into the operational benchmark tests to allow for data variances. This would allow for a larger analysis spectrum. The key concern for any navigation system is its onboard compass drive system. As such, due to the sensitive nature of compass systems to localized magnetic fields, it is recommended that for increased navigation accuracy, a structured calibration technique be developed to negate the effect of additional magnetic fields on a compass drive system.

7.2 Chapter summary

Chapter seven displayed the Conclusion Chapter. This chapter very closely ties in to, and associates to the objectives of the Introduction Chapter. This chapter outlined the work performed and the research developed to create a voice interface technology that can be integrated into manufacturing environments; all in line with the objectives of the dissertation.

8. REFERENCES

[1] A. G. Frank, L. S. Dalenogare, and N. F. Ayala, "Industry 4.0 technologies: Implementation patterns in manufacturing companies," *International Journal of Production Economics,* vol. 210, pp. 15-26, 2019/04/01/ 2019, doi: https://doi.org/10.1016/j.ijpe.2019.01.004.

[2] B. T. Andresen AL, Nielsen K. , "Reconfigurable Manufacturing on Multiple Levels: Literature Review and Research Directons," *Springer*, Advances in production management systems: innovative production management towards sustainable growth., pp. pp. 266–273, 2015.

[3] A.-L. Andersen, K. Nielsen, and T. D. Brunoe, "Prerequisites and Barriers for the Development of Reconfigurable Manufacturing Systems for High Speed Ramp-up," *Procedia CIRP,* vol. 51, pp. 7-12, 2016/01/01/ 2016, doi: https://doi.org/10.1016/j.procir.2016.05.043.

[4] A.-L. Andersen, T. D. Brunoe, K. Nielsen, and C. Rösiö, "Towards a generic design method for reconfigurable manufacturing systems : Analysis and synthesis of current design methods and evaluation of supportive tools," (in eng), *Journal of manufacturing systems,* article vol. 42, pp. 179-195, 2017 2017, doi: 10.1016/j.jmsy.2016.11.006.

[5] D. T. Matt, E. Rauch, and P. J. P. C. Dallasega, "Mini-factory–a learning factory concept for students and small and medium sized enterprises," vol. 17, pp. 178-183, 2014.

[6] B. Wang, "The Future of Manufacturing: A New Perspective," *Engineering*, vol. 4, no. 5, pp. 722-728, 2018, doi: 10.1016/j.eng.2018.07.020.

[7] M. S. Kandhari, F. Zulkemine, and H. Isah, "A Voice Controlled E-Commerce Web Application," in *2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, 1-3 Nov. 2018 2018, pp. 118-124, doi: 10.1109/IEMCON.2018.8614771.

[8] Y. Koren *et al.*, "Reconfigurable Manufacturing Systems," *CIRP Annals*, vol. 48, no. 2, pp. 527-540, 1999/01/01/ 1999, doi: https://doi.org/10.1016/S0007-8506(07)63232-6.

[9] J. Padayachee, "Development of a Modular Reconfigurable Machine for Reconfigurable Manufacturing Systems," 2010, pp. 1-13.

[10] M. Bortolini, F. G. Galizia, and C. Mora, "Reconfigurable manufacturing systems: Literature review and research trend," *Journal of Manufacturing Systems*, vol. 49, pp. 93-106, 2018/10/01/ 2018, doi: https://doi.org/10.1016/j.jmsy.2018.09.005.

[11] Y. Koren and M. Shpitalni, "Design of reconfigurable manufacturing systems," *Journal of Manufacturing Systems*, vol. 29, no. 4, pp. 130-141, 2010/10/01/ 2010, doi: https://doi.org/10.1016/j.jmsy.2011.01.001.

[12] A. V. Kapitanov, "Manufacturing System Flexibility Control," *Procedia Engineering,* vol. 206, pp. 1470-1475, 2017/01/01/ 2017, doi: https://doi.org/10.1016/j.proeng.2017.10.663.

[13] O. A. Makinde, K. Mpofu, and A. P. I. Popoola, "Review of the Status of Reconfigurable Manufacturing Systems (RMS) Application in South Africa Mining Machinery Industries," *Procedia CIRP*, vol. 17, pp. 136-141, 2014/01/01/ 2014, doi: https://doi.org/10.1016/j.procir.2014.02.035.

[14] O. Heimann and J. Krüger, "Affordance Based Approach to Automatic Program Generation for Industrial Robots in Manufacturing," *Procedia CIRP*, vol. 76, pp. 133-137, 2018/01/01/ 2018, doi: https://doi.org/10.1016/j.procir.2018.01.033.

[15] J. Ogbemhe, K. Mpofu, and N. S. Tlale, "Achieving Sustainability in Manufacturing Using Robotic Methodologies," *Procedia Manufacturing*, vol. 8, pp. 440-446, 2017, doi: 10.1016/j.promfg.2017.02.056.

[16] P. Mell and T. Grance, "The NIST definition of cloud computing," 2011, pp. 1-7, doi: https://doi.org/10.6028/NIST.SP.800-145

[17] T. Qu, S. P. Lei, Z. Z. Wang, D. X. Nie, X. Chen, and G. Q. Huang, "IoT-based real-time production logistics synchronization system under smart cloud manufacturing," *The International Journal of Advanced Manufacturing Technology*, vol. 84, no. 1, pp. 147-164, 2016/04/01 2016, doi: 10.1007/s00170-015-7220-1.

[18] D. S. Don Jones, "The Definitive Guide to Cloud Computing," 2010.

[19] X. Xu, "From cloud computing to cloud manufacturing, Robotics and Computer-Integrated Manufacturing," vol. 28, no. 1, pp. 75-86, 2012, doi: 10.1016/j.rcim.2011.07.002.

[20] M. G. Azam, *Application of cloud computing in library management: innovation, opportunities and challenges*. 2019, pp. 50-57, doi: 10.5281/zenodo.2536637

[21] R. P. Lalit Mohan, Sanjeev Bisht and Janmejay Pant, "A Comparative Study on SaaS, PaaS and IaaS Cloud Delivery Models," *International Journal on Emerging Technologies*, vol. 8, pp. 158-160, 2017.

[22] V. Bhavana, *Various Techniques in Deployment of Application in Cloud – Public Cloud*, 2014, pp. 393-399.

[23] P. Wang, R. X. Gao, and Z. Fan, "Cloud Computing for Cloud Manufacturing: Benefits and Limitations," *Journal of Manufacturing Science and Engineering*, vol. 137, no. 4, 2015, doi: 10.1115/1.4030209.

[24] X. Wang and X. Xu, *Virtualise manufacturing capabilities in the cloud: Requirements, architecture and implementation*. 2014, pp. 348-368.

[25] S. S. Tambe, "WIRELESS TECHNOLOGY IN NETWORKS," *International Journal of Scientific and Research Publications*, vol. 5, no. 7, 2015.

[26] J. Kaur, R. Kaur, and M. Kaur, "Bluetooth Technology," *International Journal Of Engineering And Computer Science*, 2016, pp. 16001-16003, doi: 10.18535/ijecs/v5i3.22.

[27] A. Ahmadi, A. Sodhro, C. Cherifi, V. Cheutet, and Y. Ouzrout, "Evolution of 3C Cyber-Physical Systems Architecture for Industry 4.0: Proceedings of SOHOMA 2018," 2019, pp. 448-459.

[28] M. M. Ahmadzai Ahmadi, Chantal Cherifi, Vincent Cheutet, Yacine Ouzrout, "Wireless Connectivity of CPS for Smart Manufacturing:A survey," *12th International Conference on Software, Knowledge, Intelligent Management and Applications (SKIMA 2018)*, 2018, pp. 1-8, doi: 10.1109/SKIMA.2018.8631535..

[29] M. Bilac, M. Chamoux, and A. Lim, "Gaze and filled pause detection for smooth human-robot conversations," in *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*, 15-17 Nov. 2017 2017, pp. 297-304, doi: 10.1109/HUMANOIDS.2017.8246889.

[30] G. Mahl, "Gestures and body movements in interviews," in *Explorations in Nonverbal and Vocal Behavior*: Routledge, 2014, pp. 23-90.

[31] M. E. Fabrizio Ghigi, M. Ines Torres, Sungjin Lee, "Incremental Dialog Processing in a Task-Oriented Dialog," *Language Technologies Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania*, 2014, pp. 309-312.

[32] M. A. Iqbal, S. K. Asrafuzzaman, M. M. Arifin, and S. K. A. Hossain, "Smart home appliance control system for physically disabled people using kinect and X10," in *2016 5th International Conference on Informatics, Electronics and Vision (ICIEV)*, 13-14 May 2016 2016, pp. 891-896, doi: 10.1109/ICIEV.2016.7760129.

[33] N. R. Council, *Technology and Assessment: Thinking Ahead: Proceedings from a Workshop*. Washington, DC: The National Academies Press (in English), 2002, p. 104.

[34] F. J. D. Nitin Indurkhya, "Handbook Of Natural Language Processing, Second Edition Chapman & Hall Crc Machine Learning & Pattern Recognition," 2010.

[35] E. Hancock, "Pattern Recognition," *Elsevier*, 2019.

[36] I. The MathWorks. "Image Recognition." MathWorks. https://www.mathworks.com/discovery/image-recognition.html (accessed).

[37] S. H. L. H. I. Kim, and Y. M. Ro, "Investigating cascaded face quality assessment for practical face recognition system," *IEEE Int'l Symposium on Multimedia (ISM),* 2014, pp. 399-400, doi: 10.1109/ISM.2014.30.

[38] H. Kim, S. H. Lee, and Y. M. Ro, "Face image assessment learned with objective and relative face image qualities for improved face recognition," in *2015 IEEE International Conference on Image Processing (ICIP)*, 27-30 Sept. 2015 2015, pp. 4027-4031, doi: 10.1109/ICIP.2015.7351562.

[39] J. Chen, Y. Deng, G. Bai, and G. Su, "Face Image Quality Assessment Based on Learning to Rank," *IEEE Signal Processing Letters*, vol. 22, no. 1, pp. 90-94, 2015, doi: 10.1109/LSP.2014.2347419.

[40] S. Elprama, C. Jewell, A. Jacobs, I. El Makrini, and B. Vanderborght, *Attitudes of Factory Workers towards Industrial and Collaborative Robots*. 2017, pp. 113-114.

[41] A. Roncone, O. Mangin, and B. Scassellati, "Transparent role assignment and task allocation in human robot collaboration," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 29 May-3 June 2017 2017, pp. 1014-1021, doi: 10.1109/ICRA.2017.7989122.

[42] D. Wurhofer, T. Meneweger, V. Fuchsberger, and M. Tscheligi, "Deploying Robots in a Production Environment: A Study on Temporal Transitions of Workers' Experiences," in *Human-Computer Interaction – INTERACT 2015*, Cham, J. Abascal, S. Barbosa, M. Fetter, T. Gross, P. Palanque, and M. Winckler, Eds., 2015// 2015: Springer International Publishing, pp. 203-220.

[43] I. E. Makrini *et al.*, "Working with Walt: How a Cobot Was Developed and Inserted on an Auto Assembly Line," *IEEE Robotics & Automation Magazine*, vol. 25, no. 2, pp. 51-58, 2018, doi: 10.1109/MRA.2018.2815947.

[44] I. E. Makrini, C. Rodriguez-Guerrero, D. Lefeber, and B. Vanderborght, "The Variable Boundary Layer Sliding Mode Control: A Safe and Performant Control for Compliant Joint Manipulators," *IEEE Robotics and Automation Letters*, vol. 2, no. 1, pp. 187-192, 2017, doi: 10.1109/LRA.2016.2587059.

[45] F. Gosselin, F. Ferlay, and A. Janot, *Development of a New Backdrivable Actuator for Haptic Interfaces and Collaborative Robots*. 2016, pp. 1-26, https://doi.org/10.3390/act5020017

[46] D. A. Birch, in *Machine Learning Workshop: Voice technology in manufacturing systems* S. Pather, Ed., ed. University of KwaZulu-Natal, 2019, pp. 1-2.

[47] V. Magan, "Industrial Engineer Interview: Implication of voice technology in industry," in *Eskom System Engineer*, S. Pather, Ed., ed. University of KwaZulu-Natal, 2019.

[48] F. Mathe, "Industrial Engineer Interview: Implication of voice technology in industry," in *Eskom System Engineer*, S. Pather, Ed., ed. University of KwaZulu-Natal, 2019.

[49] C. Plagiannos. What is GAMP®5 and how do I use it effectively? [Online] Available: https://blog.montrium.com/experts/what-is-gamp5-and-how-do-i-use-it-effectively

[50] J. Reddy, "Industrial Software Expert Interview: Implication of voice technology in industry," in *LabWare Africa (Pty) Ltd*, S. Pather, Ed., ed. University of KwaZulu-Natal, 2019.

[51] R. E. Berg. "Sound." https://www.britannica.com/science/sound-physics (accessed October 25, 2019, 2019).

[52] "TMSOFT." https://play.google.com/store/apps/details?id=com.tmsoft.whitenoise.generator.noise (accessed 17 April, 2017).

APPENDIX A: CERTIFICATES

Certificate of Completion

This is to certify that Shane Pather successfully completed 5.5 hours of Exploring AWS IoT online course on Aug. 13, 2019

Stephen Borsay

Stephen Borsay, Instructor

&

Udemy

#BeAble

Certificate no: UC-MLWH4QV2
Certificate url: ude.my/UC-MLWH4QV2

Figure 9-1: Certificate Obtained

Certificate of Completion

This is to certify that Shane Pather successfully completed 2 hours of AWS IoT: The Hobbyists Guide to Home Automation online course on

April 29, 2019

Nathan Glover

Nathan Glover, Instructor

&

Udemy

#BeAble

Certificate no: UC-MPEC5CU7
Certificate url: ude.my/UC-MPEC5CU7

Figure 9-2: Certificate Obtained

Certificate of Completion

This is to certify that Shane Pather successfuly completed 2 hours of Build your own GPS tracking system-Raspberry Pi Zero W 2019 online course on April 29, 2019

comfiny ., Instructor

& Udemy

#BeAble

Certificate no: UC-TSMRFH5M
Certificate url: ude.my/UC-TSMRFH5M

Figure 9-3: Certificate Obtained

APPENDIX B: SYSTEM DESIGN

10.1 Decibel range metre and Interference generator

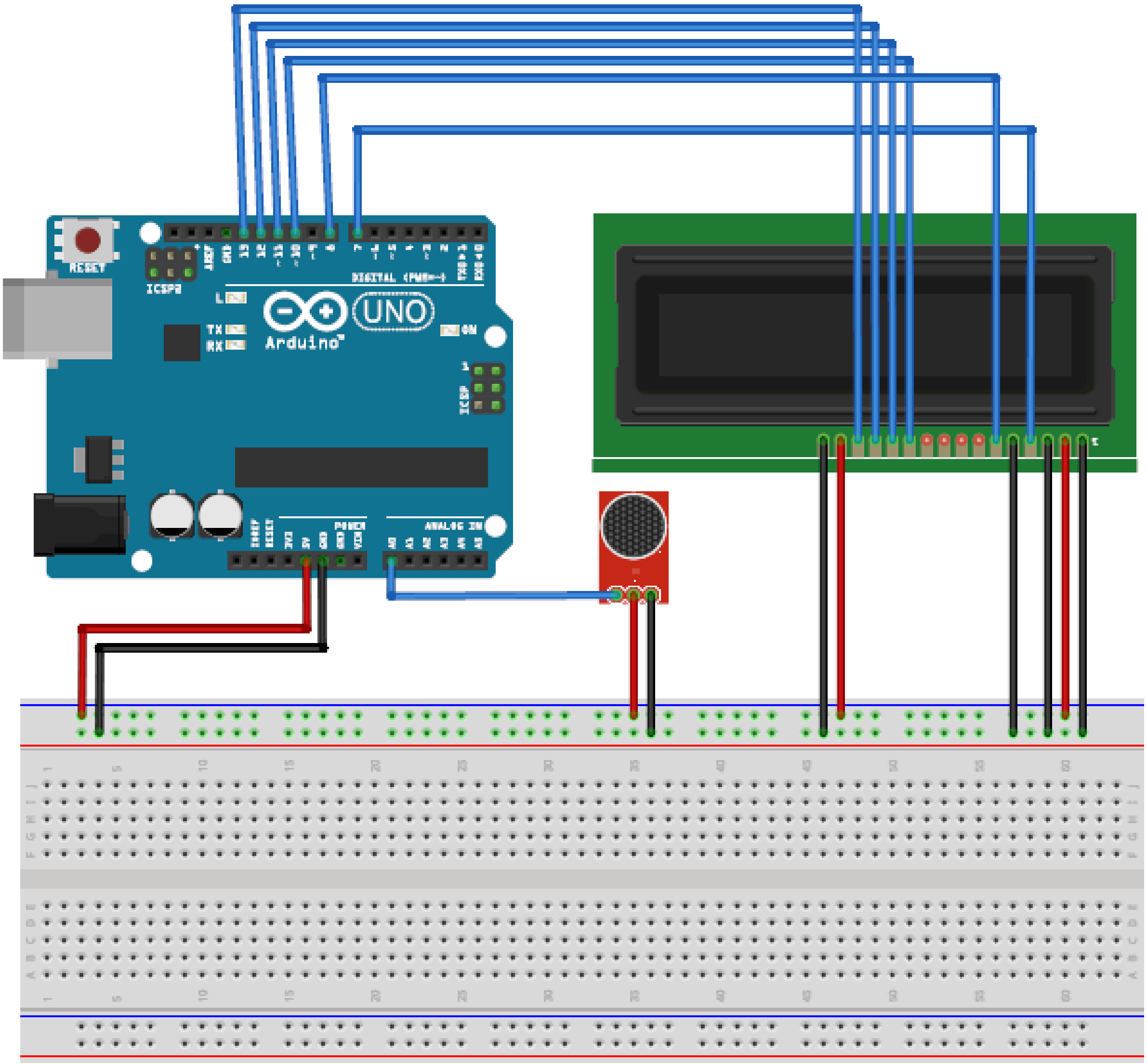


Figure 10-1: Decibel range metre

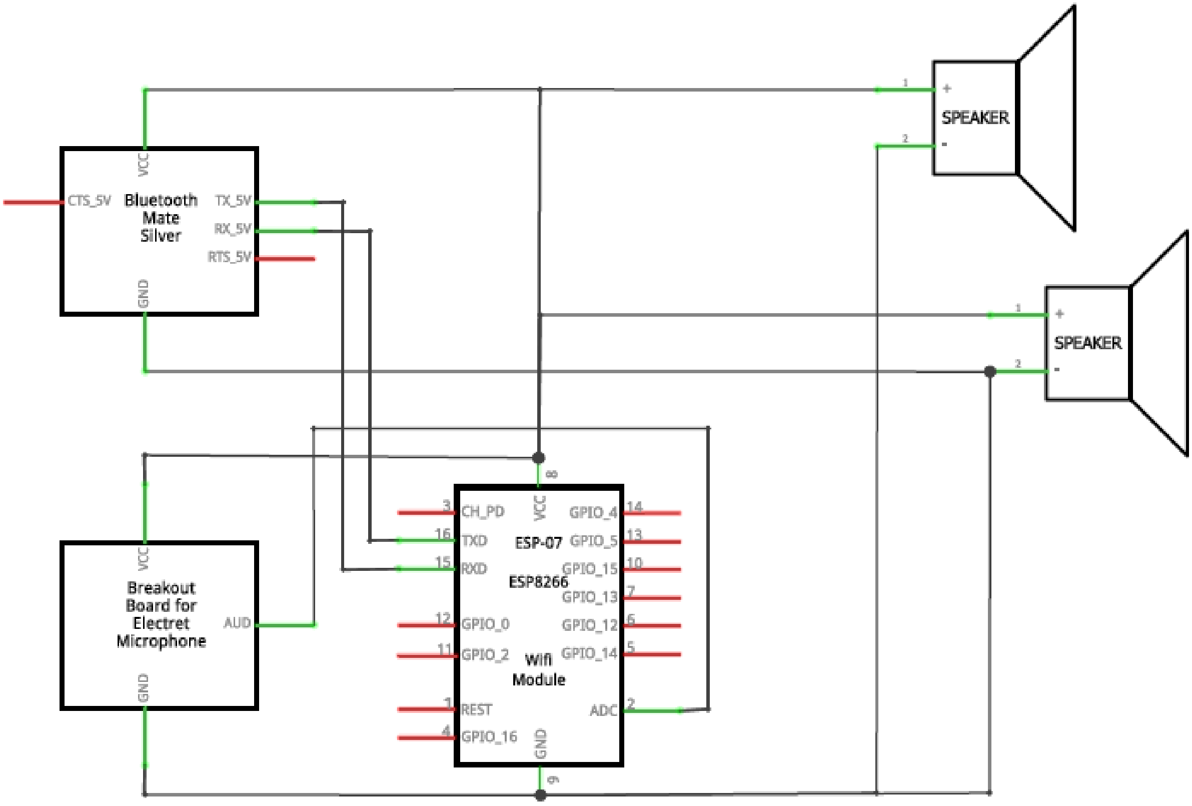


Figure 10-2: Interference generator

10.2 Python accuracy analysis code

```
def prr(ref, hyp ,debug=False):

r = ref.split()

h = hyp.split()

costs = [[0 for inner in range(len(h)+1)] for outer in range(len(r)+1)]

backtrace = [[0 for inner in range(len(h)+1)] for outer in range(len(r)+1)]

OP_OK = 0

OP_SUB = 1

OP_INS = 2

OP_DEL = 3

DEL_PENALTY=1 # Tact
```

```
INS_PENALTY=1 # Tact
SUB_PENALTY=1 # Tact
for i in range(1, len(r)+1):
costs[i][0] = DEL_PENALTY*i
backtrace[i][0] = OP_DEL
for j in range(1, len(h) + 1):
costs[0][j] = INS_PENALTY * j
backtrace[0][j] = OP_INS
for i in range(1, len(r)+1):
for j in range(1, len(h)+1):
if r[i-1] == h[j-1]:
costs[i][j] = costs[i-1][j-1]
backtrace[i][j] = OP_OK
else:
substitutionCost = costs[i-1][j-1] + SUB_PENALTY # penalty is always 1
insertionCost    = costs[i][j-1] + INS_PENALTY   # penalty is always 1
deletionCost     = costs[i-1][j] + DEL_PENALTY   # penalty is always 1
costs[i][j] = min(substitutionCost, insertionCost, deletionCost)
if costs[i][j] == substitutionCost:
backtrace[i][j] = OP_SUB
elif costs[i][j] == insertionCost:
backtrace[i][j] = OP_INS
else:
```

```
backtrace[i][j] = OP_DEL
i = len(r)
j = len(h)
numSub = 0
numDel = 0
numIns = 0
numCor = 0
if debug:
print("OP\tREF\tHYP")
lines = []
while i > 0 or j > 0:
if backtrace[i][j] == OP_OK:
numCor += 1
i-=1
j-=1
if debug:
lines.append("OK\t" + r[i]+"\t"+h[j])
elif backtrace[i][j] == OP_SUB:
numSub +=1
i-=1
j-=1
if debug:
lines.append("SUB\t" + r[i]+"\t"+h[j])
```

```
elif backtrace[i][j] == OP_INS:
numIns += 1
j-=1
if debug:
lines.append("INS\t" + "****" + "\t" + h[j])
elif backtrace[i][j] == OP_DEL:
numDel += 1
i-=1
if debug:
lines.append("DEL\t" + r[i]+"\t"+"****")
if debug:
lines = reversed(lines)
for line in lines:
print(line)
print("Ncor " + str(numCor))
print("Nsub " + str(numSub))
print("Ndel " + str(numDel))
print("Nins " + str(numIns))
return (numSub + numDel + numIns) / (float) (len(r))
wer_result = round( (numSub + numDel + numIns) / (float) (len(r)), 3)
return {PRR':prr_result, 'Cor':numCor, 'Sub':numSub, 'Ins':numIns, 'Del':numDel
# Run:
ref=' '// inserted as required
```

```
hyp="// inserted as required

prr(ref, hyp ,debug=True)
```

10.3 Calibration equipment

Figure 10-4: Calibration Control Tower

Figure 10-3: National Instruments Precision Mic

Figure 10-6: Calibration Interference Generator

Figure 10-5: VRTC Testing Facility

Figure 10-7: CompactDaq

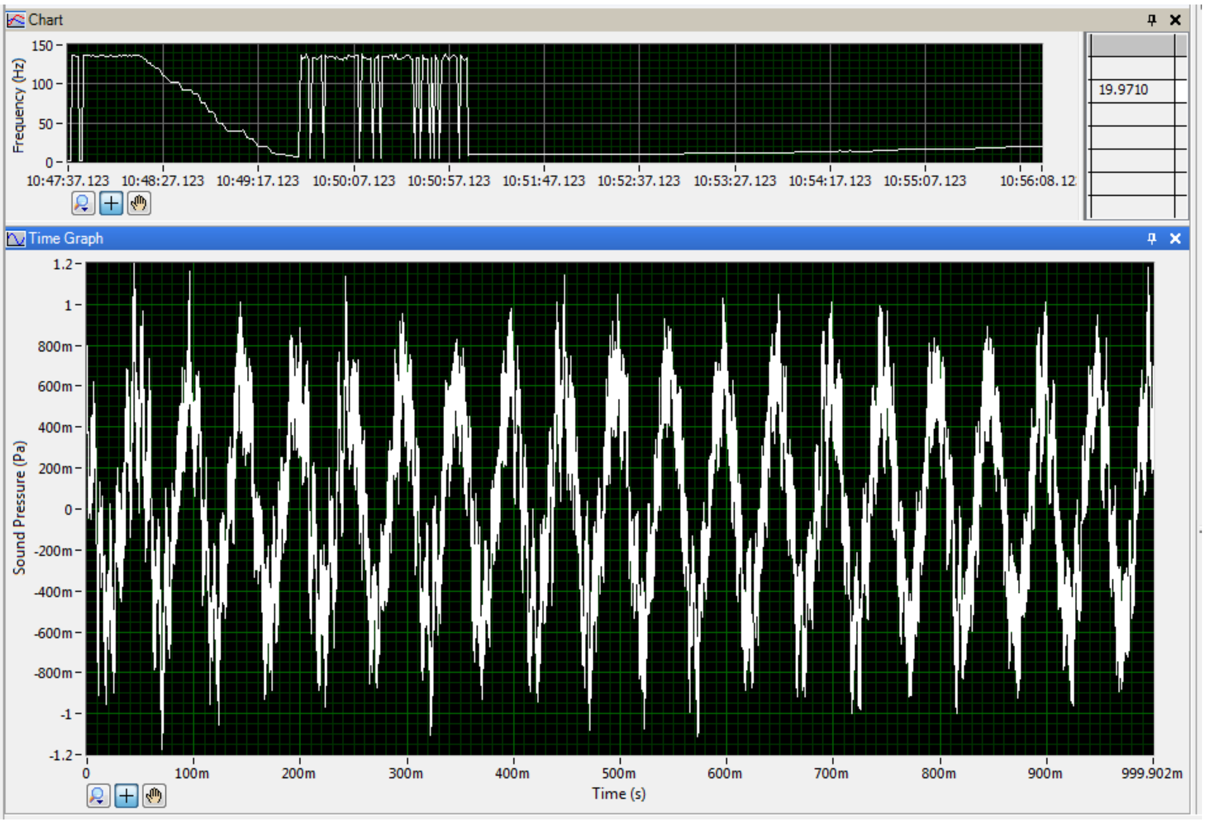


Figure 10-8: Calibration Test data

10.4 Calibration procedure LabVIEW

1. On the GUI interface of LabVIEW SignalExpress, select Add Step.
2. Click on Acquire signals
3. Select DAQ mx Acquire
4. Select Analog input
5. Select Sound Pressure
6. Choose appropriate channels
7. Establish required sensitivity
8. Select Data view
9. Drag required signals to grid

10.5 Final configurations and designs

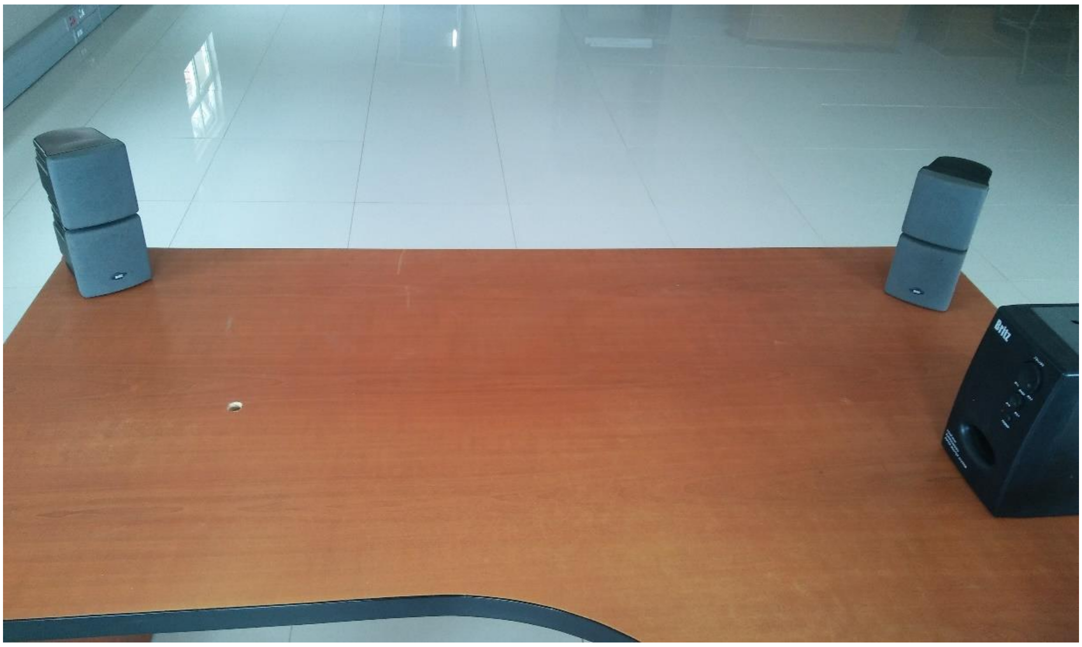


Figure 10-9: Interference configuration

Figure 10-10: Mobile device and microphone used

Figure 10-11: Interference cancelling device

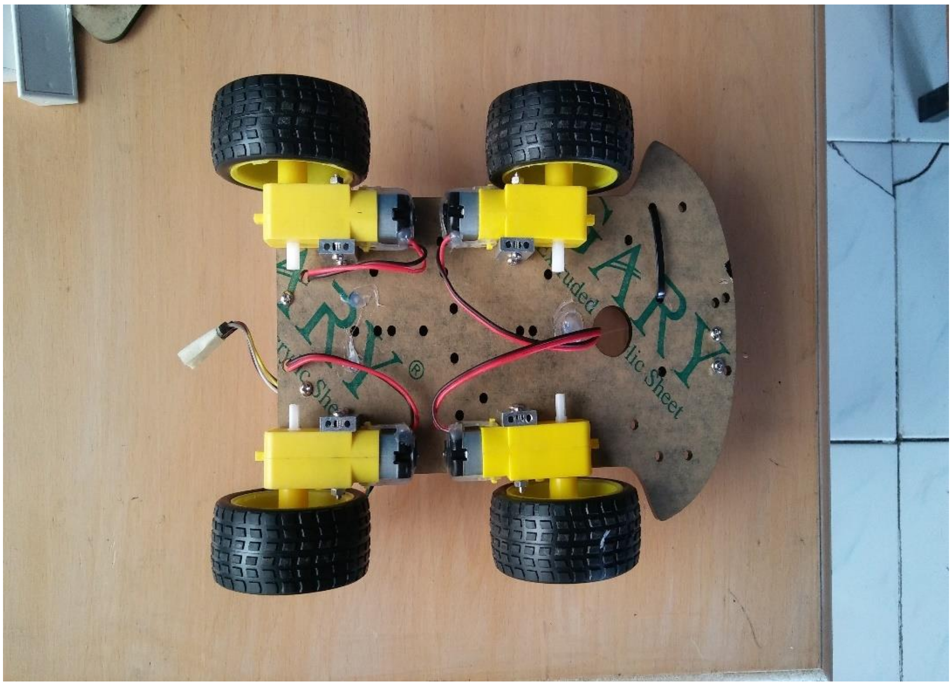


Figure 10-12: Mobile robot motor layout

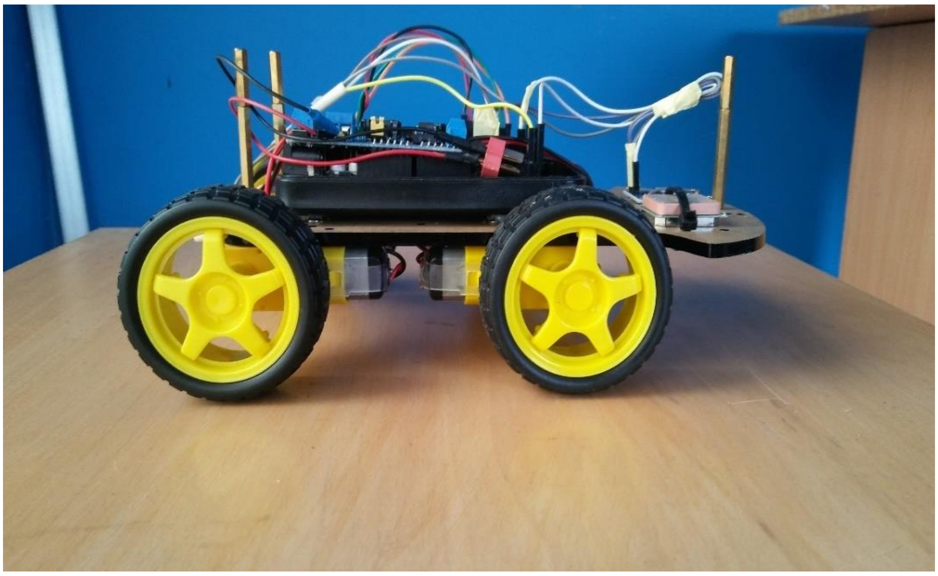


Figure 10-13: Navigation system front view

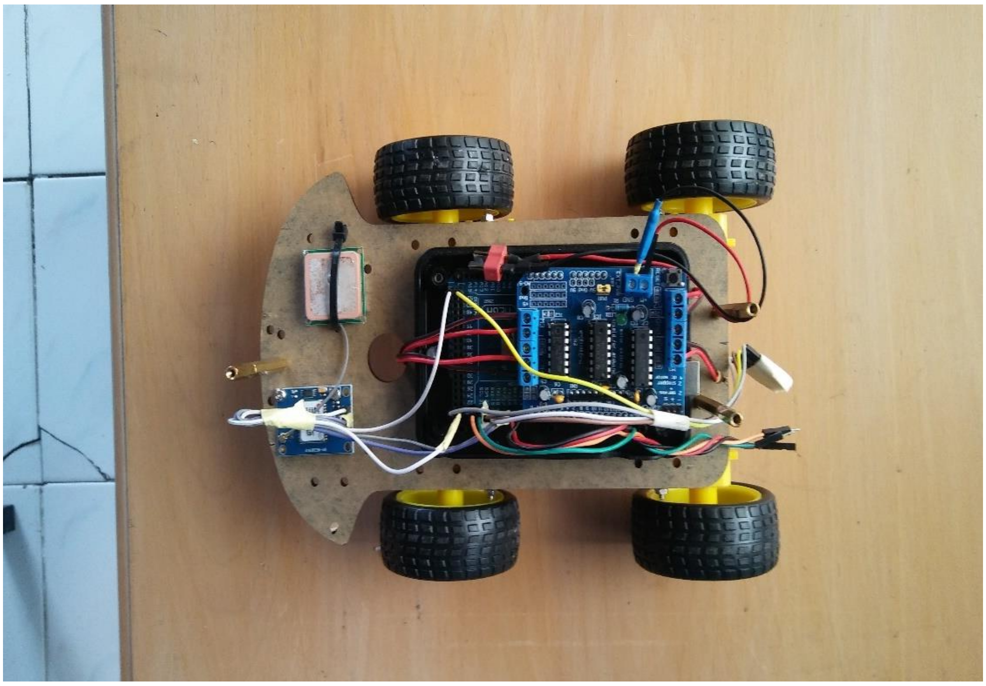


Figure 10-14: Navigation system top view

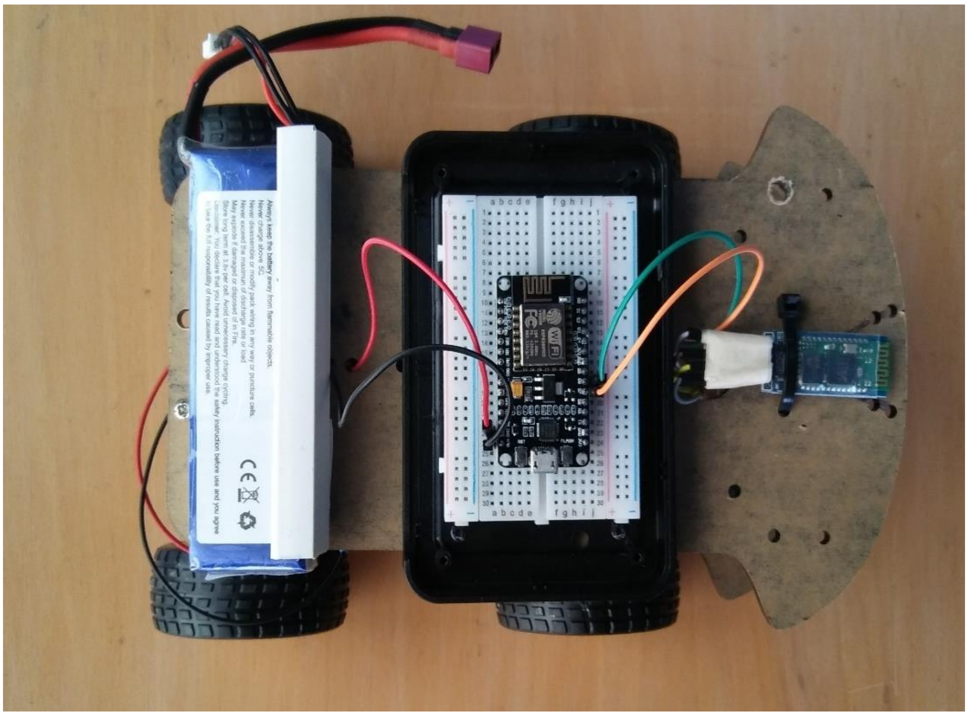


Figure 10-15: IoT platform integration

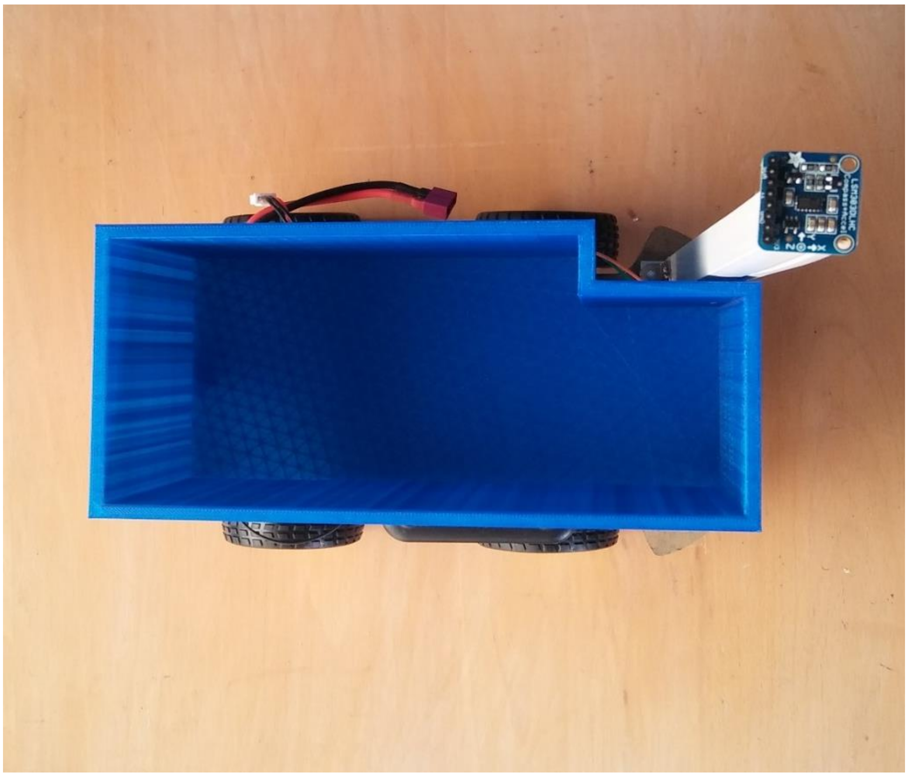


Figure 10-16: Tool handling and compass configuration

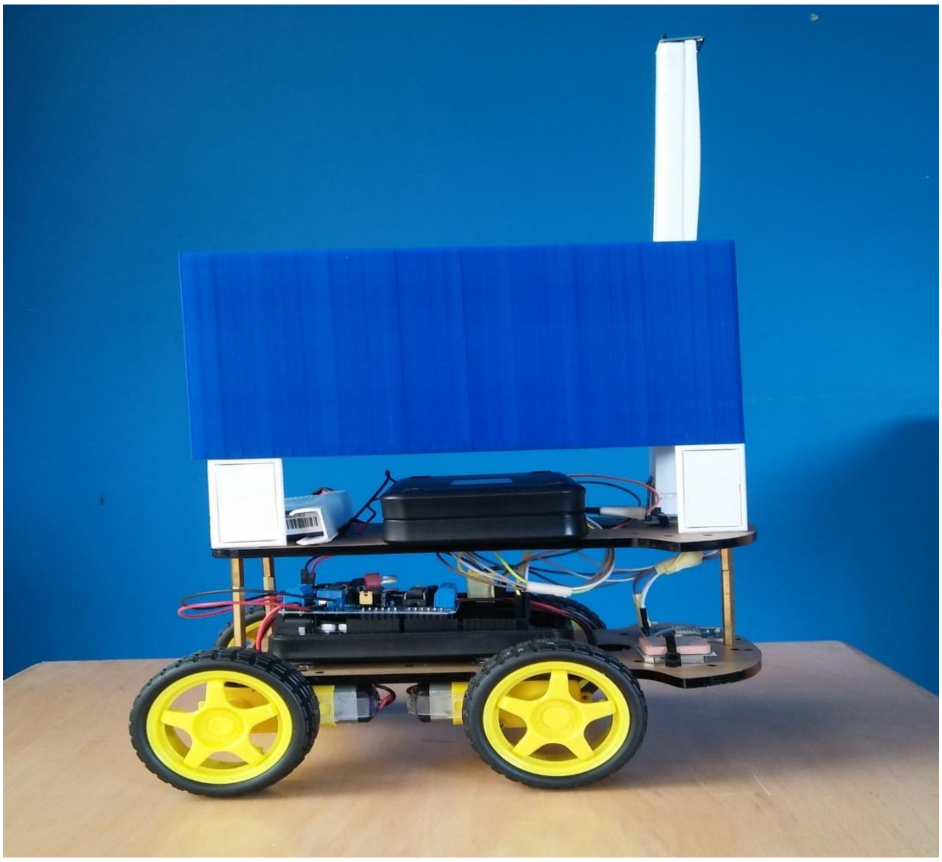


Figure 10-17: Full final design

Figure 10-18: Eskom readings 1

Figure 10-19: Eskom reading 2

APPENDIX C: CLOUD-BASED IOT PLATFORM AND WEB-BASED SERVER INFORMATION

11.1 Instructions for IoT platform:

1. Download IoT plaform of choice to desired facilitating device – Blynk was chosen for the dissertation.
2. Create a Blynk Account. Follwing the download of the application. The user will be promted to create an account. This account created will be used to save projects, with the ability to access these projects from multiple devices. It also serves as a security measure for the application.
3. Click on the new project icon to begin a new project.
4. An option to select the project name, microcontroller used and connection type will be displayed.
5. Click create project
6. A very important piece of information provided is the "Auth Token". This token is a unique identifier whch is required for the connection of specified devices to a smartphone. Every project created has a unique "Auth Token". This token is sent to the user email address when a project is created.
7. Specify the web-server being used, the hardware being used.
8. Identify, under the URL section, the adresses of the blink cloud. The IP address of the cloud is specific to the location on use, it should contain your Auth token and hardware pins in use
9. If the user wants to create a local server – for better security, stability, lower latency and increased privacy, the user should follow the following steps;
    - Access blynk server project – Github could be used for this
    - Make sure you are using Java 11 or above
    - Download the latest version of the server
    - Access the command terminal
    - Proceed to the location of the downloaded server
    - Execute : `java -jar server-0.41.11.jar -dataFolder /path`
    - A log and server data folder will be created with a message validating the local server in operation
    - Establish IP address of user pc
    - Lastly, connect the local server to the IoT application and make sure that all devices are connected to the same network.
10. Select method as "GET", and content type as application.

11.2 Instructions for Web-based server:

1. Following from the “Instructions for IoT platform”, sign up to the IFTTT Web-based platform. An option is provided for the sign up with a google account.
2. Click on My Applets and then on new Applet
3. Click on “This”
4. Search for the voice interface technology being use, in the case of the dissertation, Google assistant.
5. Click “Connect”, and following this, IFTTT will ask for permission to use the your google account to add voice commands for google assistant.
6. Allow for the access of the account
7. You will be presented with four options for optimization
    - Optimization through confinement of the command to a specifically simple phrase
    - Optimization through confinement of the command to say a specific phrase with a number
    - Optimization through confinement of the command to say a phrase with a text ingredient. Example: “Post a tweet saying ‘New high score.’ ” The $ symbol is to specify where you'll say the text ingredient, and
    - Optimization through confinement of a command to say a phrase with both a number and a text ingredient
8. Select the optimization approach
9. Optimization process of saying a simple phrase was chose for the dissertation
10. The desired phrase is prompted for with a facility to provide a variety of variations to the phrase accounting for the areas of concern presented in Chapter 3.
11. Implement a command to trigger if the desired response is achieved
12. Click create trigger
13. Click on “That”
14. Chose an action service in response being received by the google assistant
15. Select “Webhooks” and “Connect”
16. Select the option of making a “Web-based request” with the IoT platform
17. Identify the URL being used. This note is very important as the URL being entered should be identical to that mentione in point “8” of the “Instructions for IoT platform” , this is the method of liking the two platforms.
18. Select the “Put” method and the content type as application.
19. Select under “Body” either [“0”] or [“1”], depending of the state required.
20. Click “Create action”, and “Finish”

APPENDIX D: SOFTWARE DESIGN

Main Code:

```
#include <AFMotor.h>            //  Adafruit Motor Shield Library

#include "Wire.h"               // LSM303 compass

#include "I2Cdev.h"             // I2C Library (for compass)

#include "LSM303.h"

#include <TinyGPS++.h>          // For GPS Positioning

-----------------------------------------------------------------------------------------------------------------
--

// GPS Variables

int GPSCourse;

int NumberSATS;

TinyGPSPlus gps;

-----------------------------------------------------------------------------------------------------------------
-

AF_DCMotor motor1(1, MOTOR12_64KHZ);

AF_DCMotor motor2(2, MOTOR12_64KHZ);

AF_DCMotor motor3(3, MOTOR12_64KHZ);

AF_DCMotor motor4(4, MOTOR12_64KHZ);

int TSpeed = 175;               // motor speed for turning

int mtrSpd = 250;               // motor speed – back and forward


-----------------------------------------------------------------------------------------------------------------
----

// Compass Variables

LSM303 compass;

int16_t mx, my, mz;             // variables to store x,y,z axis from compass LSM303

int desiredHeading;

int compassHeading;

int compass_dev = 7;            // the amount of deviation that is allowed
```

```
int HeadingA;                  // variable to store compass heading

int HeadingB;                  // variable to store compass heading in Opposite direction

int pass = 0;                  // variable to store which pass the robot is on


---------------------------------------------------------------------------------------------------------------------
-------

// Bluetooth Variables & Setup


String str;                    // raw string received from android to arduino

int blueToothVal;              // stores the last value sent over via bluetooth

int bt_Pin = 30;

---------------------------------------------------------------------------------------------------------------------
-------

// GPS Locations

unsigned long Distance_To_Home;      // variable for storing the distance to destination

int ac =0;                           // GPS array counter

int wpCount = 0;                     // GPS waypoint counter

double Home_LATarray[50];            // variable for storing the destination Latitude - Only
Programmed for 5 waypoint

double Home_LONarray[50];            // variable for storing the destination Longitude - up
to 50 waypoints

int increment = 0;

void setup()

{

Serial.begin(115200);                // Serial 0 is for communication with the computer

Serial1.begin(9600);                 // Serial 1 is for Bluetooth communication -

Serial2.begin(9600);                 // Serial 2 is for GPS communication at 9600 baud

                                     // Compass

Wire.begin();                        // Join I2C bus used for the LSM303 compass

compass.begin();                     // initialize the compass (LSM303)

compass.setRange(LSM303_RANGE_1_3GA);                  // Set measurement range
```

```
compass.setMeasurementMode(LSM303_CONTINOUS);            // Set measurement mode
compass.setDataRate(LSM303_DATARATE_30HZ);               // Set data rate
compass.setSamples(LSM303_SAMPLES_8);                    // Set number of samples averaged
compass.setOffset(-219,-421);                            // Set calibration offset
Startup();                                    // Run the Startup procedure on power-up one time
}
void loop()
{
bluetooth();
getGPS();                                     // Update the GPS location
getCompass();                                 // Update the Compass Heading
}
```

<u>Code integrated into the main code:</u>

1. **<u>Startup</u>**

```
void Startup(){
Serial.println("Pause for Startup... ");
for (int i=5; i >= 1; i--)                    // Count down for X seconds
{
Serial1.print("Pause for Startup... ");
Serial1.print(i);
delay(1000);                                  // Delay for X seconds
}
Serial1.println("Searching for Satellites ");
Serial.println("Searching for Satellites ");
while (Number_of_SATS <= 4)                   // Wait until x number of satellites are acquired before starting main loop
{
getGPS();                                     // Update gps data
```

```
Number_of_SATS = (int)(gps.satellites.value());   // Query Tiny GPS for the number of
Satellites Acquired

bluetooth();                                       // Check to see if there are any bluetooth
commands being received

}

setWaypoint();                                     // set intial waypoint to current location

Serial1.print(Number_of_SATS);

Serial1.print(" Satellites Acquired");

}
```

2. Bluetooth Software

```
void bluetooth()

{

while (Serial1.available())

{    {

str = Serial1.readStringUntil('\n');

}

blueToothVal = (str.toInt());                     //  convert the string 'str' into an integer and
assign it to blueToothVal

Serial.print("BlueTooth Value ");

Serial.println(blueToothVal);

switch (blueToothVal)

{

case 1:

Serial1.println("Forward");

Forward();

break;



case 2:

Serial1.println("Reverse");

Reverse();

break;
```

```
case 3:
Serial1.println("Left");
LeftTurn();
StopCar();
break;

case 4:
Serial1.println("Right");
RightTurn();
StopCar();
break;
case 5:
Serial1.println("Stop Car ");
StopCar();
break;
case 6:
setWaypoint();
break;
case 7:
goWaypoint();
break;
case 8:
Serial1.println("Turn Around");
turnAround();
break;
case 9:
Serial1.println("Compass Forward");
setHeading();
Compass_Forward();
```

```
break;


case 10:
setHeading();
break;


case 11:
gpsInfo();
break;


case 12:
Serial1.println("Compass Turn Right");
CompassTurnRight();
break;


case 13:
Serial1.println("Compass Turn Left");
CompassTurnLeft();
break;
case 16:
clearWaypoints();
break;
case 17:                    // finish with waypoints
ac = 0;
Serial1.print("Waypoints Complete");
break;
} // end of switch case
```

3. GPS and Compass data

```
void calibrateCompass()     // Experimental Use Only to Calibrate Magnetometer/ Compass
```

```
{
int minX = 0;
int maxX = 0;
int minY = 0;
int maxY = 0;
int offX = 0;
int offY = 0;

for (int i=1000; i >= 1; i--)
{
Vector mag = compass.readRaw();                    // Read compass data

// Determine Min / Max values
if (mag.XAxis < minX) minX = mag.XAxis;
if (mag.XAxis > maxX) maxX = mag.XAxis;
if (mag.YAxis < minY) minY = mag.YAxis;
if (mag.YAxis > maxY) maxY = mag.YAxis;

offX = (maxX + minX)/2;                           // Calculate offsets
offY = (maxY + minY)/2;
delay(10);
//Serial.print("Compass X & Y offset: ");
//Serial.print(offX);
//Serial.print(" ");
//Serial.print(offY);
//Serial.print("\n");

}                                                 // end of for loop
StopCar();
```

```
Serial1.print("Compass X & Y offset: ");
Serial1.print(offX);
Serial1.print(" ");
Serial1.print(offY);
Serial.print("\n");
compass.setOffset(offX,offY);                    // Set calibration offset
}
void getGPS()                                    // Get Latest GPS coordinates
{
while (Serial2.available() > 0)
gps.encode(Serial2.read());
}
void setWaypoint()                               // Set up to 5 GPS waypoints
{

//if ((wpCount >= 0) && (wpCount < 50))
if (wpCount >= 0)
{
Serial1.print("GPS Waypoint ");
Serial1.print(wpCount + 1);
Serial1.print(" Set ");
getGPS();                                        // get the latest GPS coordinates
getCompass();                                    // update latest compass heading
Home_LATarray[ac] = gps.location.lat();          // store waypoint in an array
Home_LONarray[ac] = gps.location.lng();          // store waypoint in an array
Serial.print("Waypoint #1: ");
Serial.print(Home_LATarray[0],6);
Serial.print(" ");
Serial.println(Home_LONarray[0],6);
```

```
Serial.print("Waypoint #2: ");
Serial.print(Home_LATarray[1],6);
Serial.print(" ");
Serial.println(Home_LONarray[1],6);
Serial.print("Waypoint #3: ");
Serial.print(Home_LATarray[2],6);
Serial.print(" ");
Serial.println(Home_LONarray[2],6);
Serial.print("Waypoint #4: ");
Serial.print(Home_LATarray[3],6);
Serial.print(" ");
Serial.println(Home_LONarray[3],6);
Serial.print("Waypoint #5: ");
Serial.print(Home_LATarray[4],6);
Serial.print(" ");
Serial.println(Home_LONarray[4],6);

wpCount++;                                    // increment waypoint counter
ac++;                                         // increment array counter
}
else {Serial1.print("Waypoints Full");}
}
void clearWaypoints()
{
memset(Home_LATarray, 0, sizeof(Home_LATarray));         // clear the array
memset(Home_LONarray, 0, sizeof(Home_LONarray));         // clear the array
wpCount = 0;                                              // reset increment counter to 0
ac = 0;
Serial1.print("GPS Waypoints Cleared");                  // display waypoints cleared
```

```
void getCompass()                                              // get latest compass value
{


Vector norm = compass.readNormalize();
// Calculate heading
float heading = atan2(norm.YAxis, norm.XAxis);
if(heading < 0)
heading += 2 * M_PI;
compass_heading = (int)(heading * 180/M_PI);                   // assign compass calculation to
variable (compass_heading) and convert to integer to remove decimal places
}
void setHeading()
{
for (int i=0; i <= 5; i++)                  // Take several readings from the compass to insure accuracy
{
getCompass();                                           // get the current compass heading
}
desired_heading = compass_heading;                      // set the desired heading to equal the
current compass heading
HeadingA = compass_heading;                             // set Heading A to current compass
HeadingB = compass_heading + 180;                       // set Heading B to current compass
heading + 180
if (Heading_B >= 360)                                   // if the heading is greater than 360 then
subtract 360 from it, heading must be between 0 and 360
{
Heading_B = Heading_B - 360;
}
Serial1.print("Compass Heading Set: ");
Serial1.print(compass_heading);
Serial1.print(" Degrees");
Serial.print("desired heading");
```

```
Serial.println(desired_heading);
Serial.print("compass heading");
Serial.println(compass_heading);


}
void gpsInfo()                                        // displays Satellite data to user
{
NumberSATS = (int)(gps.satellites.value());        //query tiny GPS for the number of Satellites Acquired
Distance_To_Home = TinyGPSPlus::distanceBetween(gps.location.lat(),gps.location.lng(),Home_LATarray[ac], Home_LONarray[ac]);  //Query Tiny GPS for Distance to Destination
Serial1.print("Lat:");
Serial1.print(gps.location.lat(),6);
Serial1.print(" Lon:");
Serial1.print(gps.location.lng(),6);
Serial1.print(" ");
Serial1.print(Number_of_SATS);
Serial1.print(" SATs ");
Serial1.print(Distance_To_Home);
Serial1.print("m");
Serial.print("Distance to Home ");
Serial.println(Distance_To_Home);
}
---------------------------------------------------------------------------------------------------------------------
```

4. Waypoint Software

```
void goWaypoint()
{
Serial1.println("Go to Waypoint");
//Serial.print("Home_Latarray ");
```

```
//Serial.print(Home_LATarray[ac],6);
//Serial.print(" ");
//Serial.println(Home_LONarray[ac],6);

//Serial1.print("Distance to Home");
//Serial1.print(Distance_To_Home);
//Serial1.print("ac= ");
//Serial1.print(ac);
while (true)
{
bluetooth();
if (blueToothVal == 5){break;}                    // If a 'Stop' Bluetooth command is received then break from the Loop
getCompass();                                     // Update Compass heading
getGPS();                                         // Tiny GPS function that retrieves GPS data - update GPS location// delay time changed from 100 to 10
if (millis() > 5000 && gps.charsProcessed() < 10)     // If no Data from GPS within 5 seconds then send error
Serial1.println(F("No GPS data: check wiring"));
Distance_To_Home = TinyGPSPlus::distanceBetween(gps.location.lat(),gps.location.lng(),Home_LATarray[ac], Home_LONarray[ac]);                    //Query Tiny GPS for Distance to Destination
GPS_Course = TinyGPSPlus::courseTo(gps.location.lat(),gps.location.lng(),Home_LATarray[ac],Home_LONarray[ac]);                          //Query Tiny GPS for Course to Destination
if (Distance_To_Home == 0)              // If the Vehicle has reached it's Destination, then Stop
{
StopCar();                              // Stop the robot after each waypoint is reached
Serial1.println("You have arrived!");   // Print to Bluetooth device - "You have arrived"
ac++;                                   // increment counter for next waypoint
break;                                  // Break from Go_Home procedure and send control back to the Void Loop
```

```
// go to next waypoint


}
if ( abs(GPS_Course - compass_heading) <= 15)                // If GPS Course and the Compass
Heading are within x degrees of each other then go Forward
// otherwise find the shortest turn radius and turn left or right
{
Forward();                                                   // Go Forward
} else
{
int x = (GPS_Course - 360);                                  // x = the GPS desired heading - 360
int y = (compass_heading - (x));                             // y = the Compass heading - x
int z = (y - 360);                                           // z = y - 360


if ((z <= 180) && (z >= 0))                                  // if z is less than 180 and not a
negative value then turn left otherwise turn right
{ SlowLeftTurn();  }
else { SlowRightTurn(); }
}
}                                          // End of While Loop
}                                          // End of Go_Home procedure
```

5. **Steering**

```
void Forward()
{
Serial.println("Forward");
motor1.setSpeed(mtrSpd);
motor2.setSpeed(mtrSpd);
motor3.setSpeed(mtrSpd);
motor4.setSpeed(mtrSpd);
motor1.run(FORWARD);          // go forward all wheels
```

```
    }
    motor2.run(FORWARD);
    motor3.run(FORWARD);
    motor4.run(FORWARD);
    }
    void Forward_Meter()
    {
    motor1.setSpeed(mtrSpd);
    motor2.setSpeed(mtrSpd);
    motor3.setSpeed(mtrSpd);
    motor4.setSpeed(mtrSpd);
    motor1.run(FORWARD);
    motor2.run(FORWARD);
    motor3.run(FORWARD);
    motor4.run(FORWARD);
    delay(500);
    }
void Reverse()
    {
    motor1.setSpeed(mtrSpd);
    motor2.setSpeed(mtrSpd);
    motor3.setSpeed(mtrSpd);
    motor4.setSpeed(mtrSpd);
    motor1.run(BACKWARD);    // Reverse all wheels
    motor2.run(BACKWARD);
    motor3.run(BACKWARD);
    motor4.run(BACKWARD);
    }
    void LeftTurn()
    {
```

```
motor1.setSpeed(mtrSpd);
motor2.setSpeed(mtrSpd);
motor3.setSpeed(mtrSpd);
motor4.setSpeed(mtrSpd);
motor1.run(BACKWARD);    // Turn left
motor2.run(FORWARD);
motor3.run(FORWARD);
motor4.run(BACKWARD);
delay(100);
}

void RightTurn()
{
motor1.setSpeed(mtrSpd);
motor2.setSpeed(mtrSpd);
motor3.setSpeed(mtrSpd);
motor4.setSpeed(mtrSpd);
motor1.run(FORWARD);
motor2.run(BACKWARD);
motor3.run(BACKWARD);
motor4.run(FORWARD);
delay(100);
}
void SlowLeftTurn()
{
motor1.setSpeed(TSpeed);
motor2.setSpeed(TSpeed);
motor3.setSpeed(TSpeed);
motor4.setSpeed(TSpeed);
```

```
motor1.run(BACKWARD);

motor2.run(FORWARD);

motor3.run(FORWARD);

motor4.run(BACKWARD);

}


void SlowRightTurn()

{

motor1.setSpeed(TSpeed);

motor2.setSpeed(TSpeed);

motor3.setSpeed(TSpeed);

motor4.setSpeed(TSpeed);

motor1.run(FORWARD);

motor2.run(BACKWARD);

motor3.run(BACKWARD);

motor4.run(FORWARD);

}

void StopCar()

{

motor1.run(RELEASE);

motor2.run(RELEASE);

motor3.run(RELEASE);

motor4.run(RELEASE);

}

void CompassTurnRight()                    // This Function Turns the car 90 degrees to
the right based on the current desired heading

{

StopCar();

getCompass();                              // get current compass heading
```

```
    desired_heading = (desired_heading + 90);     // set desired_heading to plus 90 degrees

    if (desired_heading >= 360) {desired_heading = (desired_heading - 360);}       // if the desired heading is greater than 360 then subtract 360 from it

while ( abs(desired_heading - compass_heading) >= compass_dev)                    // if the desired heading is more than Compass Deviation in degrees from the actual compass heading then

    {                                              // correct direction by turning left or right

    getCompass();                                  // update compass heading during While Loop

    bluetooth();                                   // if new bluetooth value received break from loop

    if (blueToothVal == 5){break;}                 // if a Stop Bluetooth command ('5') is received then break from the Loop

    if (desired_heading >= 360) {desired_heading = (desired_heading - 360);}     // if the desired heading is greater than 360 then subtract 360 from it

    int x = (desired_heading - 359);                            // x = the GPS desired heading - 360

    int y = (compass_heading - (x));                            // y = the Compass heading - x

    int z = (y - 360);                                          // z = y - 360

    if ((z <= 180) && (z >= 0))                                 // if z is less than 180 and not a negative value then turn left

    {                                                           // otherwise turn right

    SlowLeftTurn();

    }

    else

    {

    SlowRightTurn();

    }

    }

    {

    StopCar();                                          // Stop the Car when desired heading and compass heading match

    }

    }
```

```
void CompassTurnLeft()                          // This procedure turns the car 90 degrees to the left based on the current desired heading

{

StopCar();

getCompass();                                   // get current compass heading

//desired_heading = (compass_heading - 90);     // set desired_heading to compass value minus 90 degrees



desired_heading = (desired_heading - 90);       // set desired_heading to minus 90 degrees

if (desired_heading <= 0) {desired_heading = (desired_heading + 360);}      // if the desired heading is greater than 360 then add 360 to it

while ( abs(desired_heading - compass_heading) >= compass_dev)              // If the desired heading is more than Compass Deviation in degrees from the actual compass heading then

{                                               // correct direction by turning left or right

getCompass();                                   // Get compass heading again during While Loop

bluetooth();                                    // if new bluetooth value received break from loop

if (blueToothVal == 5){break;}                  // If a 'Stop' Bluetooth command is received then break from the Loop

if (desired_heading >= 360) {desired_heading = (desired_heading - 360);}    // if the desired heading is greater than 360 then subtract 360 from it

int x = (desired_heading - 359);                          // x = the desired heading - 360

int y = (compass_heading - (x));                          // y = the Compass heading - x

int z = (y - 360);                                        // z = y - 360

if (z <= 180)                                             // if z is less than 180 and not a negative value then turn left

// if ((z <= 180) && (z >= 0))

{                                                         // otherwise turn right

SlowLeftTurn();

}

else

{
```

```
SlowRightTurn();
}
}
{
StopCar();                          // Stop the Car when desired heading and compass heading
match
}
}

void Compass_Forward()
{
while (blueToothVal == 9)      // Go forward until Bluetooth 'Stop' command is sent
//while (true)
{
getCompass();                       // Update Compass Heading
bluetooth();                        // Check to see if any Bluetooth commands have been sent
if (blueToothVal == 5) {break;}     // If a Stop Bluetooth command ('5') is received then
break from the Loop
if ( abs(desired_heading - compass_heading) <= compass_dev )      // If the Desired Heading
and the Compass Heading are within the compass deviation, X degrees of each other then
Go Forward
// otherwise find the shortest turn radius and turn left or right
{
Forward();
} else
{
int x = (desired_heading - 359);                    // x = the GPS desired heading - 360
int y = (compass_heading - (x));                    // y = the Compass heading - x
int z = (y - 360);                                  // z = y - 360
if ((z <= 180) && (z >= 0))                         // if z is less than 180 and not a negative
value then turn left
```

```
{                                            // otherwise turn right
SlowLeftTurn();
}
else
{
SlowRightTurn();
}
}
}                                        // End While Loop
}                                         // End Compass_Forward
void turnAround()                        // This procedure turns the Car around 180
degrees, every time the "Turn Around" button is pressed
{                                        // the car alternates whether the next turn will be
to the left or right - this is determined by the 'pass' variable
// Imagine you are cutting the grass, when you get to the end of the row - the first pass you
are turning one way and on the next pass you turn the opposite
if (pass == 0) { CompassTurnRight(); }   // If this is the first pass then turn right


else { CompassTurnLeft(); }              // If this is the second pass then turn left
//Forward_Meter();                       // Go forward one meter (approximately)
StopCar();                               // Stop the car
if (pass == 0)                           // If this is the first pass then Turn Right
{
CompassTurnRight();                      // Turn right
pass = 1 ;                               // Change the pass value to '1' so that the next
turn will be to the left
}
else
{
if (desired_heading == Heading_A)        // This section of code Alternates the desired
heading 180 degrees
```

```
{                                                  // for the Compass drive forward
desired_heading = Heading_B;
}
else if (desired_heading == Heading_B)
{
desired_heading = Heading_A;
}
CompassTurnLeft();                               // If this is the second pass then Turn Left
pass = 0;                                        // Change the pass value to '0' so that the next turn
will be to the right
}
Compass_Forward();                               // Maintain the 'desired heading' and drive forward
```

APPENDIX E: PHRASE ACCURACY DATA

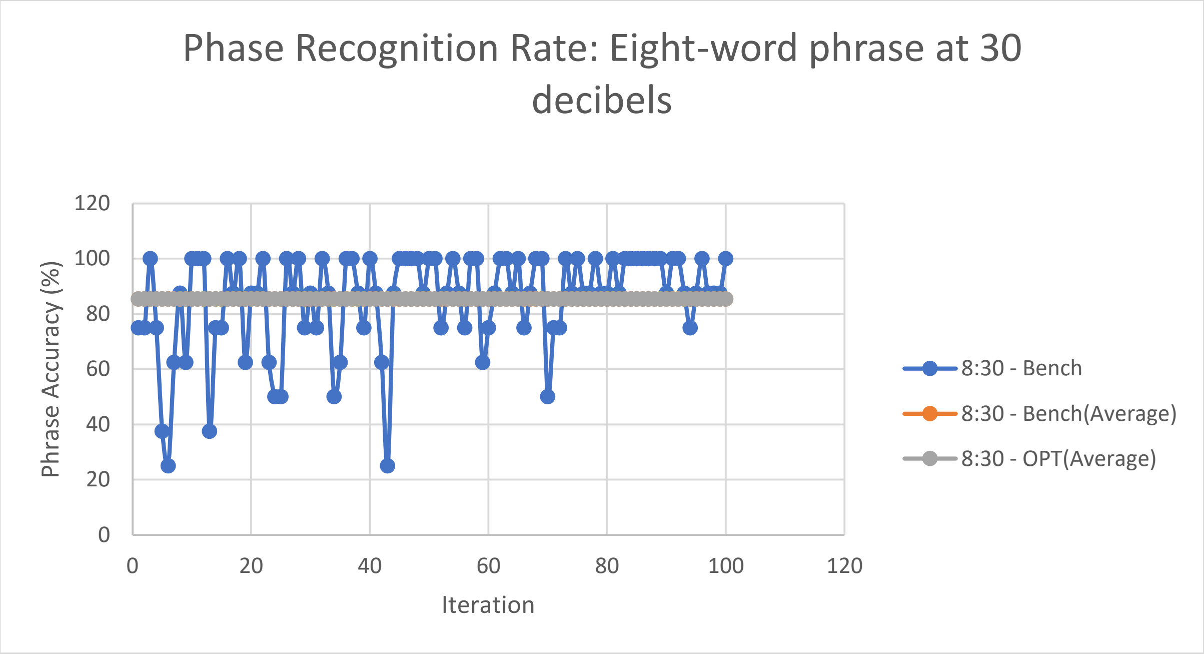


Figure 13-1:Phase Recognition Rate: Eight-word phrase at 30 decibels

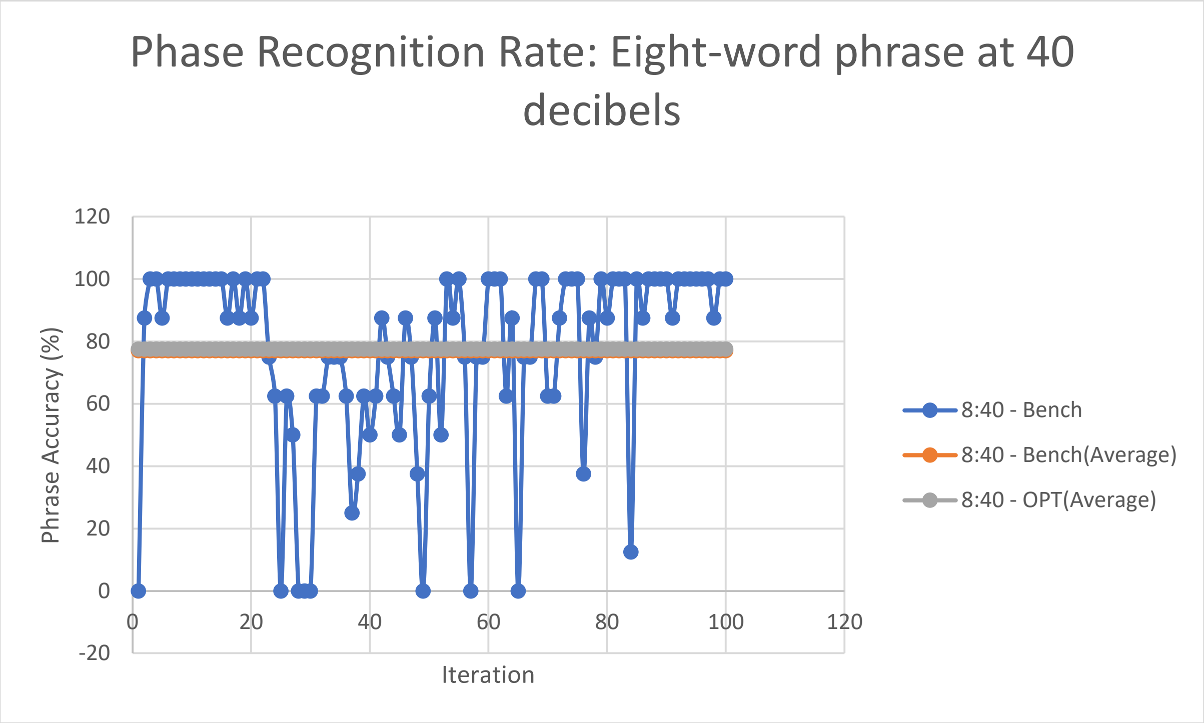


Figure 13-2: Phase Recognition Rate: Eight-word phrase at 40 decibels

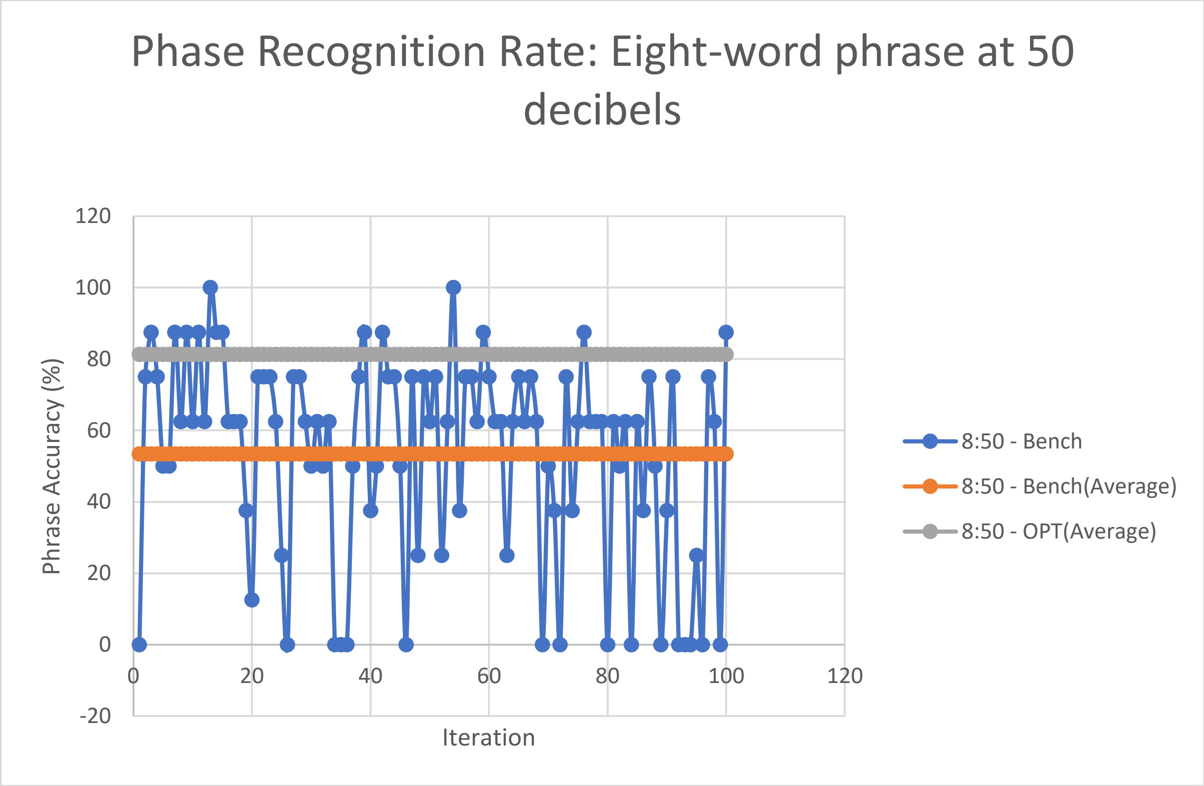


Figure 13-3: Phase Recognition Rate: Eight-word phrase at 50 decibels

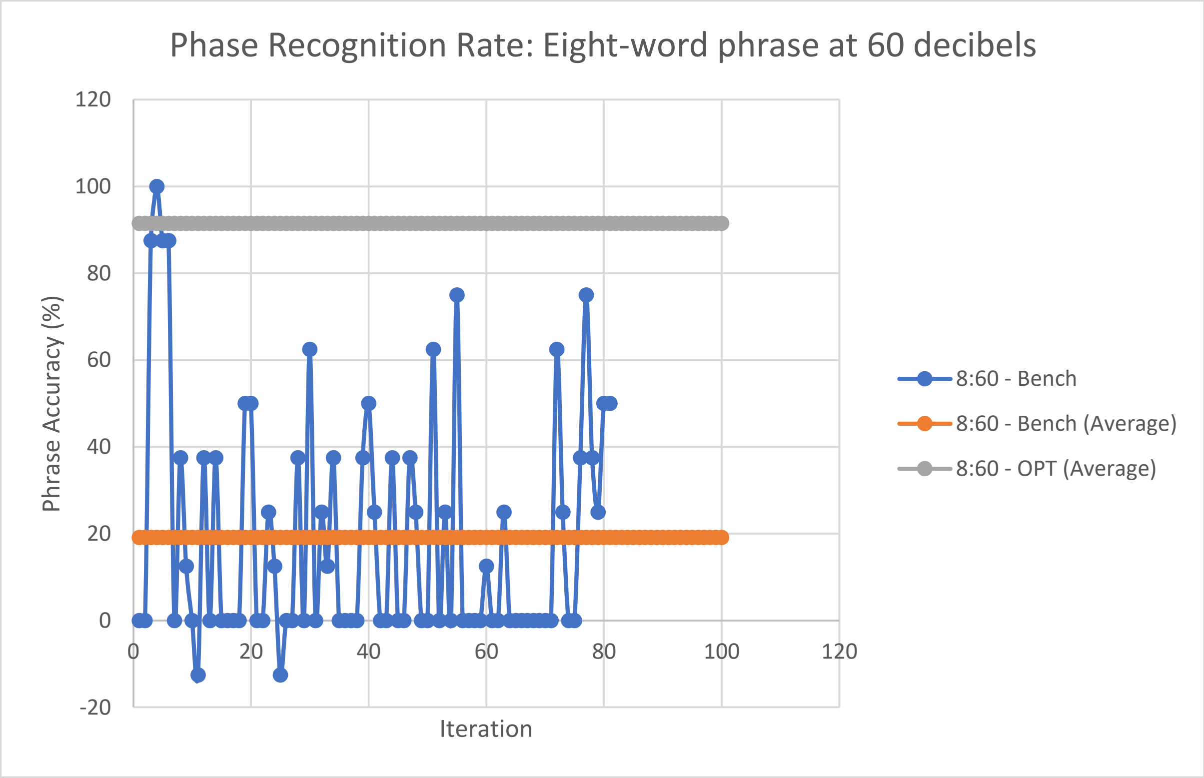


Figure 13-4: Phase Recognition Rate: Eight-word phrase at 60 decibels

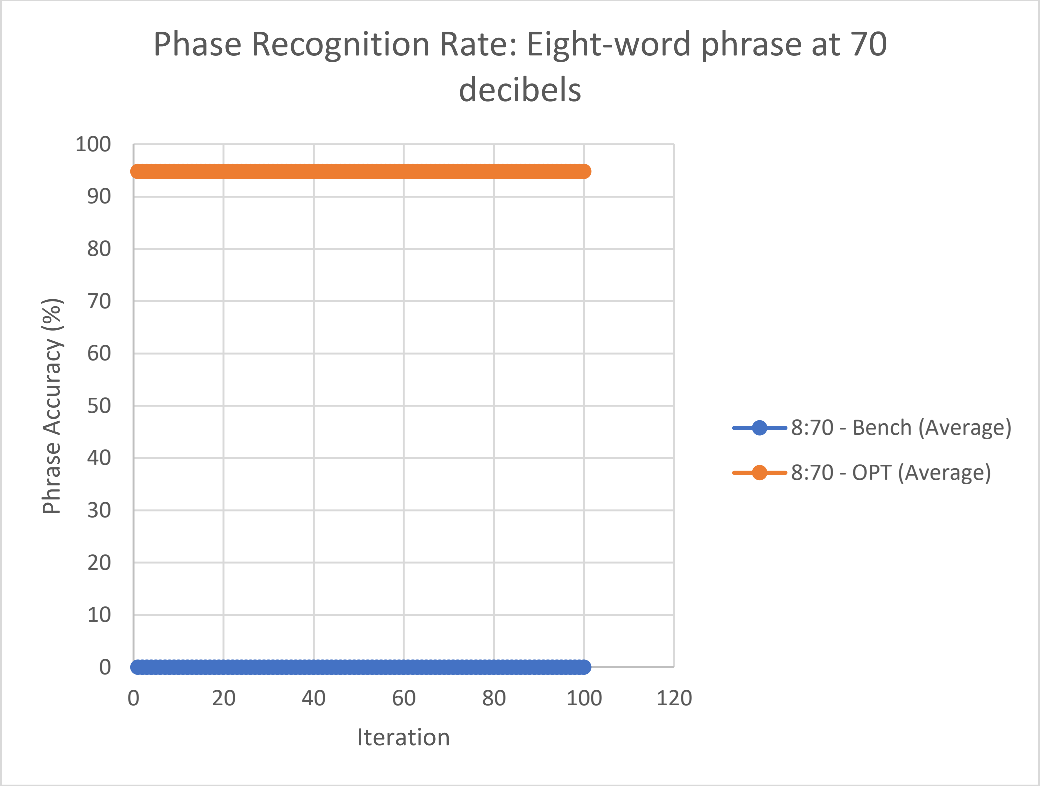


Figure 13-5: Phase Recognition Rate: Eight-word phrase at 70 decibels

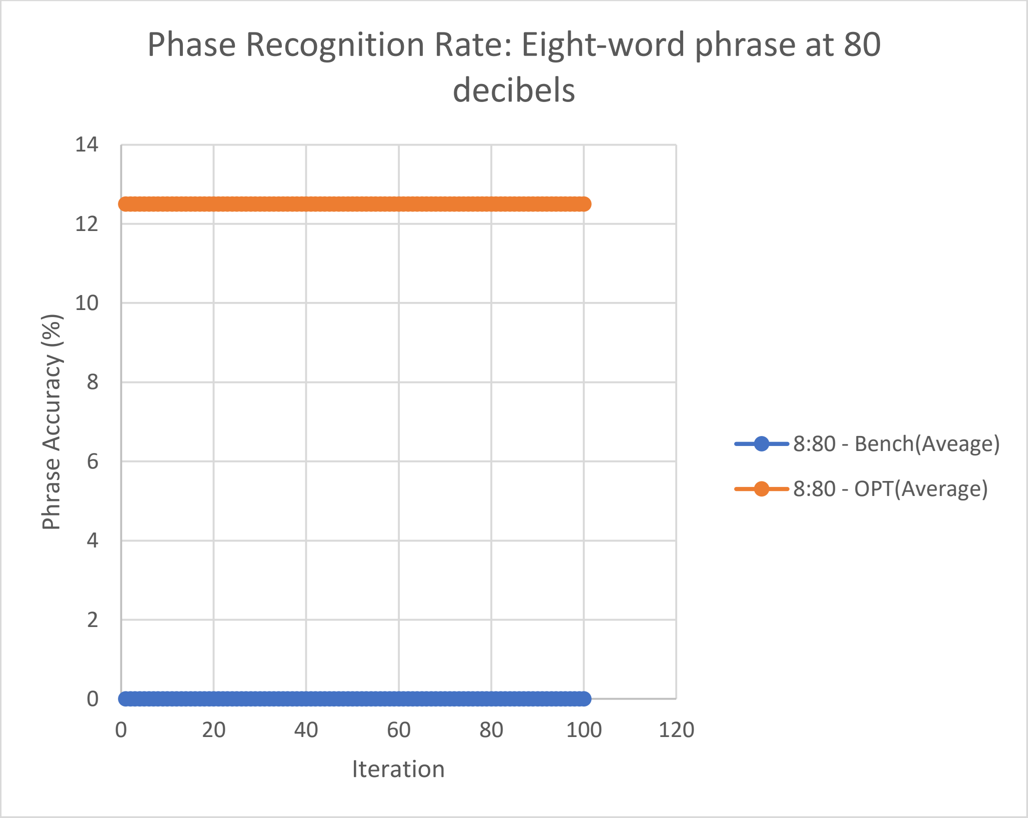


Figure 13-6: Phase Recognition Rate: Eight-word phrase at 80 decibels

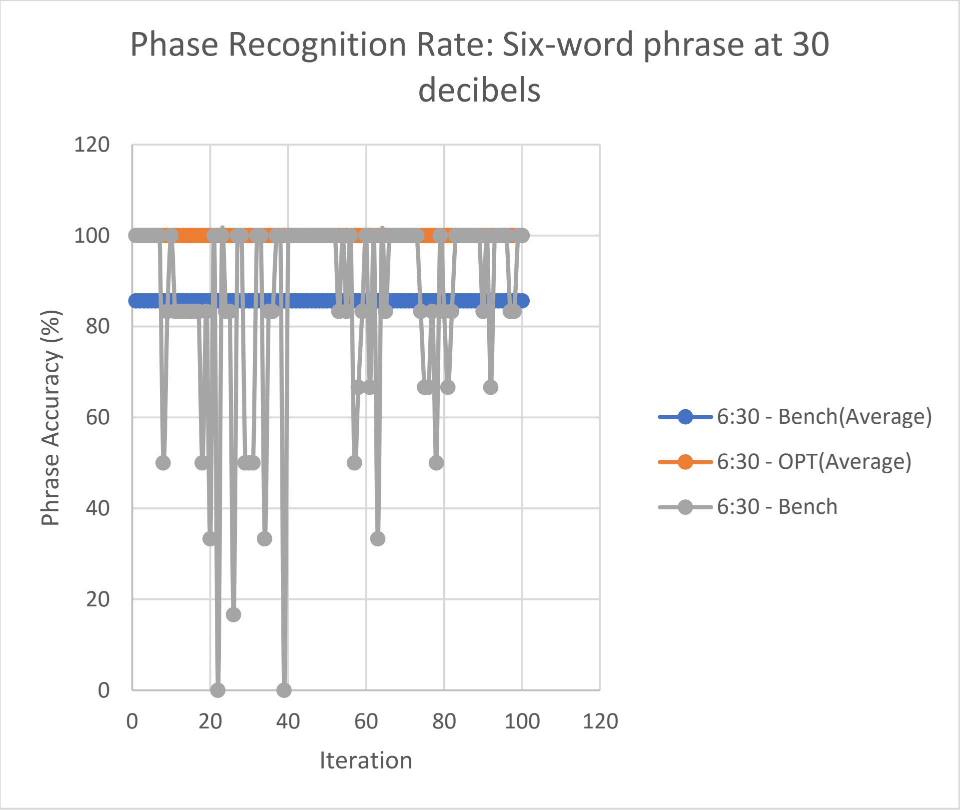


Figure 13-7: Phase Recognition Rate: Six-word phrase at 30 decibels

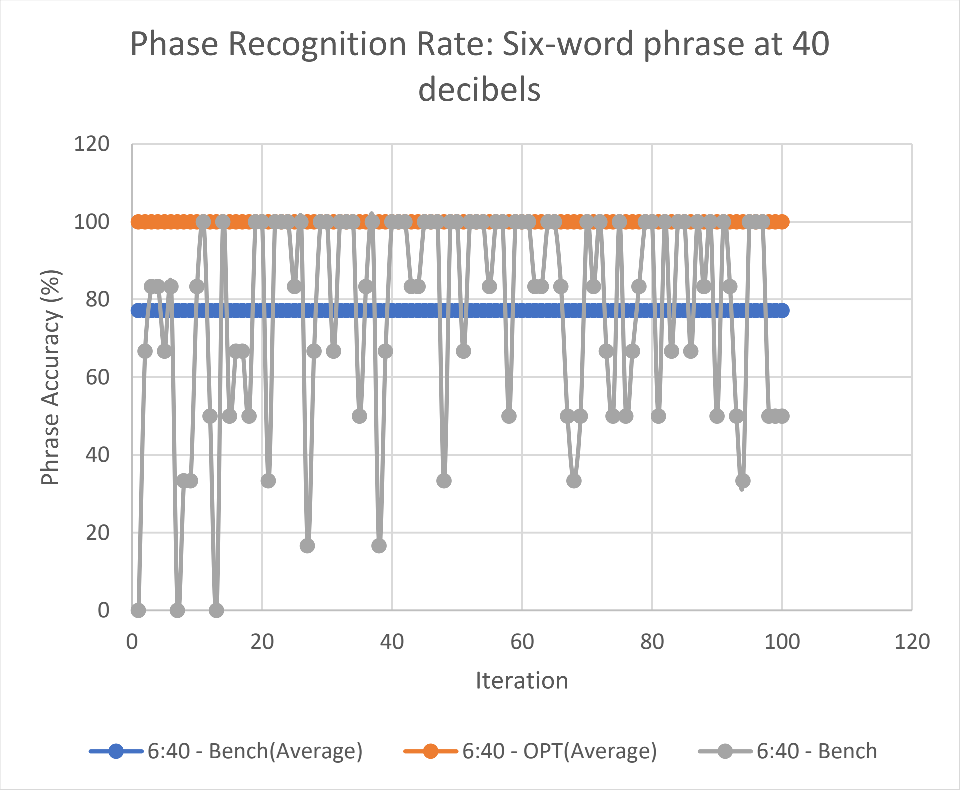


Figure 13-8: Phase Recognition Rate: Six-word phrase at 40 decibels

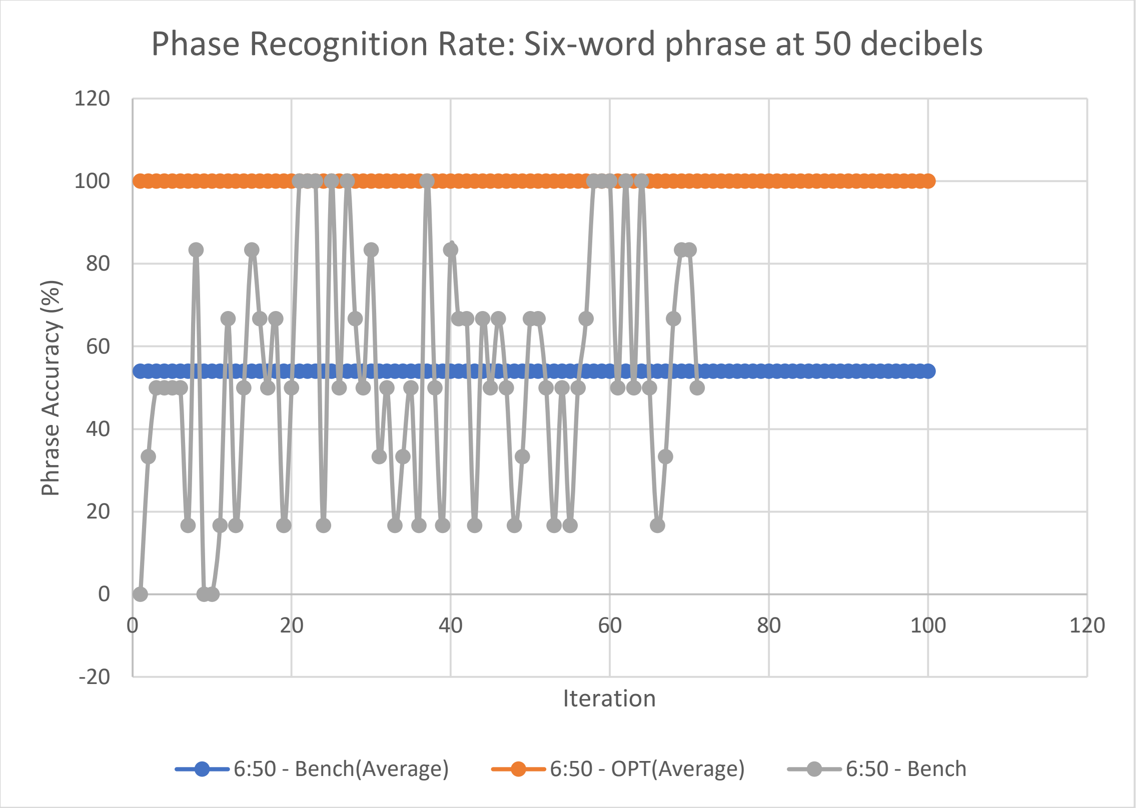


Figure 13-9: Phase Recognition Rate: Six-word phrase at 50 decibels

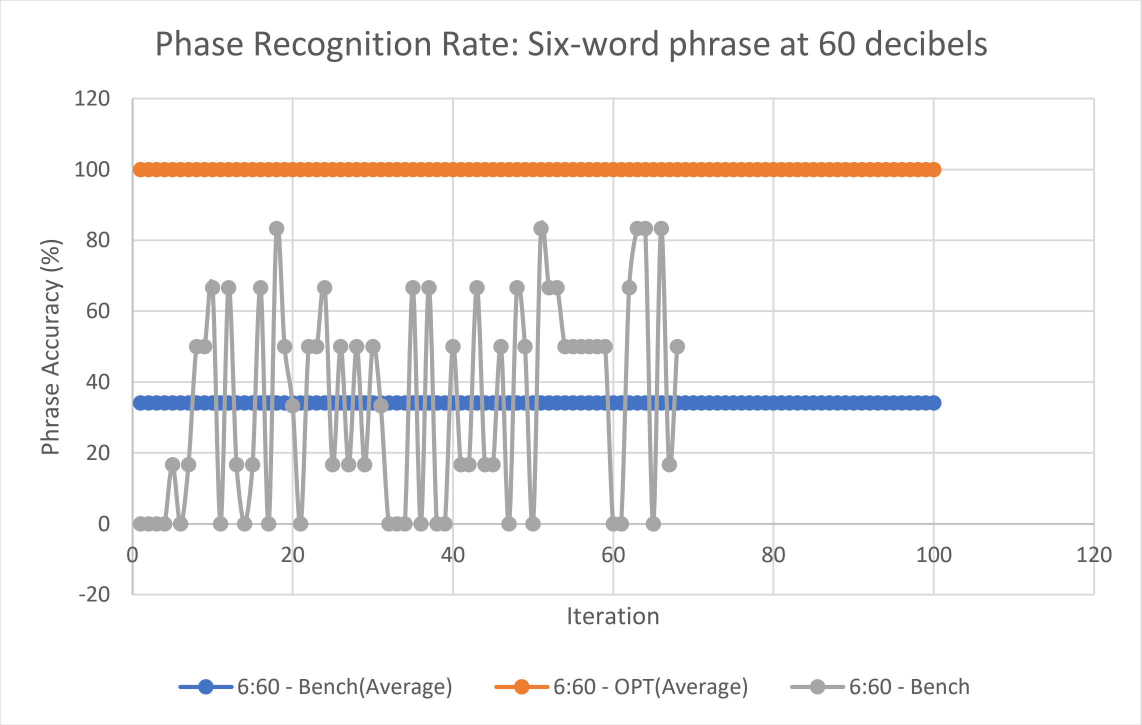


Figure 13-10: Phase Recognition Rate: Six-word phrase at 60 decibels

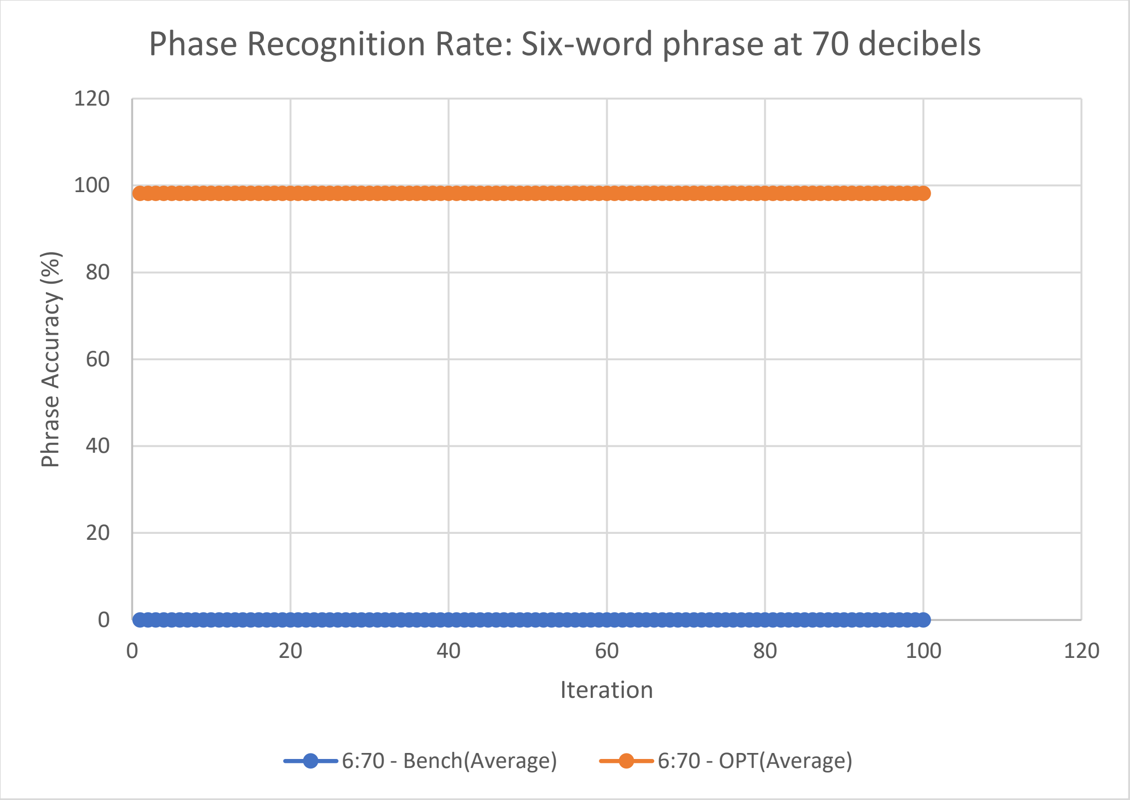


Figure 13-11: Phase Recognition Rate: Six-word phrase at 70 decibels

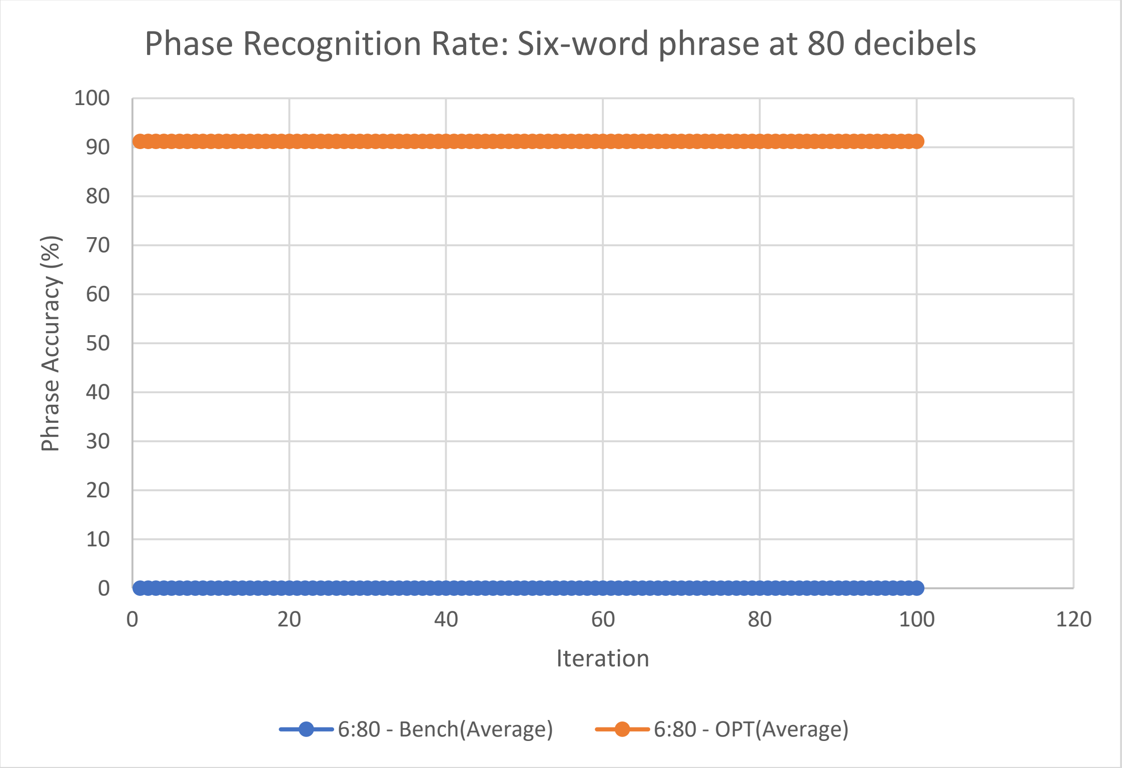


Figure 13-12: Phase Recognition Rate: Six-word phrase at 80 decibels

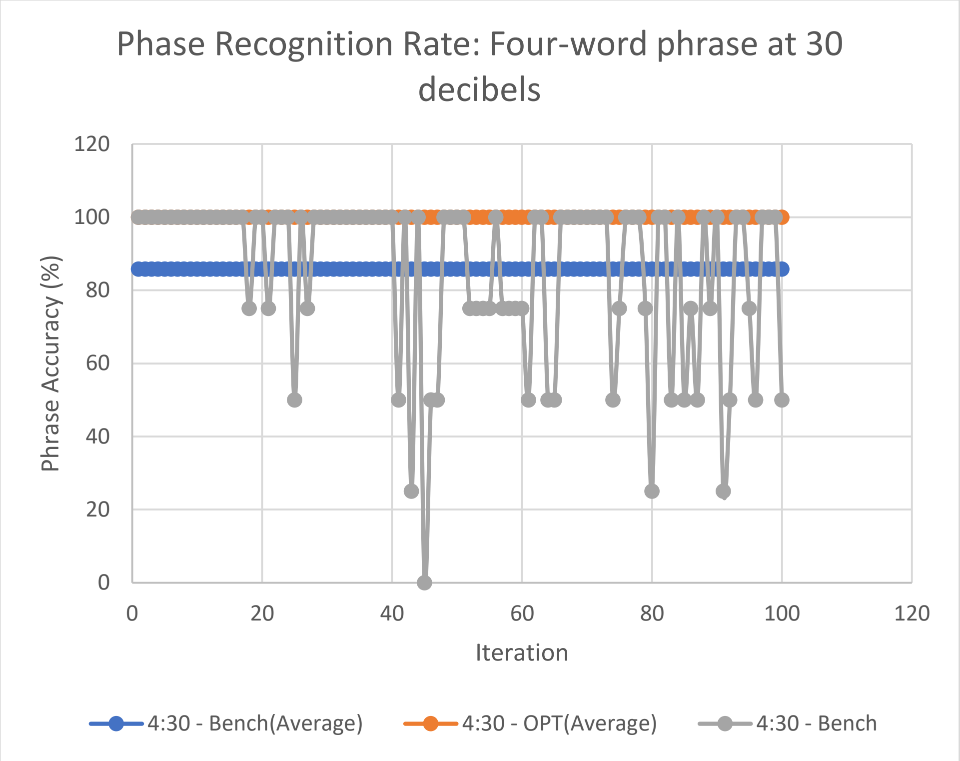


Figure 13-13: Phase Recognition Rate: Four-word phrase at 30 decibels

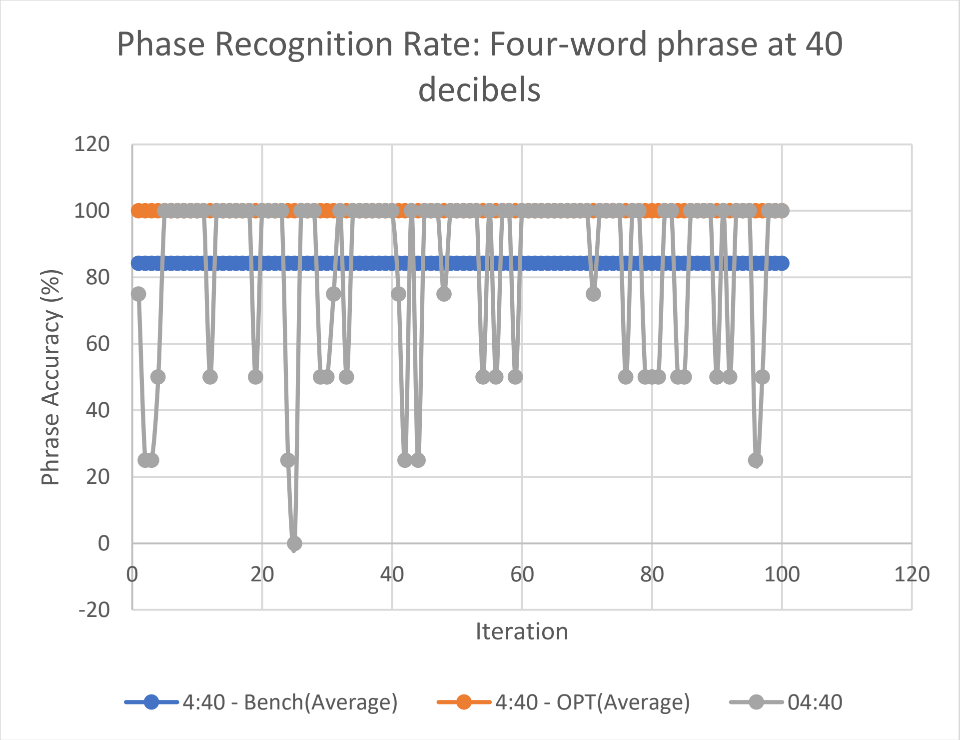


Figure 13-14: Phase Recognition Rate: Four-word phrase at 40 decibels

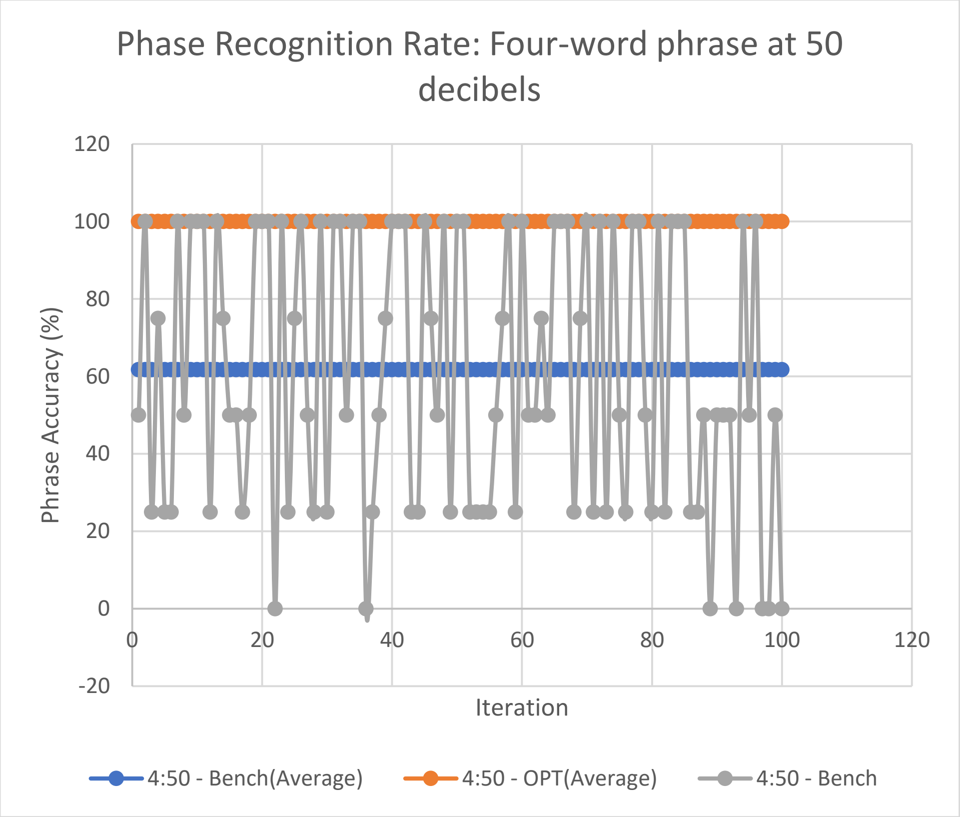


Figure 13-15: Phase Recognition Rate: Four-word phrase at 50 decibels

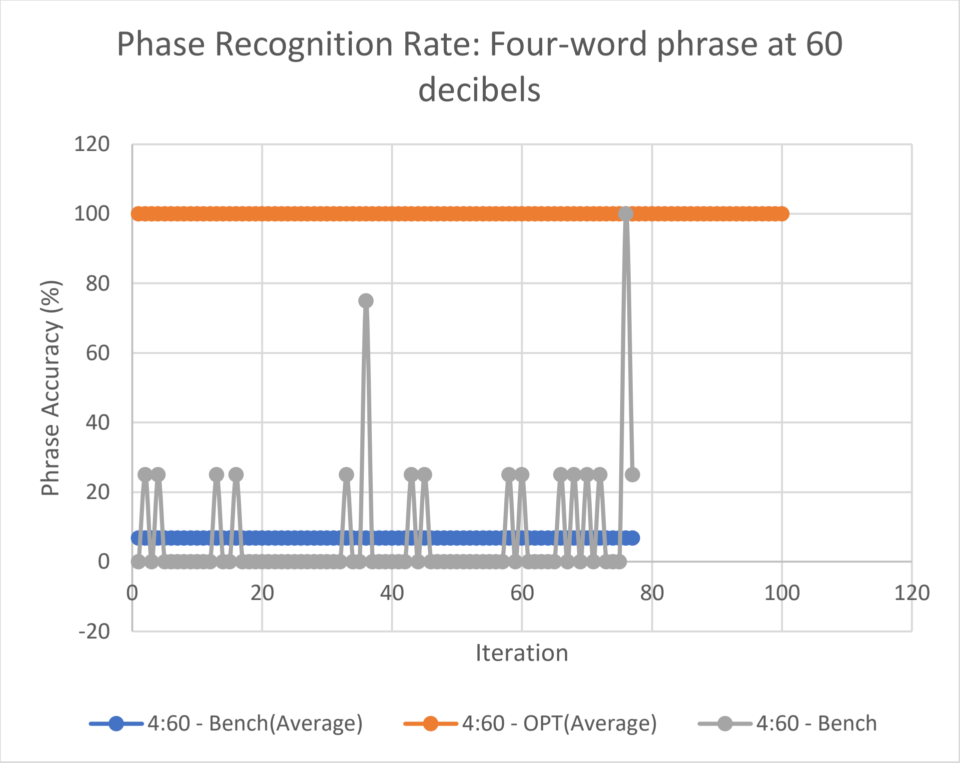


Figure 13-16: Phase Recognition Rate: Four-word phrase at 60 decibels

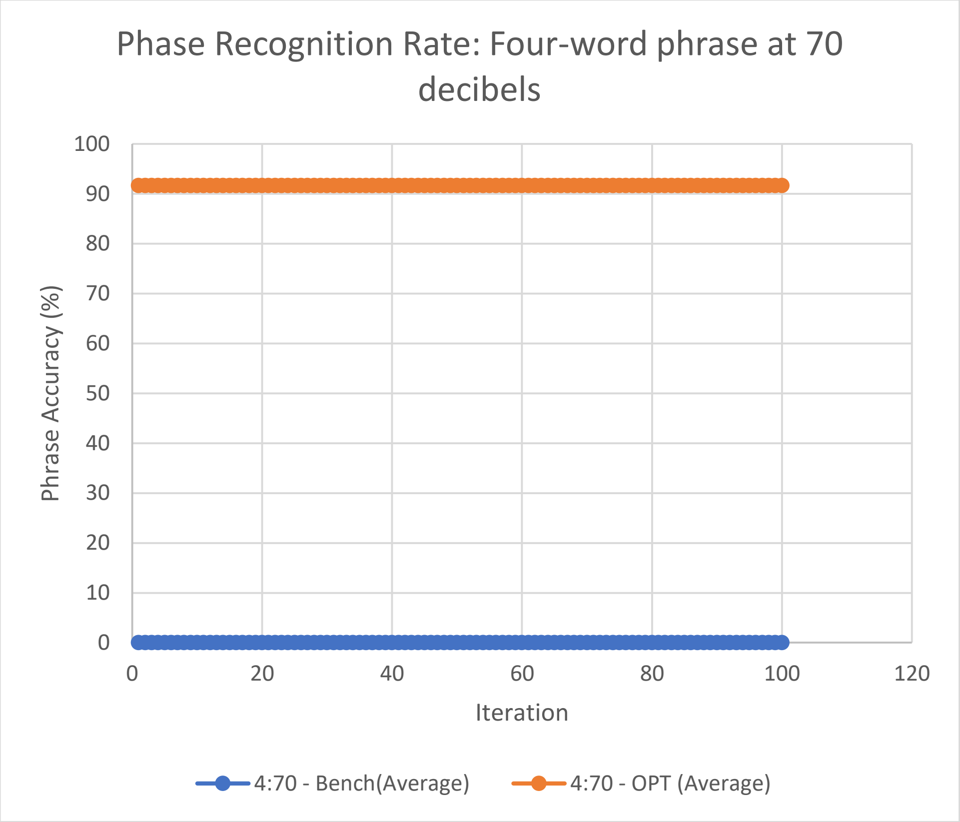


Figure 13-17: Phase Recognition Rate: Four-word phrase at 70 decibels

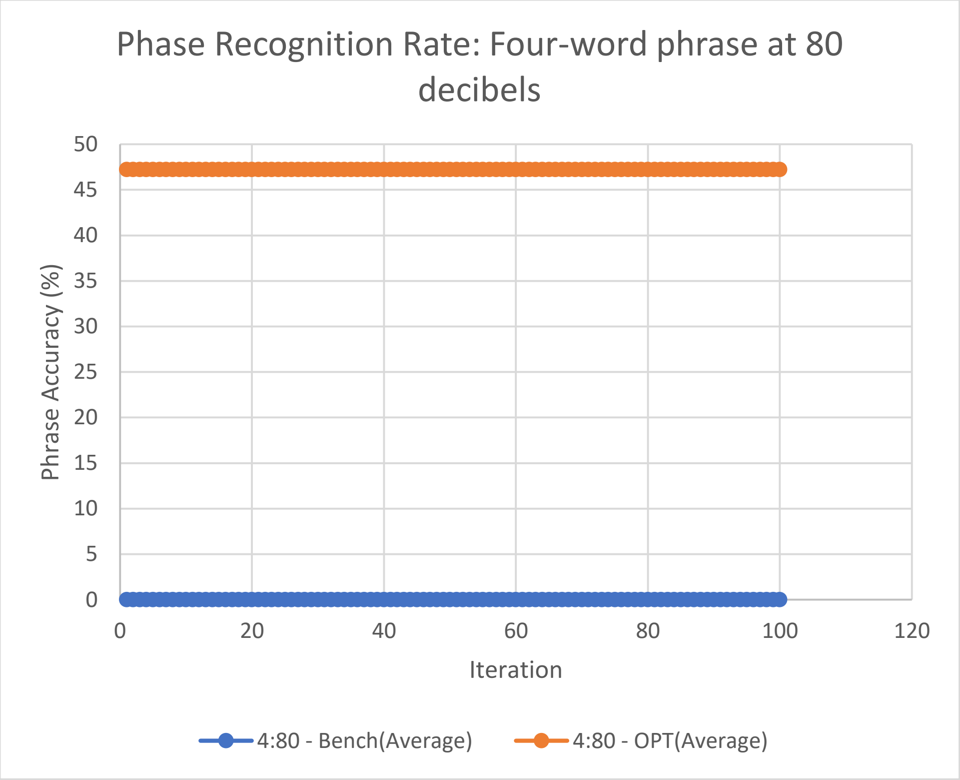


Figure 13-18: Phase Recognition Rate: Four-word phrase at 80 decibels

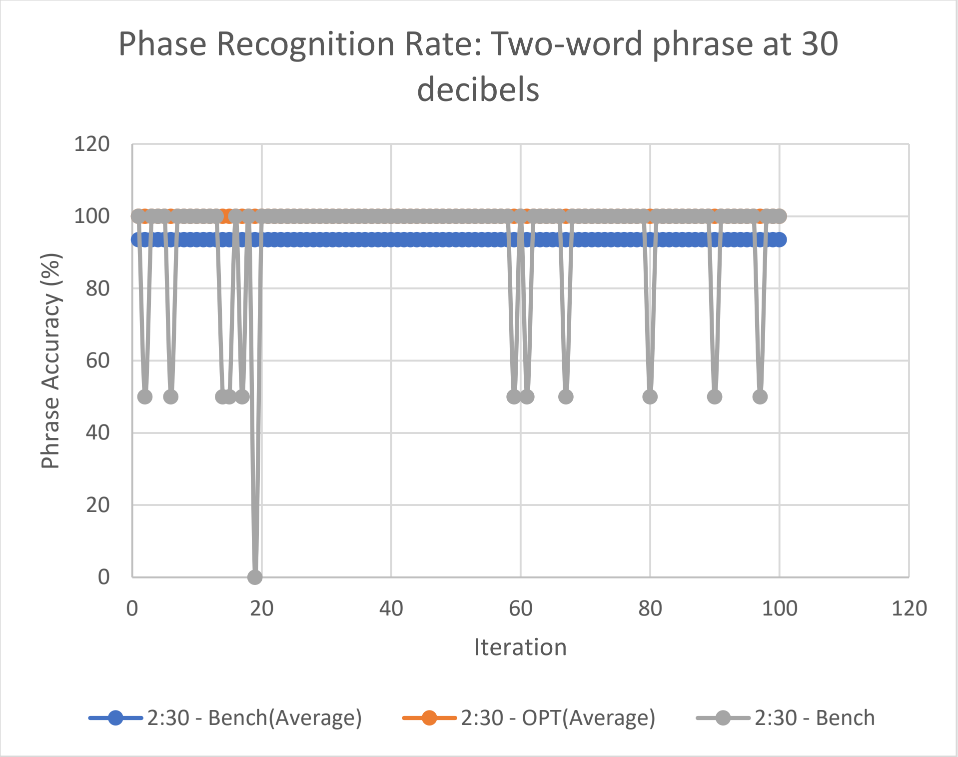


Figure 13-19: Phase Recognition Rate: Two-word phrase at 30 decibels

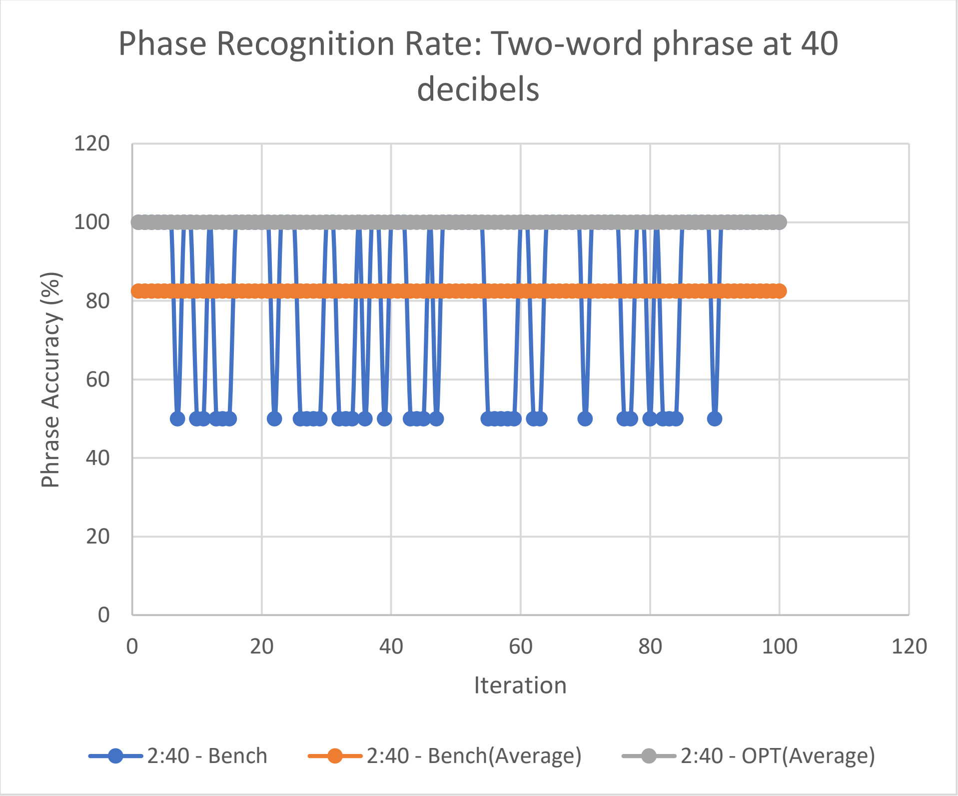


Figure 13-20:Phase Recognition Rate: Two-word phrase at 40 decibels

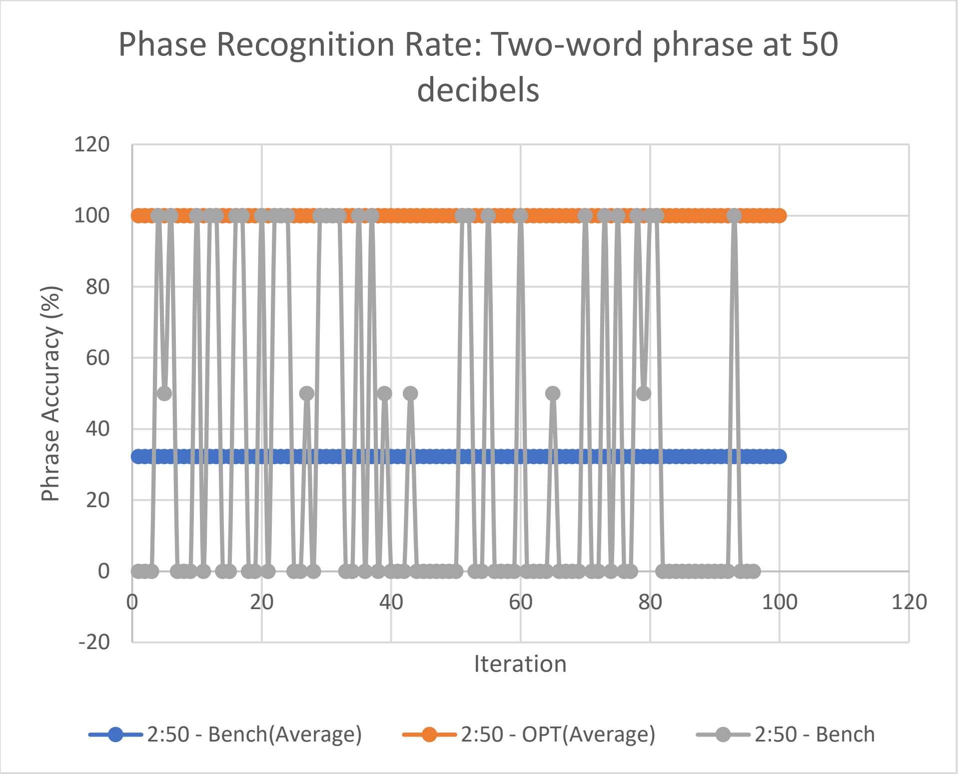


Figure 13-21: Phase Recognition Rate: Two-word phrase at 50 decibels

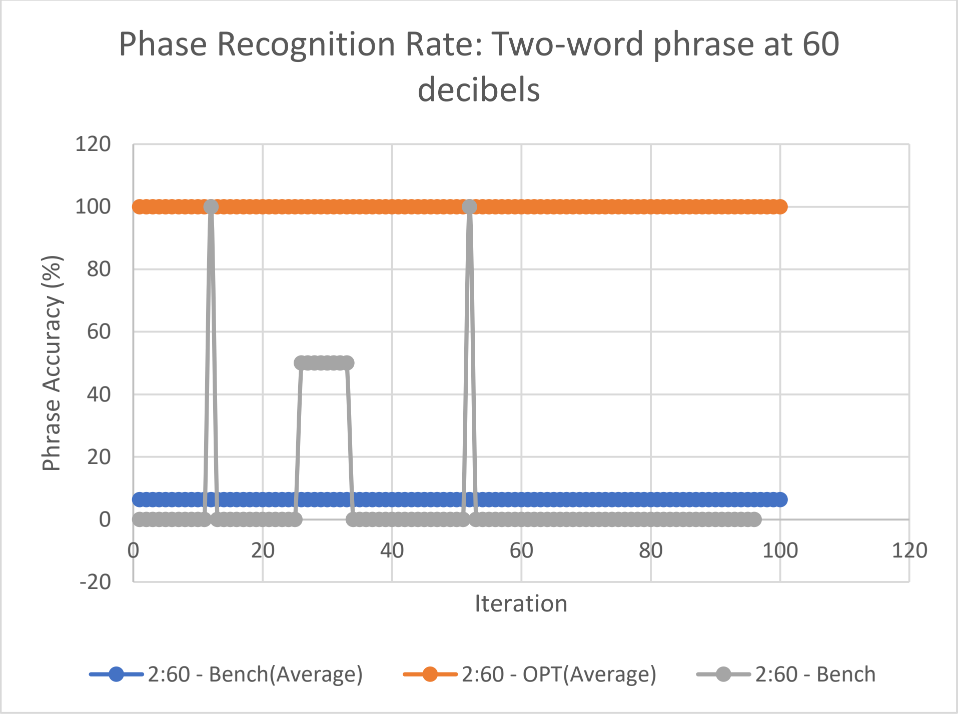


Figure 13-22: Phase Recognition Rate: Two-word phrase at 60 decibels

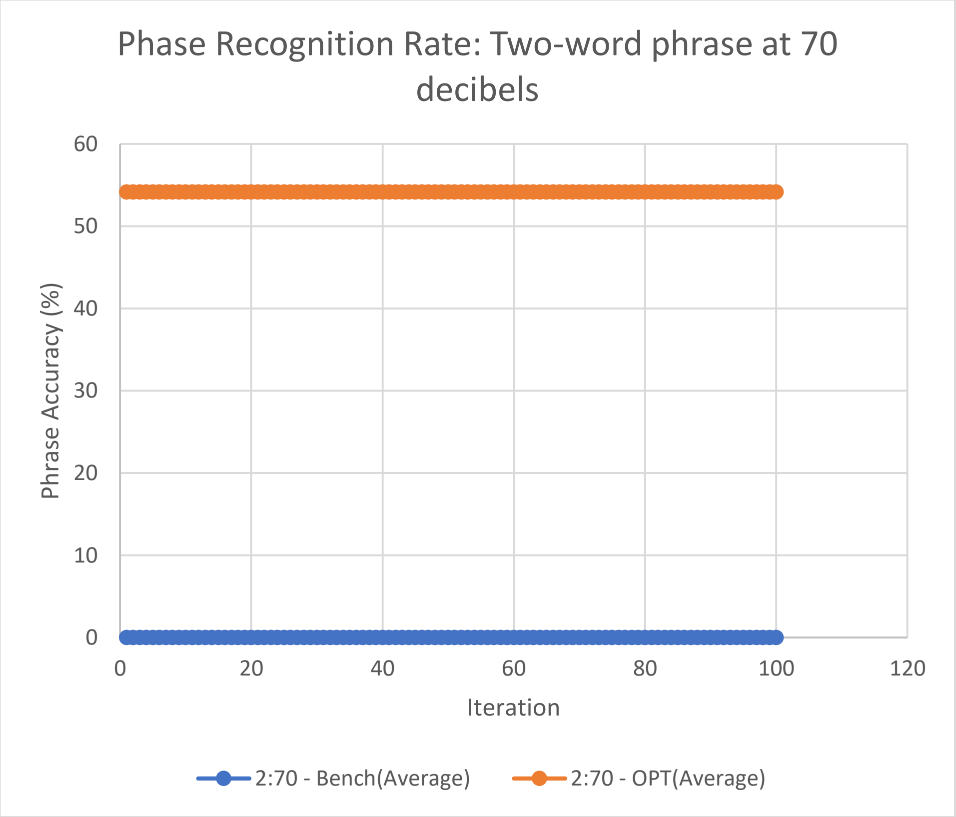


Figure 13-23: Phase Recognition Rate: Two-word phrase at 70 decibels

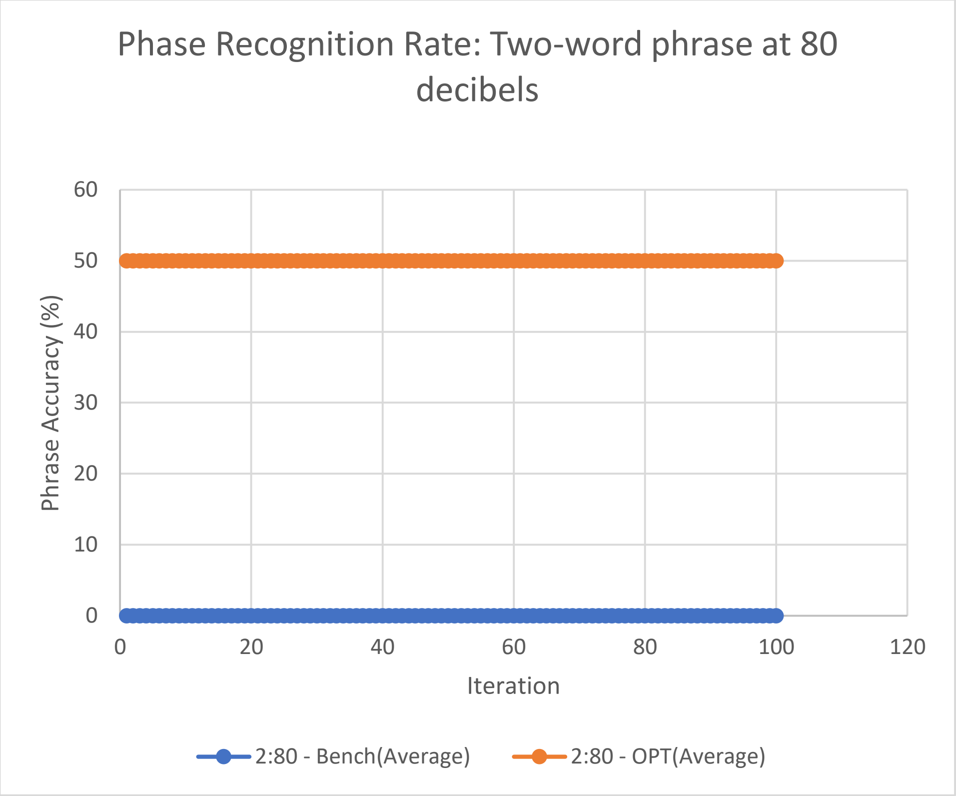


Figure 13-24: Phase Recognition Rate: Two-word phrase at 80 decibels

Figure 13-25: Phase Recognition Rate: One-word phrase at 30 decibels

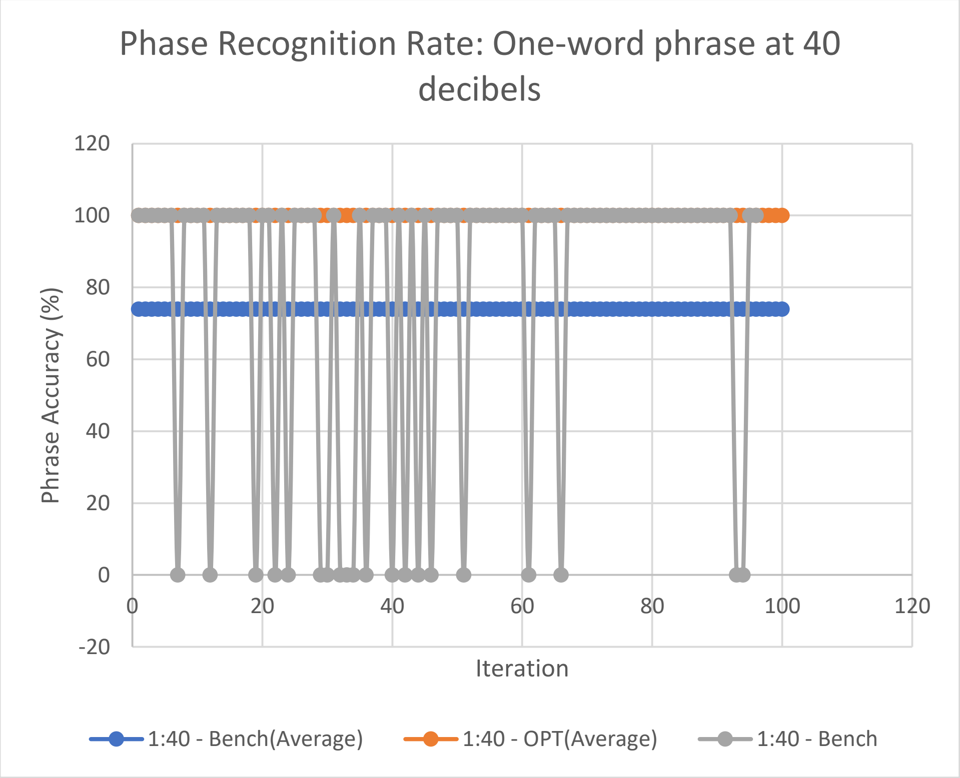


Figure 13-26: Phase Recognition Rate: One-word phrase at 40 decibels

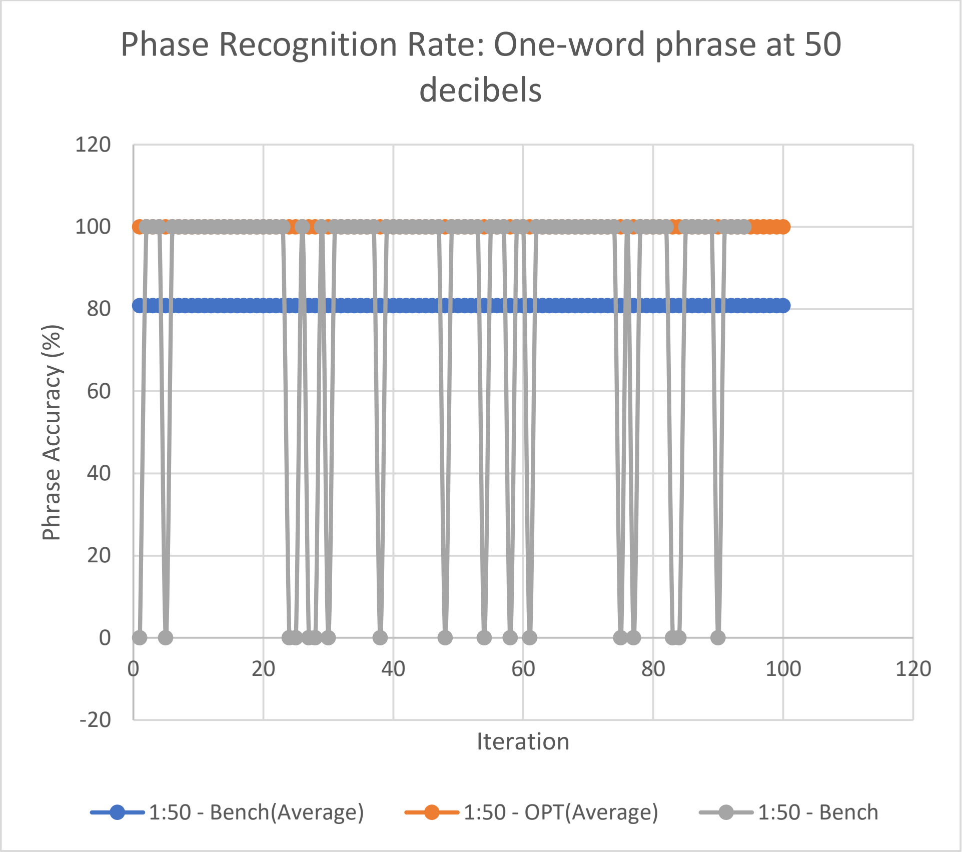


Figure 13-27: Phase Recognition Rate: One-word phrase at 50 decibels

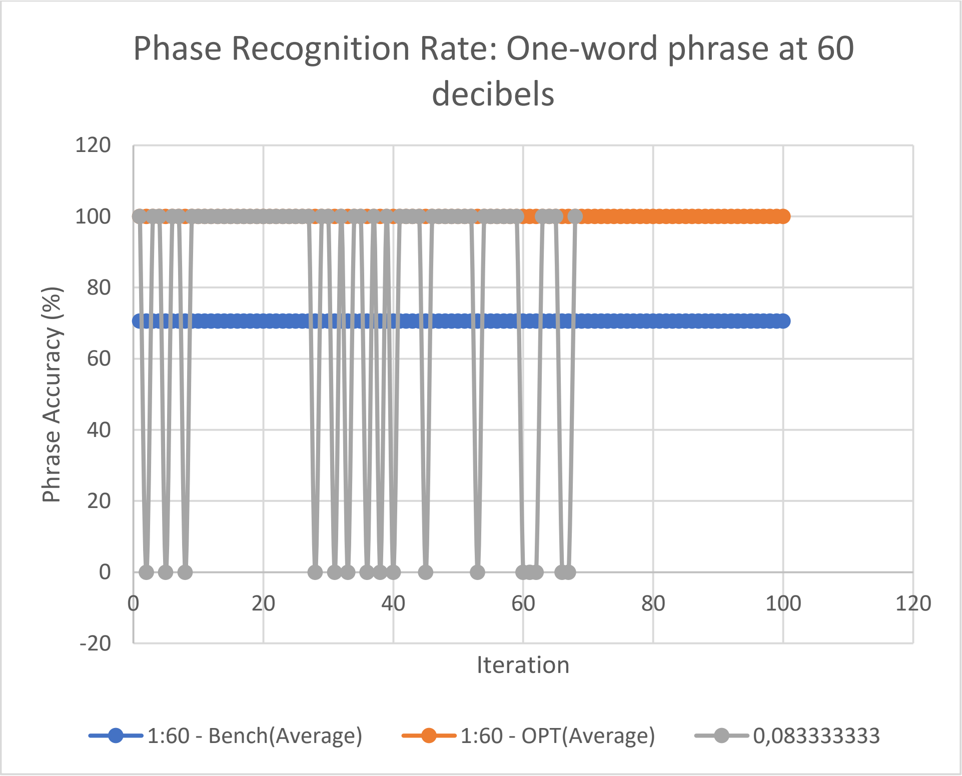


Figure 13-28: Phase Recognition Rate: One-word phrase at 60 decibels

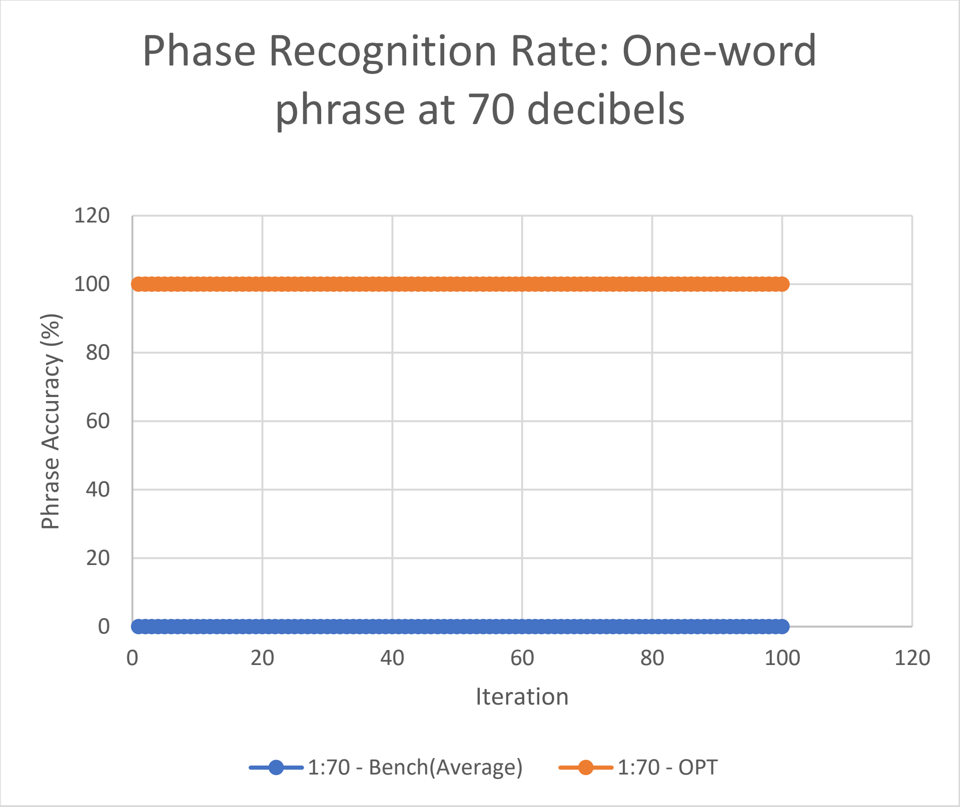


Figure 13-29: Phase Recognition Rate: One-word phrase at 70 decibels

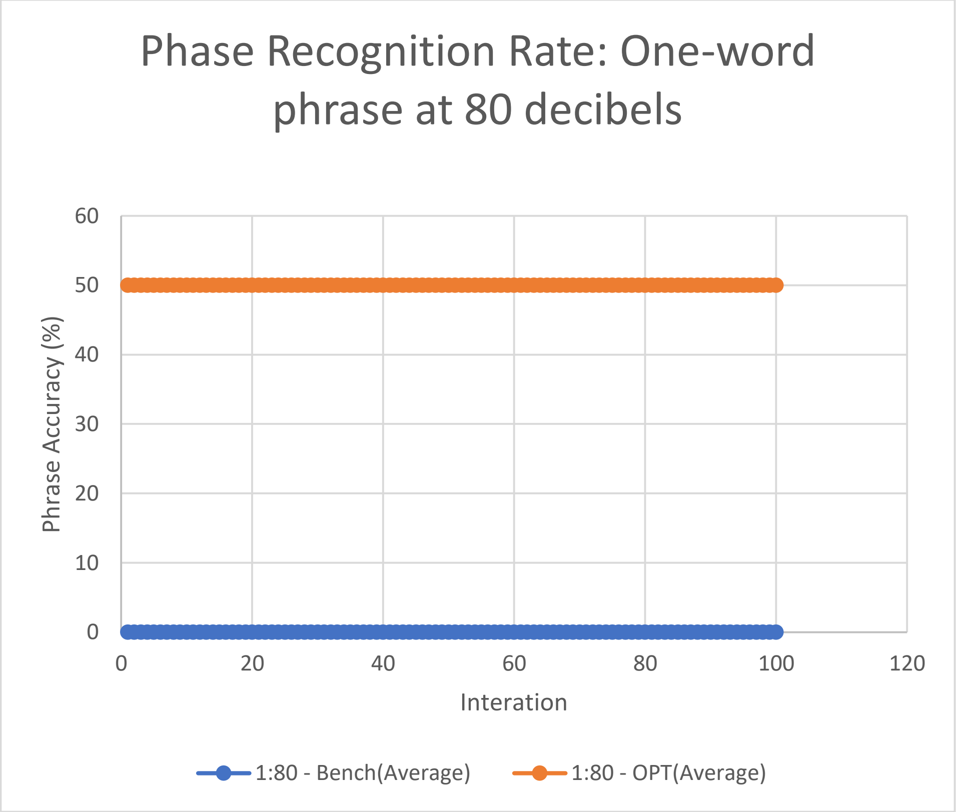


Figure 13-30:Phase Recognition Rate: One-word phrase at 80 decibels