IRIS: EMOTIONALLY INTELLIGENT COMPANION ROBOT

The
BRITISH UNIVERSITY
IN EGYPT

By

MARIAM MOSTAFA HUSSIEN KAMEL SHEHATA

RAMI HESHAM MOHMED ABDOU ABDELHAMEED

SAIF EL DIN MOHAMED SAAD MOHAMED RADWAN

ZAHY YOUSSEF ELGENDY LABIB ELGENDY

A Thesis Submitted to the Faculty of Engineering,

The British University in Egypt, (**BUE**)

In Partial Fulfilment of the Requirements for

The **Bachelor of Engineering (BEng)** Degree in

Robotics Engineering

June 2025

IRIS: EMOTIONALLY INTELLIGENT COMPANION ROBOT

Thesis Approved:

Thesis Advisor Name	**Signature / Date**
----------	**----------**

Internal Committee Member	**Signature / Date**
----------	**----------**

External Committee member	**Signature / Date**
----------	**----------**

Head of Mechanical Engineering Department

Faculty of Engineering. The British University in Egypt

Name: _____

Signature: _____ **Date: _____**

Declaration of Authorship

I hereby declare that this work titled “IRIS: EMOTIONALLY INTELLIGENT COMPANION ROBOT” submitted to the Mechanical Engineering department of the British University in Egypt is an original piece of work and has not been published or showcased anywhere else in the past. This work was carried out under the supervision of DR. Mostafa Abdelaziz

Name: Mariam Mostafa Hussien Kamel Shehata

Signature / Date: 14th June 2025

Name: Rami Hesham Mohamed Abdou Abdelhameed

Signature / Date: 14th June 2025

Name: Saif El Din Mohamed Saad Mohamed Radwan

Signature / Date: 14th June 2025

Name: Zahy Youssef Elgendy Labib Elgendy

Signature / Date: 14th June 2025

Acknowledgements

We would like to express our deep gratitude to Dr. Mostafa Abdelaziz, our graduation research supervisor and mentor, for his patient guidance, enthusiastic encouragement, and valuable critiques throughout the course of this project.

We also extend our sincere appreciation to Prof. Ayman Abbas, Head of the Mechatronics and Robotics Department, for his continuous support and commitment. His leadership has transformed our department into an inspiring and supportive environment that fosters innovation, hands on learning, and academic excellence.

We gratefully acknowledge the support of The Academy of Scientific Research & Technology (ASRT) for funding our project with 58,750 EGP, which significantly contributed to the successful development of our robot.

We are truly grateful to the British University in Egypt for providing us with the opportunity and resources to carry out this research successfully.

Finally, we would like to thank our families for their unwavering support and encouragement throughout our academic journey. Their patience and belief in us have been the foundation of our achievements.

Abstract

The current thesis outlines the development, design, and experimentation of IRIS, an emotionally intelligent social robotic assistant aimed at enhancing human interaction via advanced AI algorithms and robotics techniques. Furthermore, IRIS was developed to offer functional and emotional support for senior citizens and children with autism spectrum disorder (ASD). The system has various advantageous and advanced features such as speech-to-text (STT), chatbot conversation, sentiment analysis, text-to-speech (TTS), and facial expression output along with real-time forecast information updates. A mechanistic module organization and robust electronic design ought to support mobility and autonomous function empowered by SLAM, localization, and path planning via ROS. Therefore, emotion recognition and visual cues are displayed on a dynamic LED user interface to mimic human-like empathy. Finally, experimental trials and simulations took place to assure that the robot could navigate, interact, and respond in an appropriate manner to user emotion. The research incorporates an important step toward intuitive, emotionally intelligent human-robot interaction systems.

Keywords: Companion Robot, Emotion Recognition, Human-Robot Interaction, Autonomous Navigation, ROS, Sentiment Analysis, SLAM.

Table of Contents

List of Figures

List of Tables

Nomenclature

Symbol	Definition	Unit
v	Linear velocity of the robot	m/s
ω	Angular velocity of the robot	rad/s
R	Radius of the wheel	m
L	Distance between wheels (wheelbase)	m
θ	Orientation angle of the robot	radians or deg
x,y	Robot position in global coordinates	m
t	Time	s
I	Moment of Inertia	$kg \cdot m^2$
τ	Torque	N·m
F	Force	N
M	Mass	kg
a	Acceleration	m/s^2
d	Distance traveled	m
Δt	Time interval	s

Abbreviations

Abbreviation	Full Term
AI	Artificial Intelligence
ASD	Autism Spectrum Disorder
AMCL	Adaptive Monte Carlo Localization
ASR	Automatic Speech Recognition
CPU	Central Processing Unit
DHT11	Digital Humidity and Temperature Sensor
FEA	Finite Element Analysis
GUI	Graphical User Interface
HRI	Human-Robot Interaction
IMU	Inertial Measurement Unit
IoT	Internet of Things
LED	Light Emitting Diode
LiDAR	Light Detection and Ranging
NLP	Natural Language Processing
OLED	Organic Light Emitting Diode
PCA9685	PWM Servo Driver IC
PID	Proportional–Integral–Derivative Controller
ROS	Robot Operating System
SLAM	Simultaneous Localization and Mapping
STT	Speech-to-Text
TTS	Text-to-Speech
URDF	Unified Robot Description Format

Chapter 1 : Introduction

1.1. Background

Various developmental impacts in civilization, society, and industry were proven to occur due to the presence of a science such as robotics. Various aspects were studied to standardize the impact of robotics in multiple fields such as economy, healthcare, employment, education, and even ethics. As robotics is a field which offers enormous diversities of applications in various types of fields; therefore, improvements in quality of goods and services were increased, due to the boost of the automation and robotic systems' efficiency, productivity, and accuracy. For illustration, it inherits the fundamentals of several engineering fields, such as mechanical, computer, and electrical engineering; as a result, robotics offers different techniques and engineering technologies to develop sufficient automated or semi-automated systems.

In the time being, whenever robotics is mentioned, most likely the initial robotic system that crosses minds is robots. Robots are autonomous machines which include the capabilities of sensing their surrounding environments, compute the given and collected data to make decisions and perform specific tasks. Consequently, robots comprise distinct classifications relying on three major matters as function, design, and autonomy; moreover, such classifications identify the optimal type of robot to be implemented in the required, as they are considered the guidance for the purchaser.

Accordingly, each classification consists of multiple types; for example, autonomy, the design of the robot could indicate its working environment, as the robot could be utilized on ground, underwater, and fly, as illustrated in figure 1 below. Moreover, the design of the robot defines the functionalities and capabilities of the robot. Furthermore, different autonomy levels could be obtained, which would define the robot's ability to adapt to the surrounding environment and perform the task. Henceforth, due to the massive diversities of each classification, it could be illustrated that robots could be employed in multiple fields

such as medical and healthcare, industry, home appliances, security and surveillance, military and law enforcements, and research domain.

To emphasize the diversities of robotic systems, various applications introduced to illustrate the accommodation of such technologies, as robots could be employed in multiple domains and industrial fields, such as:

- **Industrial Robots and robotic arms:** Remotely controlled manipulators, actuators, and control systems for different tasks; for example, painting, pick and place, and spot welding.
- **Home monitoring:** To monitor energy consumption and the security state of the property; for example, Amazon Astro.
- **Military and law enforcements:** Surveillance and reconnaissance missions have highly relied on robotics in the meantime; therefore, improving the soldiers' mobility in the battle fields.
- **Aerospace:** Robotics could not only serve in drilling, painting, coating, inspection, and maintenance of aircraft components, but could aid in the control system enhancements of the flights' experiences.
- **Bioengineering and healthcare:** Multiple examples could illustrate the importance of robotics in such domains, as surgical robots, assistive robots, lab robots, and

telemedicine robots have proven great success in the present days; for example, the surgical robot Da Vinci.

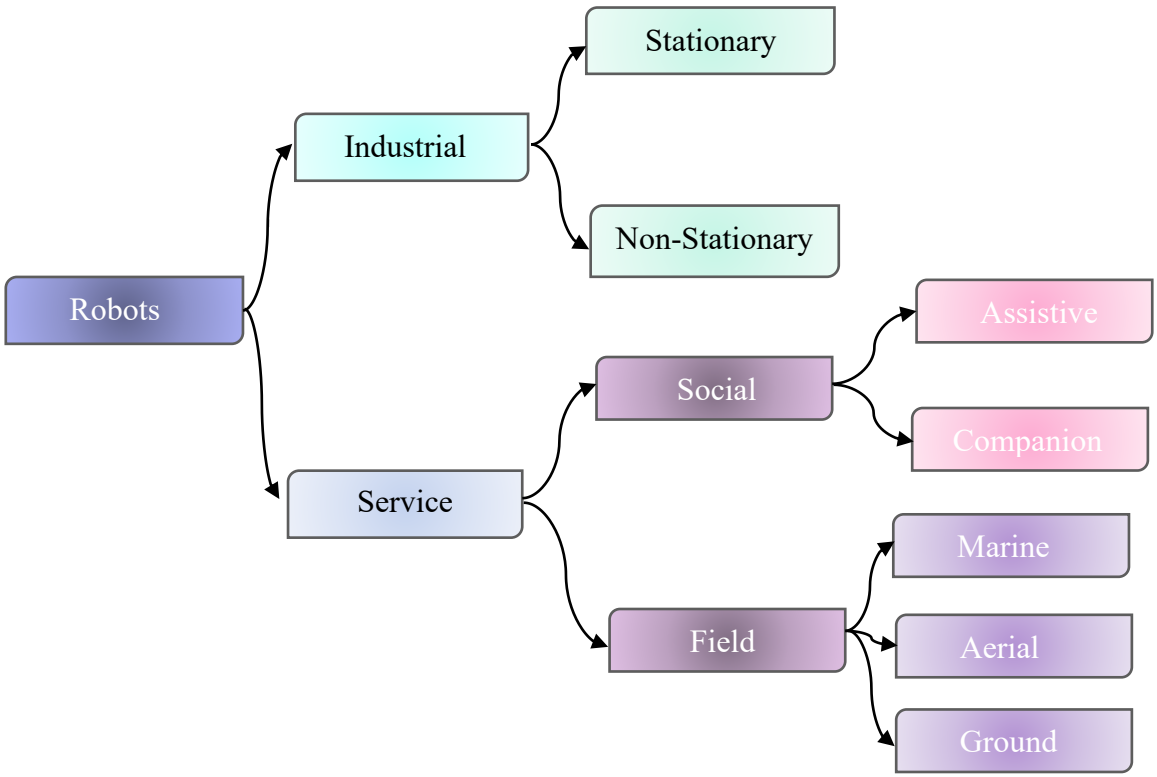


Figure 1 Types of Robots

As shown in figure 1 above, it could be illustrated that robots have two main types, which are the industrial and service robots. Briefly, industrial robots are the type of robots which could be found in factories and production lines; consequently, they could be classified into stationary robots such as robotic arms and the non-stationary robots as the transportation robots. Furthermore, social robots are classified into two main aspects, which are the field robots and the social robots. Field robots differ according to their locomotion type; particularly, locomotion is the ability of the robots to move depending on bio-inspired models and artificial designs. Therefore, as could be illustrated in figure 1 above, robots could fly, swim, and walk, such as drones, underwater robots, and humanoids, respectively.

Regardless, in the following report, social companion robots are to be discussed and developed. Social robots are developed to ease the lives of humans. For example, service robots can be found everywhere at the present time; such robots can be found at households, restaurants, airports, and even hospitals. Briefly, such types of robots are employed to be able

to communicate with humans and create some interactions between the robots and the surrounding environments.

Consequently, a type of service robot has gained great popularity and success rates at the present times, which is social robots. Social robots are robots usually developed to interact with human and different environments, static environments or dynamic environments. To demonstrate, social robots satisfy the need in various domains and applications, such as telepresence robots in the medical field, tourists' guide in the entertainment field, waiters and waitresses in restaurants, and many more types and applications. Social robots do not usually maintain a specified shape or features, as they differ from each individual application and surrounding environment. Therefore, in the following report, developing a social robot would be discussed, such as developing a smart system to be able to interact with the surrounding individuals and environments.

1.2. Aim and Objectives

In this section, we define what the overall goal of the project is and what specific objectives will both inform and guide the building of a social companion robot that exhibits an autonomous appearance. The goal describes the big picture while the objectives are more detailed and specific to formulating an overall view of how we are to achieve the goal.

1.2.1. Aim of the project

The aim of the project is to design and develop an autonomous social companion robot which has the capabilities of meaningfully interacting with the users with dynamic facial expressions which appear on a screen. The intention in designing and developing this companion robot is to be able to assist users in a friendly and interactively way, it offers emotional support and communication without the need of constant human supervision by implementing features like autonomy, speech-to-text (STT), chatbot (text or response generation), text-to-speech (TTS), sentiment analysis and real-time weather updated. The robot aims to achieve high-level of interacting with users.

The project addresses applications in companionship, education, healthcare, and customer services. For example, the robot can be used in providing support to individuals who live alone, helping students learning new skills and languages or provide interactive guidance in public places etc. museums. As mentioned before the project aims to apply autonomy for the robot which helps the robot to act as an individually and independently robot as it responds to user inputs without the need of human intervention. Through the integration of sentiment analysis,

the robot is capable of determining the user sentiment and respond with the suitable text and facial expression response leading to demonstrating empathy with the user which makes the robot one of a kind.

The project aims to confront and treat the challenges involved in creating autonomous, adaptable and intuitive interactions between the robot and the users. By exploiting and utilizing advanced AI and robotics technologies, the project success depends on achieving natural, intelligent and responsive relationship between the robot and the user which make it a significant advancement in the field of social robots.

1.2.2. Objectives of the project

In order to achieve the main aim of the project, these objectives have been outlined.

- **Implement an interaction system:**

    1. Utilize a Speech-to-text (STT) system: Utilizing an effective and accurate speech-to-text system, used to convert the input speech into a text-based words to be analysed later by the robot.

    2. Integrate sentiment analysis: integrating sentiment analysis to the project helps the robot identify the user sentiment and respond with the suitable face expression based on the tone of the user input text.

    3. Implement a chatbot model: with the use of reliable chatbot model the robot will be able to respond with a suitable text response based on the input text-based words.

4. Integrate a Text-to-speech (TTS) system: integrating a TTS model to make the robot capable of converting the generated response into speech which helps in achieving seamless communication between the robot and the user.

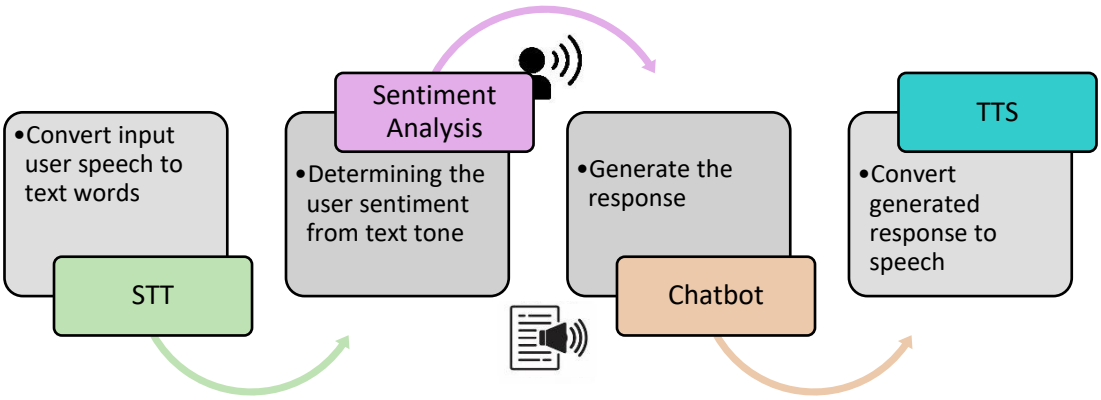


Figure 2 Interaction Process

- **Implement real-time weather information retrieval:** the robot will be able to access to weather information and updates based on user input question about the weather in a certain city.
- **Implement a dynamic screen:** the robot will contain a dynamic screen which is used to display the facial expressions with animations.
- **Develop the mechanical design of the robot:** designing and developing the mechanical design of the robot based on shape, base and materials.

    1. Develop the internal design of the robot with taking into account wiring holes and components locations
    2. Integrate appropriate wheels that will withstand the load

3. Design the outer chassis of the robot body.

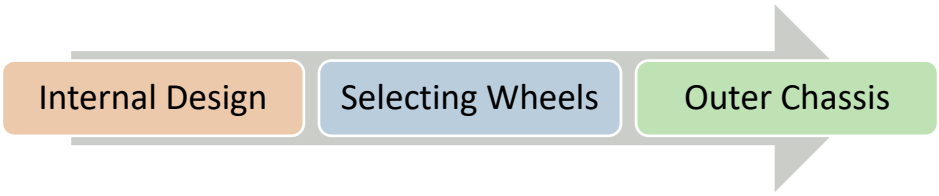


Figure 3 Design Steps

- **Achieve autonomy:** the robot will operate autonomously (moving, decision making, interactions) with no need of human intervention leading to a fully autonomous social companion robot.

  1. Robot Simulation in virtual environments: the robot is simulated in virtual environments for testing, allowing safer development.

  2. Mapping: the robot generates a map using different sensors data.

  3. Localization: the robot is capable of identifying its position relative to a global frame.

  4. SLAM: the robot will simultaneously update the map and localize itself inside the predefined map.

  5. Navigation: the robot will be able to navigate from one point to another inside the map.

Figure 4 Autonomy Steps

- **Integrate hardware:** integrate hardware components like sensors and actuators.

1.3. Hardware Design and Integration

In order to create a cohesive and functional emotionally intelligent companion robot, the design methodology combines hardware design, software integration, and advanced algorithm development. Therefore, such a design methodology provides a responsive and engaging experience to the user as well as fulfilling the project's objectives of emotional intelligence and autonomous navigation by ensuring that all components work together seamlessly.

1.3.1. Hardware Design and Integration

In this section the design and the integration of the robot's physical components are outlined, which form the foundation of the robot's functionality. To ensure a robust and reliable performance, the hardware is categorized into mechanical, electronic or electrical, and sensory systems.

Mechanical

The mechanical structure for the robot is designed for stability and efficiency which is the focus of the mechanical design. The main elements of the mechanical structure include but not explicit to:

- **Chassis:** The structural basis to which all components are mounted, the chassis is therefore must be designed for stability and durability.
- **Motors:** Precisely controlled motors for the movement to be smooth and accurate.
- **Internal Design:** To ensure efficient use of space, an optimized arrangement of the internal components is critical and taken into account during the mechanical design process.
- **Outer Body:** To enhance the user-friendliness and visual appeal of the robot and aesthetic and ergonomic design must be implemented in the design of the outer body.
- **Head Design:** Through which emotional expressions will be displayed and the main channel for user interaction, as it houses communication peripherals such as the OLED screen and microphones.

Additional features include considerations for joint flexibility and modularity for future upgrades.

Electrical/Electronic Systems

The robot's operational logic and power distribution is managed by the electrical and electronic systems. The main components of these systems are:

- **Central Processing Unit (CPU):** The main processor of the robot acting as the brain, receiving and sending all main commands. For example, the CPU manages AI services such as NLP and vision processing. This is in addition to several autonomous driving tasks such as localization, path planning, and obstacle avoidance are also managed by the CPU.
- **Peripheral Microcontroller:** Communication between various sensors, actuators, and the main processor is coordinated by the peripheral microcontroller.
- **Motor Drives:** Precise control by the motor drives to achieve smooth and accurate movement of the companion robot.

Sensors

The sensors systems implemented in the robot enables environmental perception and user-friendly interaction. The main sensors include but not explicit to:

- **Light Detection and Ranging Sensor (LIDAR):** The environment is mapped using the LIDAR sensor which is also integral for obstacles detection and avoidance.
- **Inertial Measurement Unit (IMU):** Motion data and orientation of the robot are provided by the IMU to aid robot to localize itself accurately.
- **Encoders:** Measuring the motor's rotations, odometry data can be obtained which will assist the robot in its autonomous driving tasks.
- **Microphone:** Captures user speech for interaction and voice communication with the user.
- **Speaker:** Outputs audio responses to improve communication.

Display Systems

The display system is mainly embodied in the OLED screens which is the core of the emotional expressions in the companion robot. As emotional connection is integral feature in the

companion robot, OLED screens display feelings such as happiness, sadness, or surprise as facial gestures.

Power Systems

The power management system is designed for reliability and efficiency since sustained and continuous operation of the robot without any disruptions is critical for building of stable companion robot. Therefore, the major power sources for the companion robot are LiPo batteries and a power bank. LiPo batteries power the motor, while the power bank feeds the central processor unit and sensors that will work continuously.

The whole system of the companion robot hardware design and its integration is illustrated in the figure 5 below

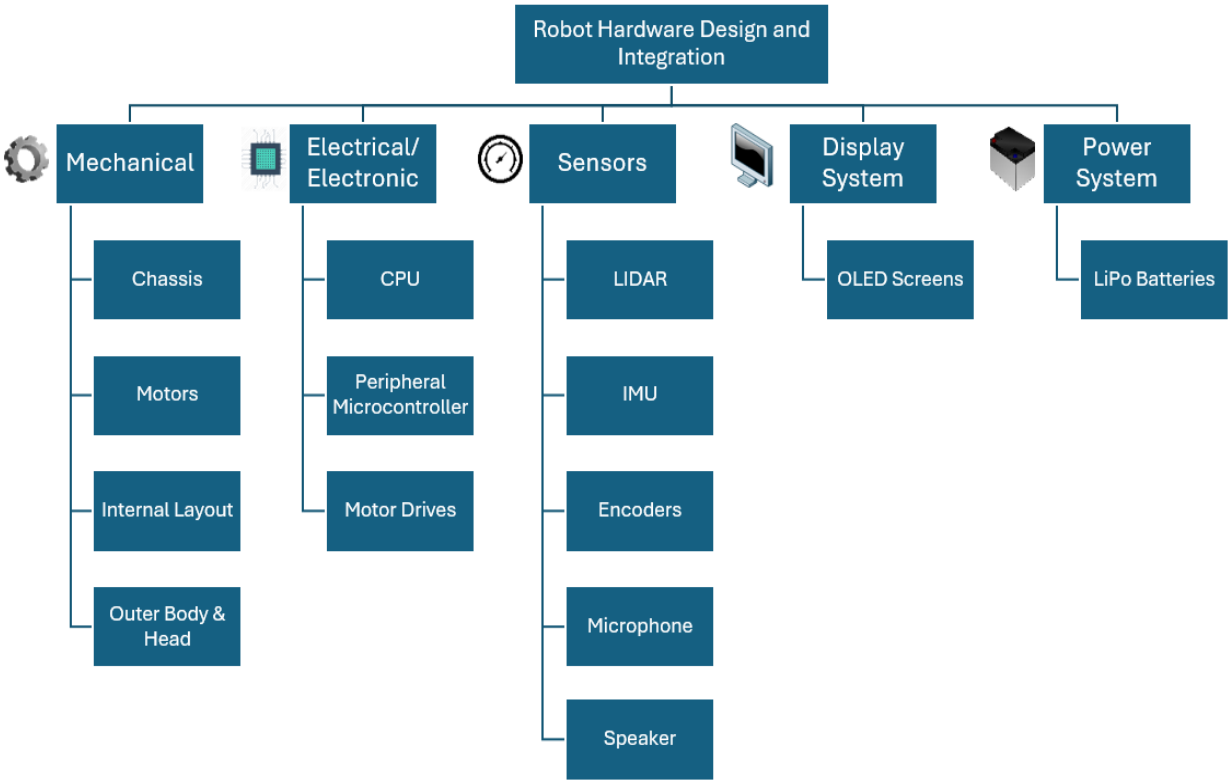


Figure 5 Robot hardware design and components integration.

1.3.2. Software Design and Algorithms

To design an intelligent and autonomous companion robot, the software framework is designed to integrate robotic operating system (ROS) with AI algorithms, thereby exhibiting intelligent behaviour. This section discusses the software design methodology by employing both ROS system coordination and artificial intelligence as show in figure 6 for interaction using the previously mentioned sensors for real-time perception, navigation and emotional engagement.

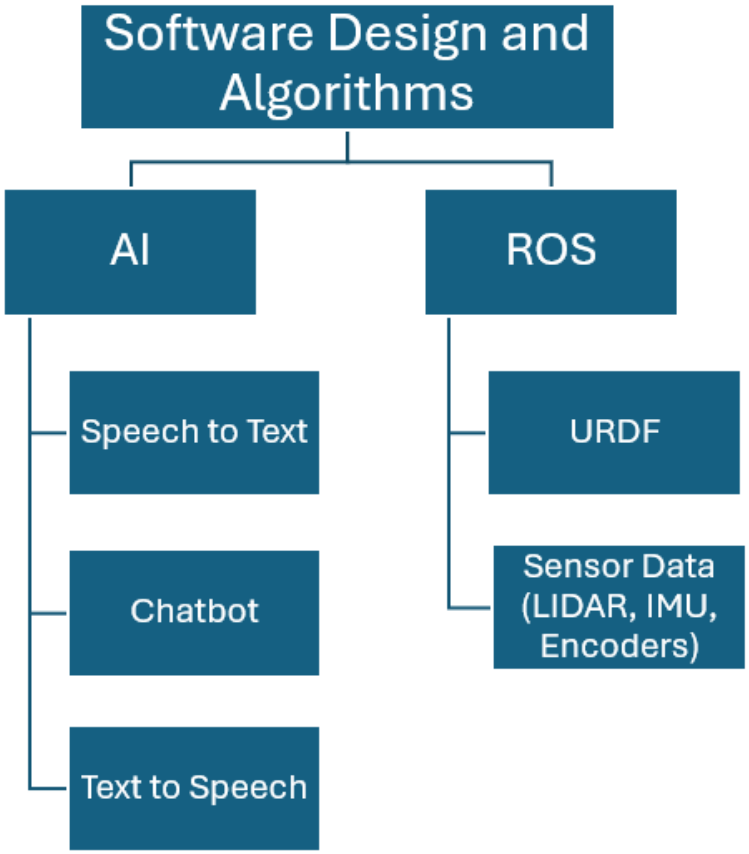


Figure 6 - Diagram of software design and algorithms.

Robotic Operating System (ROS)

The backbone of the robot's software framework is ROS, as it facilitates the interaction between hardware and software for seamless communication. The key subtasks include:

- **Robot Modeling**: Creating visualization and modelling for the companion robot using XACRO files for simulation purposes.
- **Sensor Data Fusion**: Data from various sensors are integrated together for effective autonomous driving and interaction.
    - **Encoders Data (Odometry):** Motor rotations are measured by encoders and processed with the peripheral microcontroller to be used for precise motion tracking.
    - **IMU Data:** Data such as orientation, acceleration and angular velocities of the companion robot are processed for accurate localization and navigation tasks.
    - **LIDAR Data:** For efficient path planning and obstacle avoidance, LIDAR data is used for environment map generation and obstacle detection.

Simulation Validation Phase: To ensure that components function as intended before hardware deployment, the simulation phase tests and validates sensor integration and algorithms in virtual environment. The following are the key tasks for the simulation:

- **Teleoperated Control:** Simulates joystick/keyboard operation to verify manual control mechanisms.
- **SLAM:** Simulates both mapping of environment and localization accuracy.
- **Adaptive Monte Carlo Localization (AMCL)**: Testing localization and navigation capabilities.
- **Navigation Simulation**: It validates the path planning and execution capability of the robot in a controlled virtual space.

Hardware Implementation Phase: Hardware deployment transitions components tested and validated during the simulation phase into physical hardware operating under real conditions for the purpose of experimental deployment. Main tasks of the hardware implementation phase are listed below:

- **Teleoperation Control:** The robot can be controlled directly and remotely using a keyboard or a joystick for immediate human intervention and easier testing purposes.
- **SLAM (Gmapping):** Real-world mapping and localization to navigate physical spaces.
- **AMCL:** Physical testing of the robot's localization system.
- **Navigation:** To execute real-time path planning and movement, integrating voice commands and chatbot functionality for enhanced interactivity.

The key subtasks of the ROS framework can be summarized in figure 7.

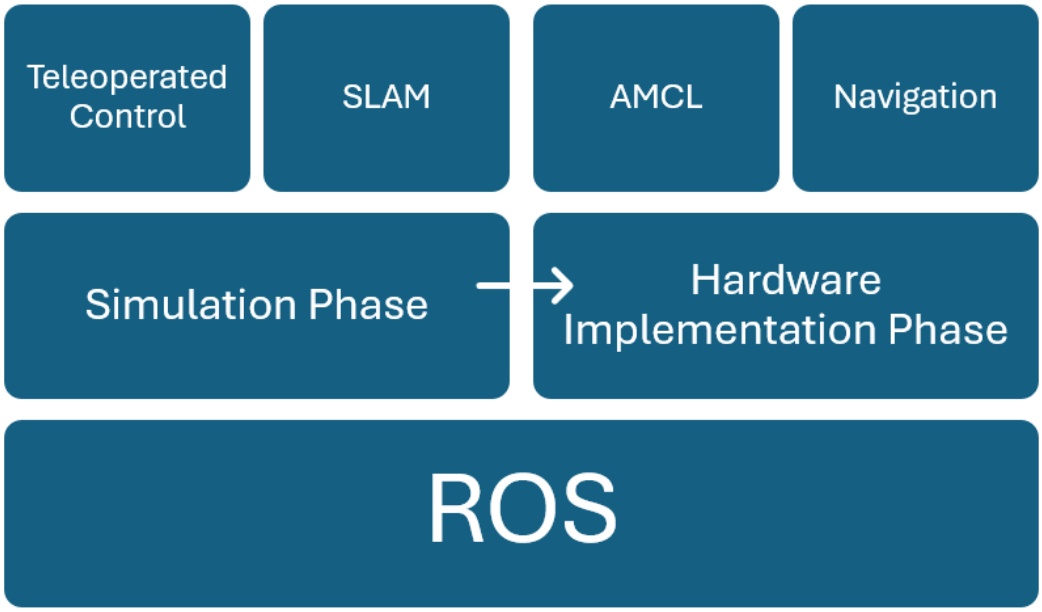


Figure 7 - ROS framework of the companion robot

Artificial Intelligence (AI)

AI algorithms are central to the robot capabilities of intelligence and emotional engagement with the user to provide optimal human-robot interaction. Main subtasks of AI include:

- **Speech to Text:** Transcription of user speech into text inputs for further processing is the very first and foremost step of the NLP system.
- **Chatbot:** Communicates with the user using adaptive learning algorithms to learn the personality of the user, thus providing an interactive and personalized engagement.
- **Text to Speech:** The generated text responses from the chatbot are converted to natural-sounding audio responses, thus enhancing the robot's communicative abilities.

1.3.3. Real-World Testing

Real-world testing will ensure that the robot operates efficiently and meets its users' necessities and expectations. Testing stages include:

- **User Interaction Trials:** Testing emotional engagement and communication with elderly users and children suffering from Autism Spectrum Disorder (ASD).
- **Navigation Testing:** Validates the robot’s ability to move autonomously in homes and indoor spaces.
- **System Integration Testing:** Ensures that the various hardware and software components function seamlessly.

This structured design methodology ensures that IRIS is a reliable, emotionally intelligent companion robot capable of providing meaningful interactions and robust support for its users.

1.4 Report Outline

This report has a structure designed to properly document the design, development, implementation, and evaluation of the IRIS social companion robot. Each chapter builds logically upon the previous one, in terms of theory, engineering workflows, experimentations, and discussions.

Chapter 2: Literature Review provided a survey of the state of the art in companion robotics, enabling technologies, retrospective, subsystems, and emerging technologies for the development of IRIS.

Chapter 3: Mechanical Design presented the complete conceptual and detail design of the physical structure of Robinson's robot. This encompasses the stress analysis and finite element analysis (FEA) of main components, as well as the rationale for material selections and the fabrication of components.

Chapter 4: Electric and Electronic Design addresses the system-wide electrical architecture, the subsystem integration, and signal distribution of the robot for actuation, sensing, and control.

Chapter 5: Control System describes the mathematical modeling of the robot, the control algorithms to implement, and the implementation of odometry, kinematics, and logic systems, that ultimately drive the robot's autonomous movement and behavior.Chapter 6 - Navigational

and Autonomous Mobile Robot (AMR). This chapter completely discusses the robot's mobility stack, including coordinate transformations, SLAM, localization, and the random waypoint navigation that was completed in autonomous path planning in a mapped world applied through ROS frameworks.

Chapter 7 - Artificial Emotional Intelligence System. The chapter covers the AI modules that enable autonomous human-robot interaction. Some aspects of natural language processing, including speech-to-text, sentiment analysis, chatbot response generation, and facial expressions, are included.

Chapter 8 - Graphical User Interface (GUI). This chapter illustrates the develop of a user interface that allowed interactive control and feedback from the robot, enhancing accessibility and experience.

Chapter 9 - IRIS IoT Monitoring System. This chapter illustrates a cloud-integrated monitoring and alerting scene incorporated into the software to support real-time data collection and visualization, and report on the vehicle's status of health.

Chapter 10 - Experimental Work. This chapter provides a summary of the test and validation procedures, real-world aspects of this research project. It contains summaries of the simulation scenarios, the experimental conditions, and all of the notes that were taken at various times during the robot's development cycle.

Chapter 11 - Analysis and Discussion. This chapter discusses the experimental findings based on performance measures, creates relationships between design variables and outputs, and evaluates the resilience of the robot system as a whole.

Chapter 12 - Conclusions provides a concluding summary to the report. It summarizes the achievements of the project, contemplates the achievement of objectives, acknowledges the limitations current to the project, and proposes future development opportunities. This summary of how this report is structured provides a cohesive linkage from problem identification to project completion, and contributes to the technical depth of the report, while also aiding in the interdisciplinary connection.

Chapter 2 : Literature Review

Designing and developing a successful companion robot featured with advanced technologies could be inspired from previous work and functionalities in order to apply advancements to the current technologies. Therefore, previous companion and social robots have been studied to be able to develop the following companion robot for elders and children with autism. Subsequently, companion robots consist of various systems and technologies integrated to create a successful project.

2.1. Introduction

Companion robots are a leap issued for a bright and easier future. Companion robots are autonomous robots which tend to communicate with humans in an intuitive way; for instance, non-expert users at different ages ought to be able to communicate with robots. Therefore, as the development of robotics has covered an enormous roles in the field of household appliance, it has been encouraged to serve humans as to be considered as a friend, a guardian, or a smart assistant at home; as a result, these companion robots would be engaged in the social lives of individuals, enhancing different aspects as needed, such as health-care, educational functions, entertainment, or daily duties [1].

Consequently, in the following report, the robot is developed to become a companion robot for older adults and children with autism spectrum disorder (ASD), assisting them by improving the quality of their lives. Autistic children could face some difficulties in socializing with the external world and foreign people; however, they are highly intelligent when it comes to technology and could easily interact with robots. In addition to autistic children, an enormous issue was discussed, studies were held to predict the amount of older individuals by 2050; therefore, it was found that between the 2000 and the 2050 , people aged above 60 would be doubled [2]. Moreover, 50% of people aged above 75 years live by their own and only 17% could meet relatives and friends less than once a week [3]; thus having a companion robot ought to be beneficial in such cases to provide a better social and healthy life.

With the present intention of developing such an advantageous robot, studying and analyzing past and current companion robots' developments is essential to demonstrate the features occupied by companion robots; additionally, it helps in the investigation of the essential tasks and merits of the current companion robots and identification of the added features for better

end results. As an illustration, companion robots have different types, shapes, sizes, and characteristics; for example, companion robots could be designed as humanoids, autonomous robots, or desk-sized robots, as would be discussed later. Thus, studying and observing previous companion robot projects aids in developing an optimal version of a companion robot suitable to perform the task needed.

For companion robot development, multiple sub-systems are developed and integrated later to form a fully functional companion robot; particularly, these sub-systems are the AI algorithm, which is the interaction mentioned above, continuing with the chatbot, the emotion display, the mechanical design, and finally the autonomous localization and navigation system (SLAM). As a result, these mentioned systems would be discussed and classified according to their theme of function in section 2.2 below.

2.2. Companion Robot History and Sub-systems

2.2.1. Evolution of Companion Robots

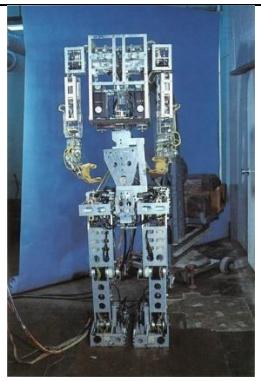
Companion robots have developed from simple mechanical devices to very complex, emotionally intelligent entities over the last decades. This was made possible by the latest developments in artificial intelligence, human-robot interaction, materials science, and actuators. In order to present innovation on this field's current day and to show which directions of future research would bring most value, the historical development has to be put into context.

a. Early Foundations: The Concept of Artificial Companions (Before 1990s)

Companion robotics emanates from early ideas of artificial intelligence and automation. During the 1960s-1980s, the design of robotic systems targeted industrial automation, where the core competence focused on repetitive task execution, rather than socially interactive tasks. Seminal projects, however, introduced core concepts that later would help mold and shape the area of social robotics which are mentioned in table 2.

Table 1 - Artificial companions before the 1990s.

Date	Name	Depiction	Description	Reference
1966	**Eliza Chatbot**		Among the pioneering AI-based conversational programs, the robot-like Eliza developed the first series of human-feigned responses-an early model, to say, of robotic social interaction.	[4]
1966-1972	**Shakey the Robot**		Developed in Stanford Research Institute International, Shakey was the very first mobile robot that could think about its actions. This developed the very foundation necessary for self-navigated autonomy in robots that would work as companions	[5]
1973	**Wabot-1**		Among the first humanoid robots, developed in Japan, featuring speech recognition and limb	

			movement—an early endeavor into interactive robotics	[6]
1984	**Wabot-2**		Improved model of Wabot-1, was able to read music sheets and perform with keyboard-type instruments. A very early prototype of multimodal interaction	[6]

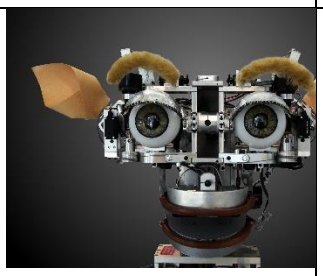
With these developments, however, technological limitations regarding underdeveloped AI, rigid materials, and limited computational power constrained meaningful human-robot interaction.

b. The Emergence of Social Robotics (1990s-2000s): Interaction Takes Shape

The 1990s were a turning point, as robotics moved from functional automation to socially interactive applications. During that time, for the first time, the appearance of true companion robots such as those in table 2, engaged users on an affective level.

Table 2 - Social robots between 1990s and 2000s.

Date	**Name**	**Depiction**	**Description**	**Reference**

1998	**Furby**		A mass-market robotic pet that could interact with users through emotional pre-programmed responses and hence became a popular item among such users.	[7]
1999-2006	**Sony AIBO**		The first robot pet in the market; AI behavior-controlled, learned by stimulus-response, and adapted to behavior changes.	[8]
1998-2001	**MIT Kismet**		Kismet was among the very first emotive robots that could detect and respond to human emotion through mimicry (Breazeal, 2003).	[9]

2001	**NeCoRo**		A robotic cat from Omron Corporation developed to react to the sense of touch, sound, and light.	[10]
2001	**PaPeRo**		One of the earliest home-based companion robots, developed by NEC for verbal communication and autonomous interaction.	[11]
2003	**QRIO**		This humanoid robot made and developed by Sony was built so that "dynamic gesture-based social interaction" may be dancing or playing soccer.	[12]
2005	**iCat**		A research-oriented emotionally expressive robotic cat with which the manufacturer Philips developed an environment for emotion-based interaction.	[13]

This period showed the feasibility of robots as interactive, social objects and introduced emotions and simple adaptive learning.

c. Humanoid and Emotional Robots' Development (2010s-Present)

The development of companion robots in the year 2010 was highly inspired by artificial intelligence, natural language processing, and emotions. All of which led to the abilities of human-like robots detecting human emotion, social interaction, and providing psychological support.

Anthropomorphic and Humanoid Companions:

The focus of humanoid companions developed to be advanced on robot behavior regarding enhanced expression and conversation such as those presented in table 3.

Table 3 - Anthropomorphic companions from 2010s till present.

Date	Name	Depiction	Description	Reference
2006-Present	**Nao**		Humanoid by SoftBank Robotics used in education, therapy, and research due to his social expressivity.	[14]

2014	**Pepper**		This humanoid robot, a social interactive machine equipped with identifying emotion capabilities, can easily empathize with human beings in interaction.	[15]
2016	**Sophia**		A truly humanoid robot created by Hanson Robotics that popularized itself with expressions that look very much real and conversations that are difficult to depart from.	[16]

Zoomorphic and Caricatured Companion Robots:

Animal and caricature robots also became very much in giving promises of a high possibility to produce emotional attachments and social comfort.

Table 4 - Zoomorphic companions from 2010s till present.

Date	**Name**	**Depiction**	**Description**	**Reference**

2004-Present	**Paro**		A robotic seal used in therapies for dementia and shown to lower stress and raise social interaction for elderly patients by 79%.	[17]
2017-2018	**Jibo**		A caricatured social robot optimized for in-home interaction based on nonverbal expression and vocal engagement.	[18]
2019	**LOVOT**		LOVOT (2019) – Groove X's cuddly emotional companion robot that is supposed to love touching and talking with humans.	[19]

The evolution of companions along the generations which is depicted in figure 8 has been fascinatingly rapid from simple mechanical structure to affective, and engaging social robots.

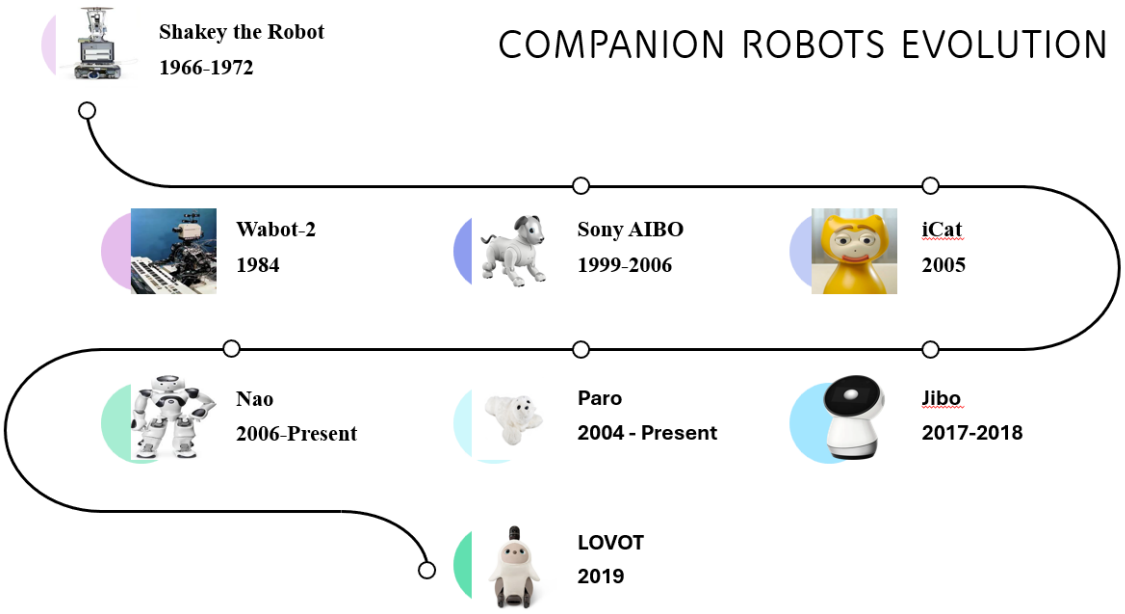


Figure 8 - Evolution of companion robots.

d. Future Directions: Beyond the 2020s

With time, next-generation companion robots will become due for change, and this transformation must now shift from a reactive inertness to an active one that stands for friendship and affection. The future pursuits toward this will include enhanced modalities of sensation, augmented context-awareness, and adaptive cooperation between human beings and robots.

Companion robots nowadays are gradually becoming more autonomous, emotionally aware, and user-adaptive, moving this time from passive/helper-like assistants to almost an active partner involved with the users. Their integration into everyday life will likely become seamless, driving deeper human-robot interaction.

The development of companionship robotics has come a long way from simple robotic pets to humanoid AI robots that are capable of very advanced social interaction. Within research, future developments will probably deal with the integration of higher-order cognitive modeling, reinforcement learning, and bio-inspired robotics in improving robotic companionship.

2.2.2. Fields of Companion Robots Applications

Companion robots can be applied in several domains. Empirical evidence substantiates successful applications in diverse sectors illustrated in figure 9 such as the health sector, education, and the social realm.

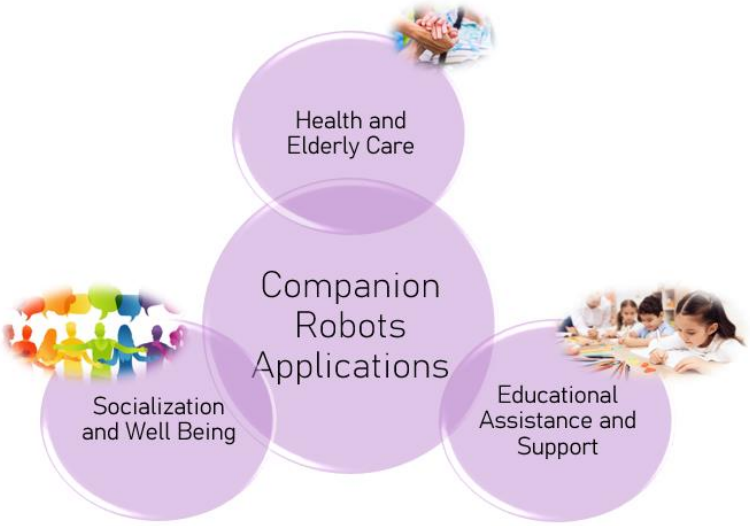


Figure 9 - Applications of Companion Robots

Health and Elderly Care: Companion robotics is one of the most significant aspects in the medical field, especially for cognitive therapy or mental health purposes.

- **Dementia and Cognitive Therapy:** Cognitive decline and dementia-related ailments are the major challenges for aged care. Companion robots are non-pharmacological interventions developed to raise mental stimulation, improving memory retention for patients who suffer from dementia, in addition to calming agitation. Paro was reported to permit a 79% increase in social interaction with people suffering from dementia [17] and has been reported to reduce agitation and improve mood. Pepper's story-telling ability seems to support memory retention [20].
- **Well-being Factors and Mental Health:** Psychotherapeutic companion robots have progressed to significant levels with respect to mental health efficacy, especially in stress, anxiety, and depressive symptom alleviation. Emotional support, improved social interactions, and overall psychological resilience are

advantages delivered by these robots. A review [21] indicates that 44% of the studies supported the efficacy of social or interactive robot-supported intervention programs in enhancing well-being. The primary advantages found are the alleviation of stress and improvement in mood. Nao and Pepper robots enhance social engagement in care facilities for the elderly by 46% [20]. These robots enhance cognitive function and lessen social isolation.

Such results now went on to indicate the strong indications that healthcare companion robots can bring enhancement possibilities to cognitive disability and mental health.

Educational Assistance and Support : Companion robots have gained much popularity in educational environments; thus, these robots are perceived as beneficial to students needing personalized assistance.

- **ASD Intervention:** People with ASD, including children and adults, find it difficult to interact with others and face communication issues. Companion robots have been developed for delivering structured and predictable interaction that enhances social interaction, emotion recognition, and learning performance. Nao has been linked to 67% greater engagement by ASD students with its structured interaction; the reason is that the robot captivates neurodiverse individuals through the predictability of social cues [22].

- **General Education:** Companion robots have been helpful in fields of education whereby it increases the learning engagement of students, improves knowledge retention, and provides a new way of learning through them. Such assistance and real-time feedback from them provide much importance to education as a valuable educational tool. Robot-assisted learning improves retention rates by 42% compared to traditional methods [23]. Robots such as Pepper facilitate interactive lessons and reinforce subject comprehension.

These cases are validation for the learning capabilities in the area of companion robots, which can help enhance interest among neurodiverse learners and retention in educational situations

Socialization and Well-being: Companion robots are social facilitators, acting against loneliness while promoting the quality of interaction.

- **Emotionally Companionable**: Pandemic-induced loneliness is a social issue; one growing increasingly disruptive in our aging societies. Companion robots are social

robots that can offer comfort and companionship to bring about meaningful interaction. Companion robots reduce loneliness by 46% in older adult users [17] Their presence alleviates social isolation, particularly in institutionalized care. Robotic pets enhance psychological well-being [24]. Non-humanoid robotic animals, such as aibo, tend to give users very similar companionship they can get from pets.

- **Social Interaction and Presence:** Companion robots are designed to interact with the user in a manner that may allow for social connectedness by means of dynamic interaction. Capabilities for eye contact, response to verbal cues, or other social behaviors greatly enhance perceived presence and engagement made possible by these robots. Casper enhances user engagement, achieving a 100% task compliance rate compared to 80% for tablet-based interfaces [25]. Social robots promote greater interaction levels.

Empirical evidence confirms the place of companion robots in social interaction, with expectations of a future of robotically mediated change from isolation to activity

2.2.3. Design Review on Automatic Speech Recognition (ASR)

Speech recognition or Automatic speech recognition (ASR) systems are essential for designing a companion robot, it is considered as the first step for implementing an interaction system between human and robot using speech. ASR systems use the users' input speech and convert it to text, the text is then processed by the robot and use it to generate an appropriate response. Engineers call this process Speech-to-text process (STT). Speech-to-text models have been evolved and developed over the years.

In September 2018 a research paper [26] was published which compares between two different automatic speech recognition systems which are Sphinx and Google Speech Recognition. The paper involves determining word error rate (WER) of each model and evaluating each. The research paper also compared each model at control and dysarthric speech.

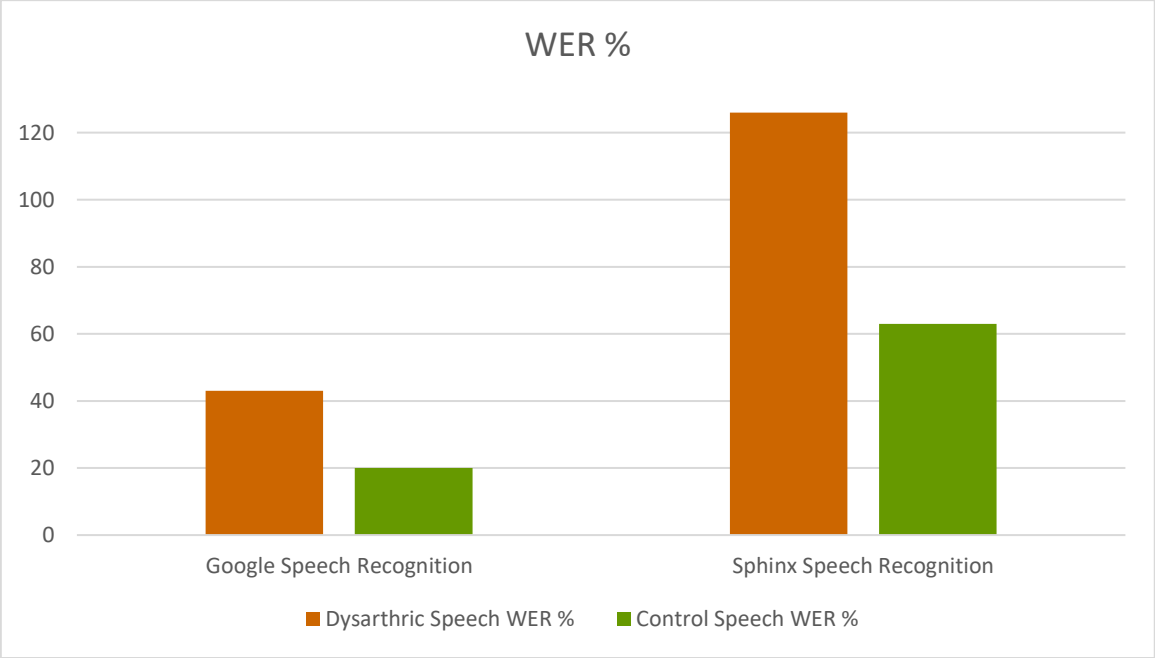


Figure 10 Average WER % for Paper [1] Models

Table 5 Paper [1] Models Performances Summary

Model	**WER % (dysarthric speech)**	**WER % (control speech)**
Google Speech Recognition	Average WER (43%)	Average WER (20%)
Sphinx Speech Recognition	Average WER (126%)	Average WER (63%)

In August 2023 paper [27] was conducted by a group of researchers, their aim was to compare existing open source speech recognition models and selecting the best one to implement it in a robot for human-robot interaction. The paper focuses only on open-source non-paid models. The paper evaluated five different models which are Google Speech Recognition, VOSK, Sphinx, DeepSpeech and Whisper AI based on WER% and execution time of each. The paper also highlighted that speech recognition systems have challenges such as background noisy environments and the childish voices which affect the accuracy of the system. The study also seeks to contribute to developing human-robot interaction (HRI). The results showed that the average WER% of Google ASR is (42.3%) which is calculated from three speakers of WER score 22%, 86% and 19% with execution time approximately 33 seconds, Sphinx ASR with average WER% (92.6%) calculated from three speakers of WER score 117%, 58% and 103% with execution time approximately 57.6 seconds, DeepSpeech with average WER% (869%) calculated from three speakers of WER score 808%, 906% and 894% with execution time approximately 8.4 seconds but note that DeepSpeech model stopped before the sentence is completely recognized with the three speakers. VOSK model with average WER% (51.7%) calculated from three speakers of WER score 47%, 89% and 19% with execution time approximately 18.9 seconds, finally Whisper AI with average WER% (9%) calculated from three speakers of WER score 8%, 6% and 14% with execution time approximately 30.4

seconds. The results showed that Whisper AI has the best accuracy the Google ASR, Sphinx and DeepSpeech has high WER% which is considered non practical, VOSK is ranked as third after Whisper AI and Google ASR but not recommended and DeepSpeech is not taken into account as the model stopped while testing.

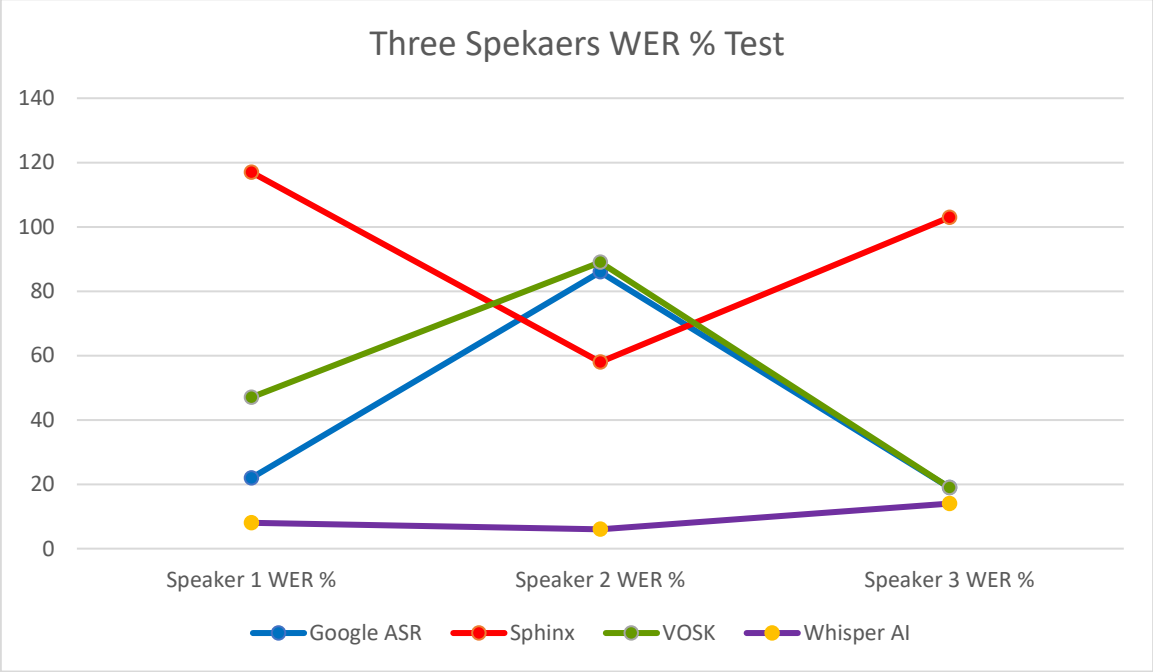


Figure 11 Three Speakers WER % Test for Each Model

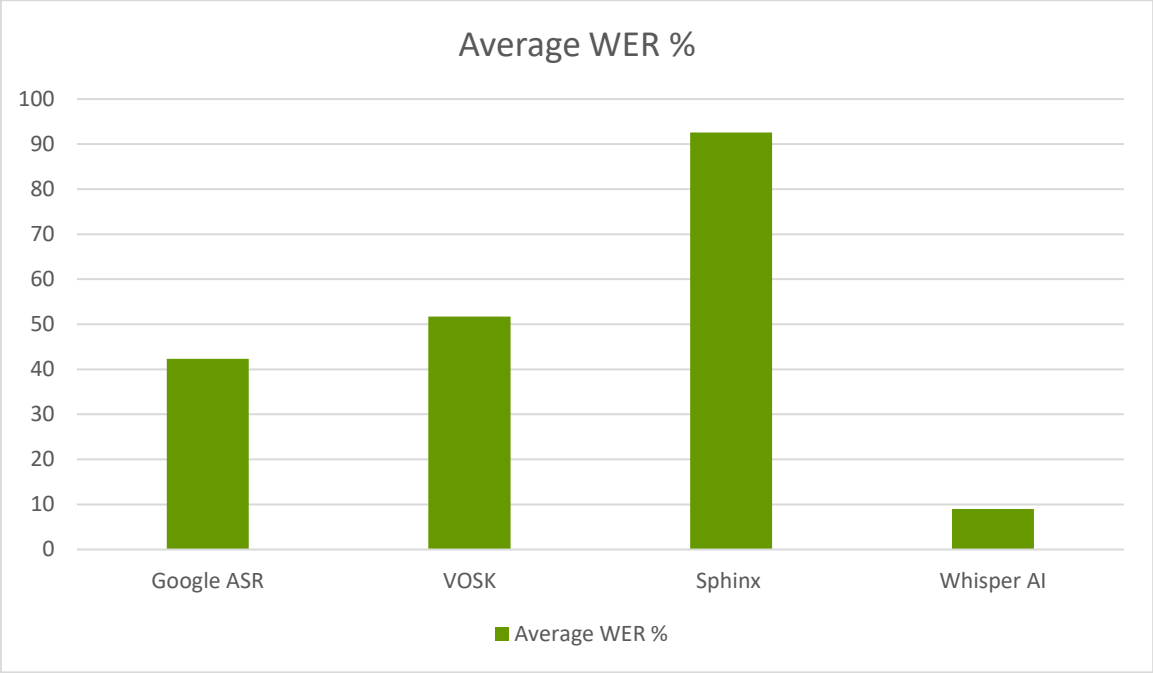


Figure 12 Average WER% for Paper [2] Models

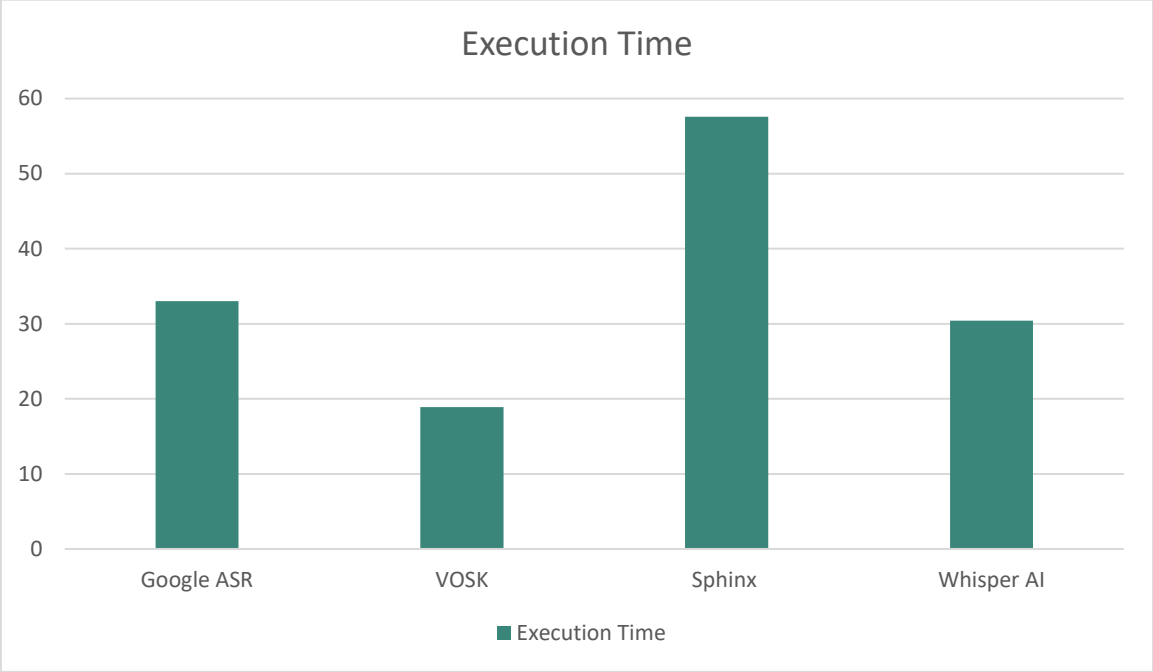


Figure 13 Execution Time for Paper [2] Models

Table 6 Paper [2] Models Performances Summary

Model	**Average WER% from Three Speakers**	**Execution Time (Approximately)**
Whisper AI	Best performance with 9% WER	30.4 seconds
Google ASR	Second Best with 42.3% WER	33 seconds

VOSK	Below Average Performance with 51.7% WER	18.9 seconds
Sphinx	Bad performance with 92.6% WER	57.6 seconds
DeepSpeech	Stopped before recognizing the full sentences with 869% WER	8.4 seconds
Summary	• Whisper AI best performance with high accuracy. • Google ASR comes after Whisper AI in the performance. • VOSK is below average. • Sphinx has bad performance. • DeepSpeech has stopped during recognizing.	

In October 2023 survey [28] was conducted to compare different speech-to-text models, and its aim was to deliver proper information about different speech-to-text models to use them effectively and improving quality. The survey focuses on STT models such as Google Cloud Speech-to-Text API, Meta wav2vec 2.0, NVIDIA NeMo and OpenAI Whisper. Each model was evaluated and rated based on its word error rate (WER). Before calculating WER for each model, the researchers have to remove punctuation and whitespace, converting numbers into words and converting text into lowercase to insure accurate WER results. The results showed that OpenAI Whisper had WER (5.6%), NVIDIA NeMo had WER (7.2%), Google Cloud API had WER (13.3%) and wav2vec had WER (21.1%), as shown in the results the lowest WER result was achieved by OpenAI Whisper and the highest was achieved by Meta wav2vec.

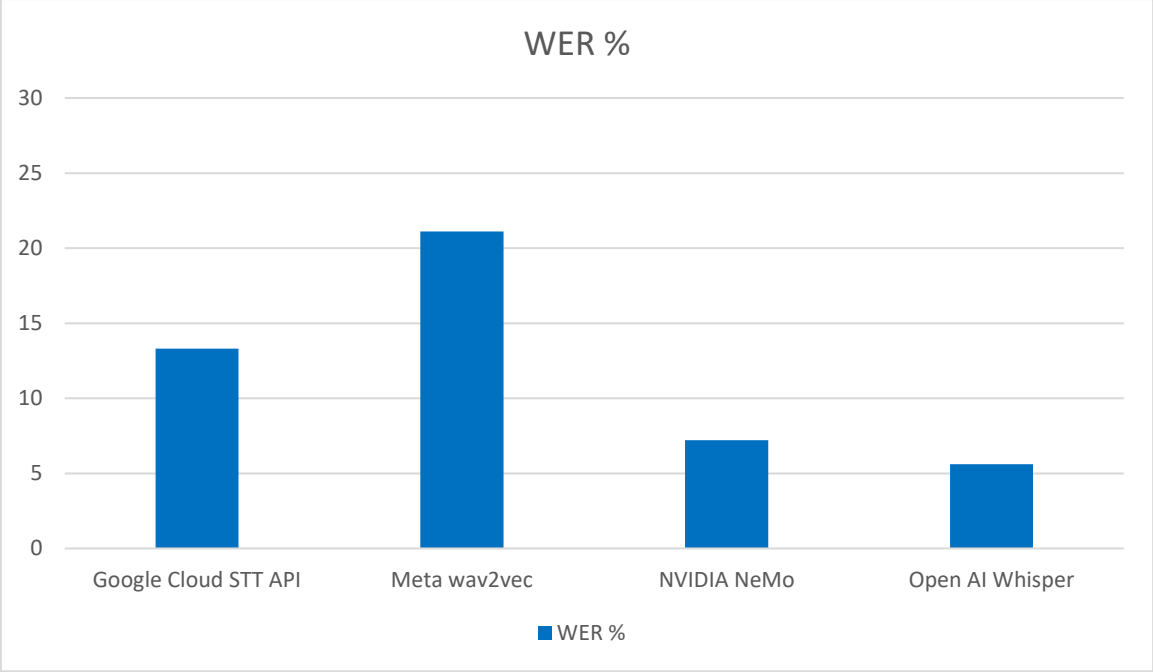


Figure 14 WER % for Paper [3] Models

Table 7 Paper [3] Models Performances Summary

	Google Cloud Speech-to-Text API	**Meta wav2vec 2.0**	**NVIDIA NeMo**	**OpenAI Whisper**
WER %	13.3%	21.1%	7.2%	5.6%

In February 2024 a group of researchers has discussed different approaches in paper [29] to try solving the challenges facing companion robots by integrating different speech-to-text models with Chat-GPT and evaluating their performances. They used Pepper robot, and their aim was to achieve a more natural and effective communication between human and Pepper robot and improving overall performance. They made a comparison between different STT models which are Whisper which is developed by OpenAI, Google automatic speech recognition (a supported python library) and Google cloud API, each approach was integrated with Chat-GPT-3.5-turbo model from OpenAI and was tested by 15 human users. Whisper model achieved the best

performance with average word error rate (WER) of (1.716%) and processing time (2.639 sec). 70% of human users who have tested Pepper robot were excited during the interaction process and 60% rated its performance as excellent and the rest rated it as good. Some challenges need to be tracked as well such as achieving multilingual ability. Whisper outperformed the other models due to achieving the lowest WER among non-English speakers such as Arabic and Filipino accents and requiring shortest processing time. Google ASR python library had average WER (20.63%) while Google cloud API had average WER (12.16%). The researchers integrated Whisper AI in their Pepper robot project and recommend it as it achieved the best performance. They also recommend Whisper Small model to balance between accuracy and processing time.

Figure 15 WER % for Paper [4] Models

Table 8 Paper [4] Models Performances Summary

	Whisper AI	**Google ASR python library**	**Google cloud API**
WER (%)	1.72%	20.63%	12.16%
Processing Time (seconds)	2.639	Not Mentioned	Not Mentioned
Recommendation	Recommended		

In March 2024 paper [30] was released in which several speech recognition models were evaluated, and the aim was to determine the most suitable model for offline, practical usage. The paper focuses on open-sources and offline models. The models evaluated were pocket Sphinx, VOSK and Deep Speech. The paper focuses on measuring each model WER in different noise environments and observe each model performance with evaluating each relative to the other models. The paper also compared the inference time, computational requirement and noise robustness of each.

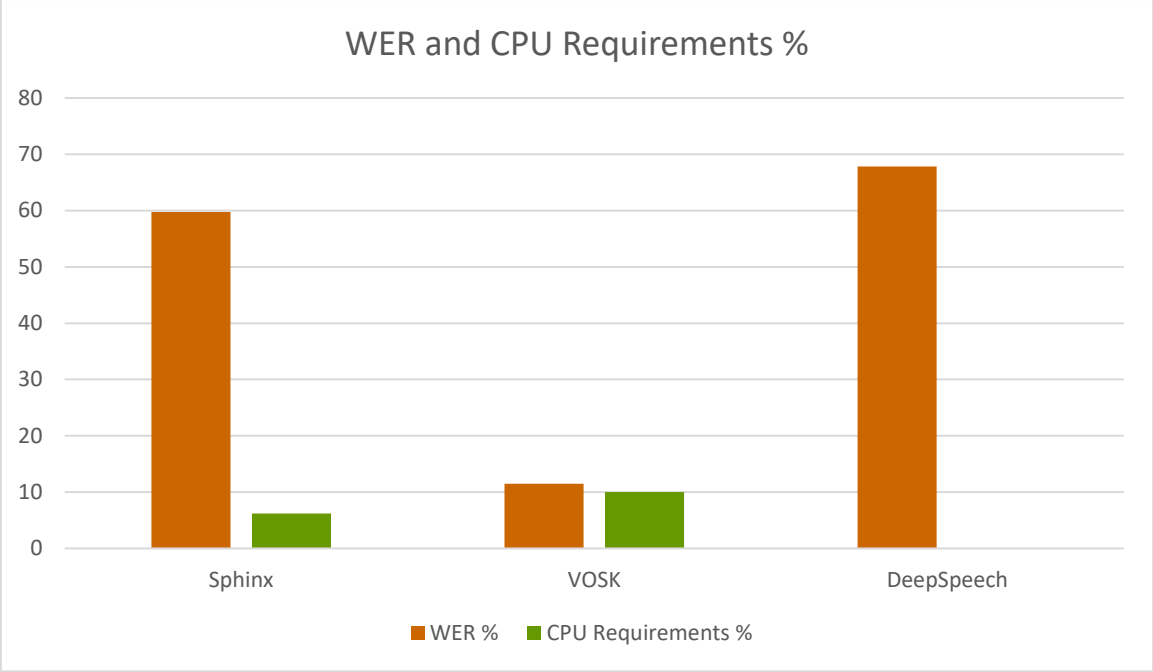


Figure 16 WER and CPU Requirements % for Paper [5] Models

Table 9 Paper [5] Models Performances Summary

Model	**WER %**	**Inference Time**	**Noise Robustness**	**Computational Requirements**
PocketSphinx	Tested on standard sentences achieving average WER (59.77%) so accuracy (40.23%)	No specific inference time was mentioned but PocketSphinx had a slightly lower inference time compared to VOSK and	As noise level increase the accuracy decrease significantly	6.2% CPU
VOSK	Tested on standard sentences achieving average WER (11.49%) so		Accuracy 93.33% in quite environments and 80% in	

	accuracy (88.51%)	the difference is negligible	communal environments	10 % CPU
DeepSpeech	Tested on standard sentences achieving average WER (67.82%) so accuracy (31.03%)	Not mentioned as it is unsuitable for research [30]		
Summary	• Results indicating VOSK has the lowest WER and highest accuracy • Pocket Sphinx has the lowest computational requirements • Both VOSK and Pocket Sphinx have low inference time • VOSK has the best accuracy in noisy environments			

2.2.4. Design Review on Chatbot

In December 2023 paper [31] compared Gemini Pro between different versions of ChatGPT such as Gemini Pro vs. GPT 3.5 Turbo and Gemini Pro vs. GPT 4 Turbo in various tasks like Knowledge-Based QA, Reasoning, Mathematics, Code Generation and Translation. In Knowledge-Based QA GPT 3.5 Turbo was more accurate than Gemini Pro and ChatGPT 4 Turbo was better than ChatGPT 3.5 Turbo, in Reasoning Chat GPT 4 Turbo was the best, Chat GPT 3.5 Turbo was the second best, in Mathematics Chat GPT 4 Turbo had the best performance then Chat GPT 3.5 Turbo, in code generation Chat GPT 4 Turbo also performed better than Chat GPT 3.5 Turbo and Chat GPT 3.5 Turbo was better than Gemini Pro especially when generating longer codes, and finally Gemini Pro outperformed both Chat GPT 4 Turbo and Chat GPT 3.5 Turbo in Translation tasks.

Table 10 Paper [6] Summary

Model	*Knowledge-Based QA*	*Reasoning*	*Mathematics*	*Code Generation*	*Translation*
Gemini Pro	Lowest	Lowest	Lowest	Lowest	Best Performance
Chat GPT 4 Turbo	Best Performance	Best Performance	Best Performance	Best Performance	Second Best
Chat GPT 3.5 Turbo	Second Best	Second Best	Second Best	Second Best	Lowest

In June 2024 review paper [32] compared between different chatbots which are ChatGPT-4, Microsoft Copilot and Google Gemini in question answering tasks (QA). The review aim was to evaluate each model based on accuracy and narrative coherence including logical reasoning (most prevalent feature), internal information (the response includes information from the input question) and external information (the response includes information outside of the input question) in the correct answer response while logical errors (the most common errors type made by chatbots), information errors and statistical errors (basic calculations errors) in the incorrect answer response. The results showed that the overall accuracy of the models is

considered high in predicting the correct response. ChatGPT-4 and Microsoft Copilot showed better accuracy than Google Gemini.

ChatGPT-4 in generating correct answer had logical reasoning (81.5%), internal information (18.5%) while in generating incorrect answer had logical errors (88.9%), information errors (8.9%) and statistical errors (2.2%).

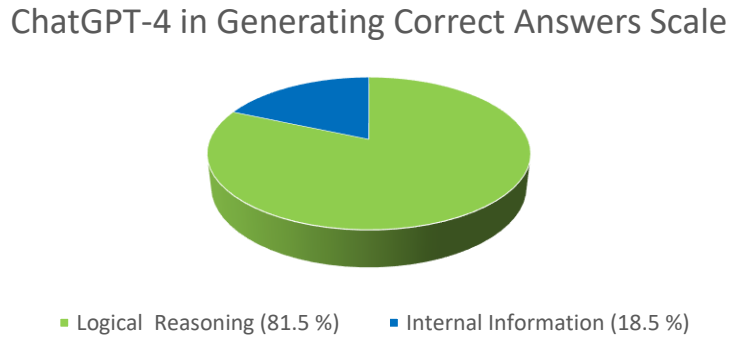


Figure 17 ChatGPT-4 in Generating Correct Answers Scale

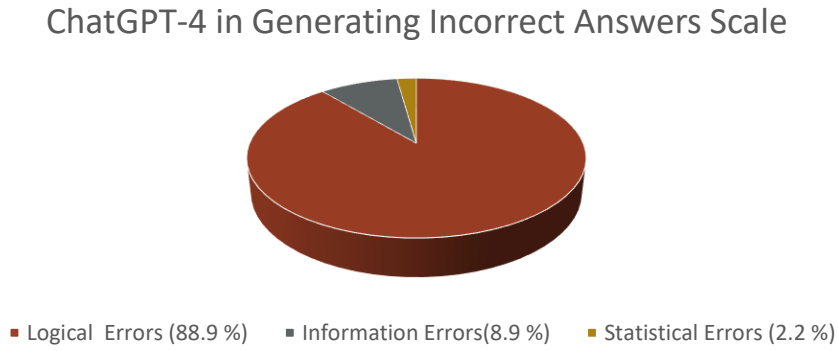


Figure 18 ChatGPT-4 in Generating Incorrect Answers Scale

Microsoft Copilot in generating correct answer had logical reasoning (55%), internal information (18.6%), external information (26.4%) while in generating incorrect answer had logical errors (79.1%), information errors (11.1%) and statistical errors (9.8%).

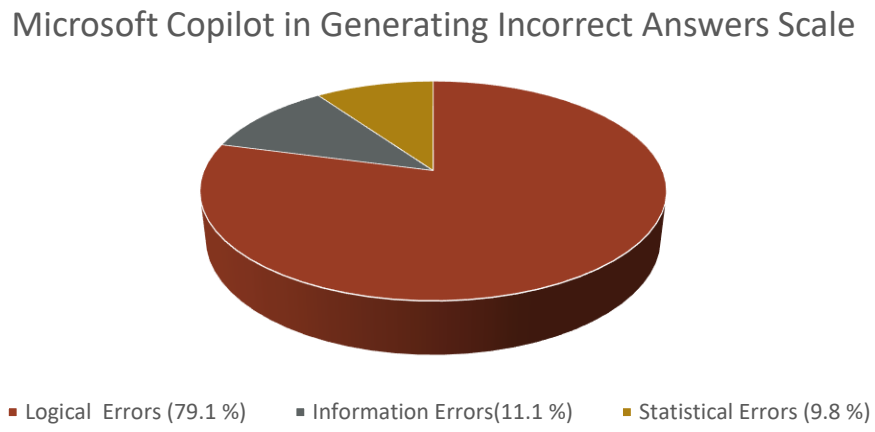


Figure 19 Microsoft Copilot in Generating Correct Answers Scale

Google Gemini in generating correct answer had logical reasoning (98.8%), internal information (1.2%) while in generating incorrect answer had logical errors (99.6%) and few information errors (0.4%).

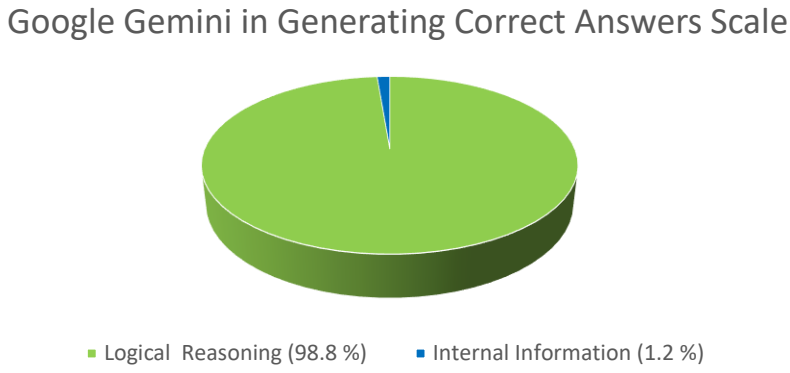


Figure 21 Google Gemini in Generating Correct Answers Scale

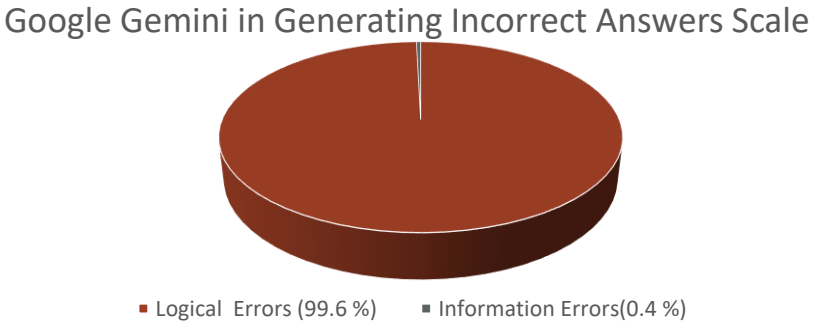


Figure 20 Google Gemini in Generating incorrect Answers Scale

Table 11 Models Performance[32]

	Generating Correct Answer			*Generating Incorrect Answer*		
Model	Logical Reasoning	Internal Information	External Information	Logical Errors	Information Errors	Statistical Errors
ChatGPT-4	81.5%	18.5%	___	88.9%	8.9%	2.2%
Microsoft Copilot	55%	18.6%	26.4%	79.1%	11.1%	9.8%
Google Gemini	98.8%	1.2%	___	99.6%	0.4%	___

In February 2025 paper [33] was published which compares between ChatGPT and Gemini based on their response to medical inquiries across various fields. The paper focuses on the accuracy and the response length and discusses the strength and limitations of each model. The results showed that ChatGPT had better overall accuracy (72.06%) and Gemini with overall accuracy (63.38%). ChatGPT response length was shorter than Gemini with response length 961 characters and 1423 characters respectively, the mean difference is 462 characters.

Table 12 Models Accuracy and Response Length [33]

Model	*Accuracy*	*Response Length*
ChatGPT	72.06%	961 characters
Gemini	63.38%	1423 characters
Summary	• ChatGPT accuracy is better than Gemini • ChatGPT response length is shorter than Gemini with mean difference 462 characters.	

Table 13 Performance Summary

Model	Merits
Chat GPT 3.5 Turbo	Good Accuracy in Knowledge-Based QA, Reasoning, Mathematics, Code Generation
Chat GPT 4 Turbo	Better performance than Chat GPT 3.5 Turbo in Knowledge-Based QA, Reasoning, Mathematics, Code Generation
Gemini Pro	Best Performance in Translation Tasks
Microsoft Copilot	Good Performance but the least

2.2.5. Emotion Display

As mentioned before, the following project aims to aid elders and children with ASD; therefore, the Human Robot Interface (HRI) ought to be simple and appealing. Consequently, the HRI ought to be implemented for the non-professional use, for the user to simply adapt to the system preventing a complication display. Moreover, it could be implemented through verbal and non-verbal communications by displaying emotions, gestures, or postures [2]. According to the following studies [2], [3], [34], [35], [36], by highlighting the importance of the HRI system

display, users could emotionally interact with the companion robot, creating a trust bond, which would ease the interaction for the consumer.

Multiple models of HRI could be applied to perceive and comply to the user's communication and commands, creating a more realistic and natural interaction between both human and companion robot [34], [36], [37], [38], [39]. For example, those models could be verbally or visually (non-verbally); thus, in the following section, visualising the robot's response ought to be discussed. Therefore, to visualize the emotions and facial expressions of a companion robot, various methods ought to be investigated and studies [36], [39], such as the following:

- **LED Arrays:** LED matrices are placed in particular places of the companion robot's face, such as eyebrows and mouth; subsequently, as illustrated in figure 23 below, this particular method could replicate specific (basic) emotions like happiness, sadness, anger, fear, disgust, and surprise [38]; for example, the companion robot NICO functions and interact using the LED Arrays method; for instance, NICO includes an 8x8 LED matrix for each eyebrow and an 8x16 matrix for the mouth ; however, this method could be relatively simple and slightly does not satisfy the natural influence of a companion robot on its consumer, it was inspired by the fMRI, functional Magnetic Resonance Imaging, studies which indicates that even the text-based response is based on the same response mechanism of the brain; for example, the companion robot NICO is trained using deep neural architectures.

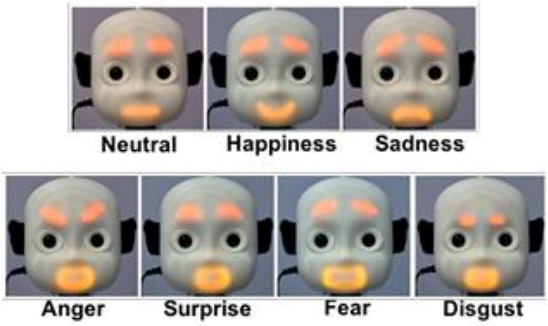


Figure 22 NICO's seven basic expressions [38].

- **LCD Screens:** in the present days, companion robots mostly rely on the LCD or LED screens to visualize the emotional expressions of the companion robot. Consequently, these emotional expressions are customized, based on the required designs along with some scientific lighting; for example, Misty II was designed based on the Plutchik's model for emotions, as it uses a 4-inch LCD display accompanied with a light at the

chest of the robot with three different levels of intensities for each of the eight categories of emotions; resulting in 24 in total of emotional states [39], as shown in figure 24 below.

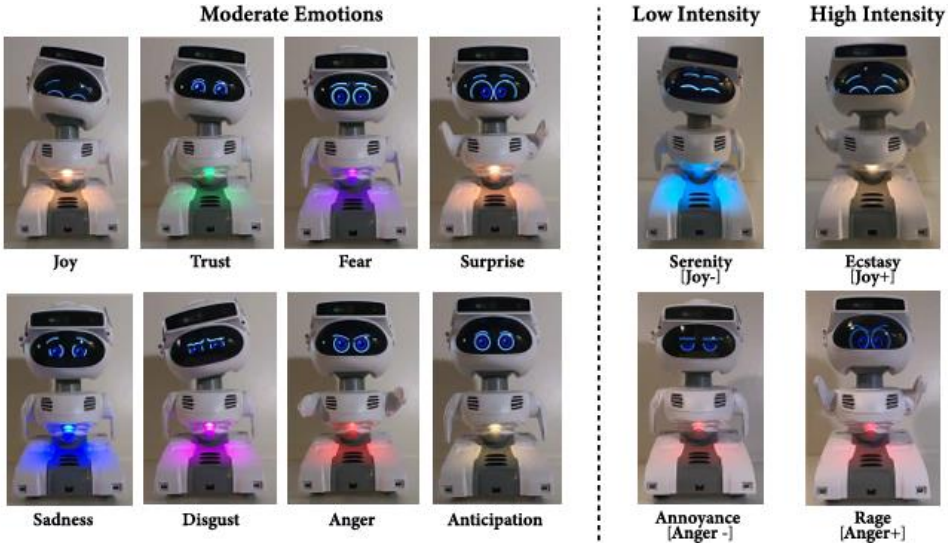


Figure 23 Misty II facial expressions and lighting [39].

- **Simple Cartoon Symbols:** in the following study [37], such method is applied in the FEER-HRI, Facial Expression Emotion Recognition based on Human Robot Interaction, which is a four layered system framework. Thus, it allows the companion robot to recognize the emotional state of the human it is interacting with at that time. Although these symbols could be useful; however, it does not meet the requirements of a companion robot which deals with neither autistic children nor elders. Moreover, as it is presented in figure 25 below, the users' category of such robot could have some difficulties in interacting with such type of HRI.

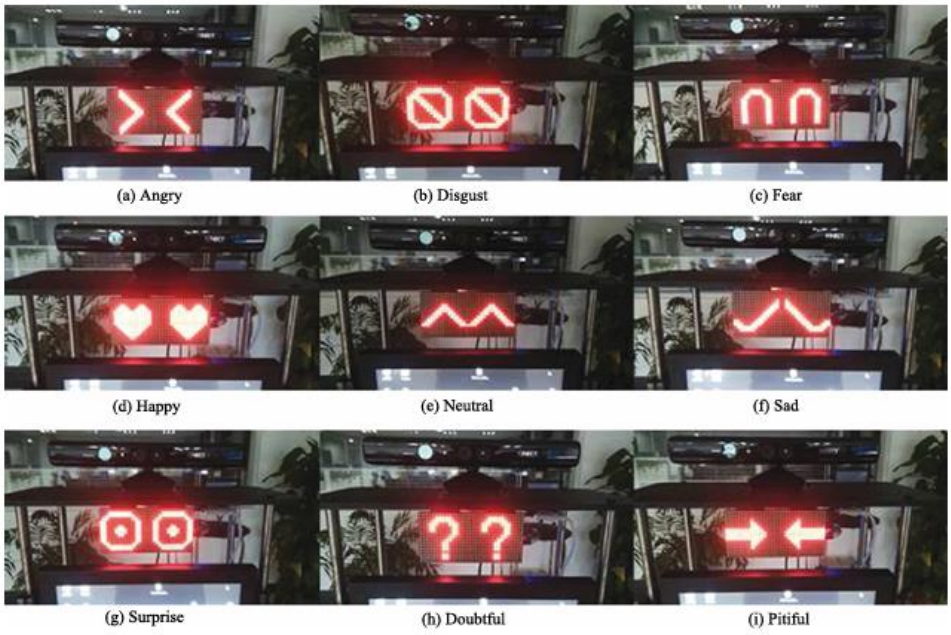


Figure 24 Emotional expressions display using symbols [37].

- **Dynamic Facial Movements:** Humanoids and companion robots could be designed on a biological basis to accomplish the goal of achieving a natural interaction between the user and the robot. In such method, the physical implementation and movement of the facial parts like mouth, eyelids, and eyebrows is required to mimic the natural human response as possible. For example, the following robot discussed in study [2] was designed to employ the dynamic physical features; resulting in the seven basic emotions, as shown in figure 26 below.

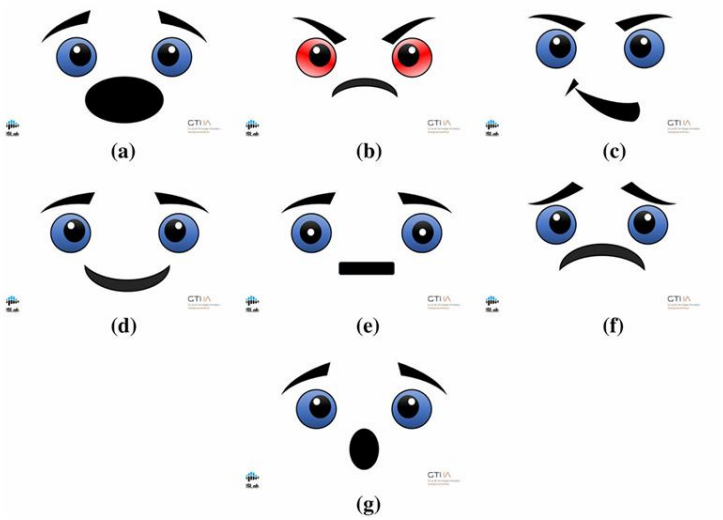
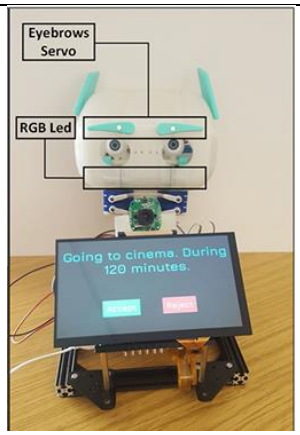


Figure 25 Dynamic robot face. (a) Fear, (b) Anger, (c) Disgust, (d) Happiness, (e) Neutral, (f) Sadness, (g) Surprise [2].

As it could be clarified from the previous information, various types of companion or social robots were developed with a type of emotion display for a successful HRI, as illustrated in table14 below:

Table 14 Emotion displayed for various companion robots.

Reference	Robot name	Robot Image	Number of emotions	Types of emotions	Method of emotion display
[34], [36]	Misty II		24	Joy Trust Fear Sadness Surprise Happiness Disgust Admiration Anger Anticipation Rage (Anger +) Terror (Fear +)	LCD display and a LED colour changing chest light to indicate the depth of the feeling

[2]	EmiR		7 (Emotion features) & 3 (Lights)	Emotions: Fear Anger Disgust Happiness Neutral Sadness Surprise Lights: Red: Happiness Green: Attentiveness Blue: Dominance	Servo motor at the eyebrow and 2 RGD LEDs at the cheeks of the robot for the dynamic physical emotion display.
[38]	NICO		7	Neutral Happiness Surprise Anger Sadness Disgust Fear	8x8 LED matrices at the eyebrows and 8x16 matrix at the mouth

[37]	FEER-HRI		9	Doubtful Surprise Happiness Sadness Disgust Neutral Anger Fear Pitiful	LED matrix to display the emotions in a cartoon like symbols.
[2], [40]	Pepper		4	Joy Sadness Anger Surprise	Human like traits such as gestures and non-verbal communication

For achieving a more realistic HRI, emotion display could be accompanied with another feature such as:

- Actuators for physical movements; for example, actuators could be located in the head, arms, or legs to deliver a natural feeling to the user of the companion robot[2].
- Camera for facial or emotion recognition; for instance, a Kinect camera was employed to capture the real-time images of the user's facial expressions which aids in the recognition of the real emotional status of the user at that specific moment and change the robot's expression accordingly [2], [3], [37].

Based on these four types of emotional display, implementing LCD or LED display was the most beneficial option for developers to implement in the emotion display sub-system, aiding in the enhancement of the interaction between the user and its companion robot. Taking into consideration the objectives of the following report which is developing a companion robot to assist children with ASD and older individuals, relying on the LCD screen could be beneficial due to multiple purposes, as follows:

- **Customable Facial Expressions:** LED matrices could be usable and useful sometimes; however, LCD screens inherit two crucial advantages such as displaying a wider range of features and a high flexibility of designing emotional features pleasant to the user [39].
- **Versatility in utilizing GUI features:** as illustrated in study [39], LCD screens could not only display the emotional state of the robot to ease the HRI, but also it could add features to the robot such as GUI and other modes by the touch screen.
- **Enhancement of HRI:** as mentioned before, a natural and effective emotion display on a robot's face ought to significantly enhance the HRI, resulting in a successful project and guarantee a fulfilment of tasks. Moreover, by clearly displaying emotions, a more enjoyable and engaging experience was achieved for both children and elderly [39].
- **Adaptability and Suitability for Research and Development:** by utilizing the advantage of the flexibility of the emotional expressions, developers could easily adjust the appearance of the robot's features to adapt with the user's requirements[35], [39].

2.2.6. Locomotion in Companion Robots

Mobility defines how social robots move and interact with their environment, and it affects their performance, flexibility, and perceptions associated with them. The two major degrees of movement used in social robots are wheeled, legged, and hybrid, each of these possessing inherent advantages and limitations. Choosing a mode of locomotion means choice with regard to ease of navigation, energy efficiency, and human interaction.

As reflected in Figure 27, social robots primarily use legs or wheels, while a small proportion utilizes hybrid designs in a bid to enhance versatility. These types of mobility allow greater compatibility with human environments.

- **Wheeled Mobility:**

This is by far the most common mobility type for social robots, particularly in indoor environments such as shopping malls, offices, or hospitals. Of a sample of 106 social robots examined [41], 20 of them were wheeled and 63 legged, clearly showing an overwhelming preference for legged mobility in human-centric environments, as seen from Figure 27. Wheeled robots had an average of 3.21 wheels, with most having two for movement and one for balance as shown in Figure 28.

They provide smooth, stable, and power-saving movement and are perfect for tasks with long hours of operation and high-speed movement on flat ground. The majority of differential drive wheeled robots have mechanisms for providing sharp turns and accurate movement. More sophisticated robots have Mecanum or omnidirectional wheels, which provide greater maneuverability in confined areas. The SoftBank Robotics Pepper robot shown in figure 29 employs a differential-drive wheeled base and SLAM-based localization and human-tracking sensors to move effectively through social spaces [42]

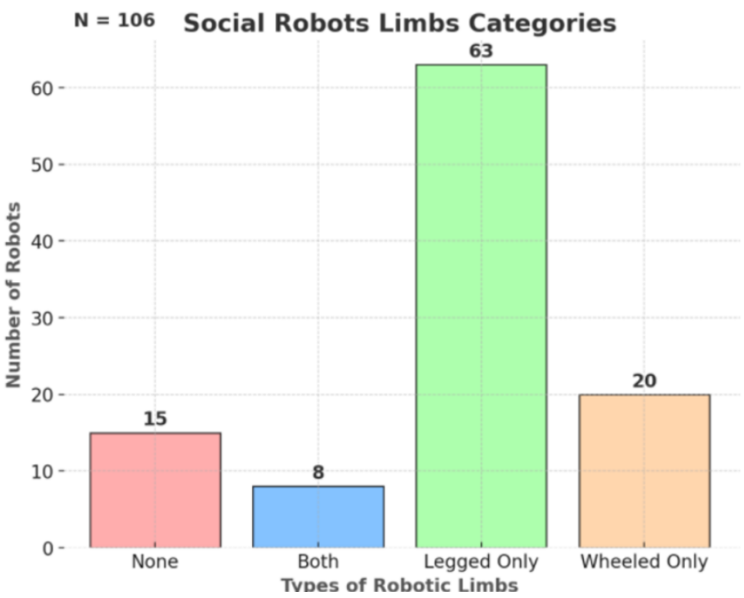


Figure 26 - Categories of social robots locomotion

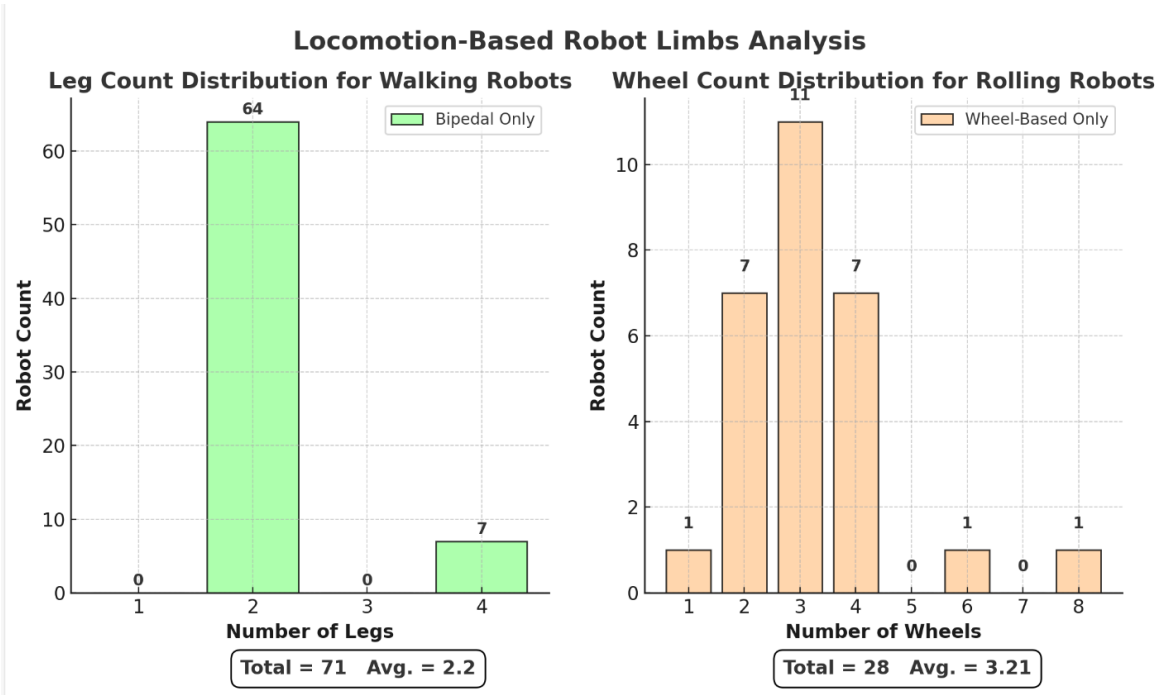


Figure 27 - Number of Legs (left), and number of wheels (right)

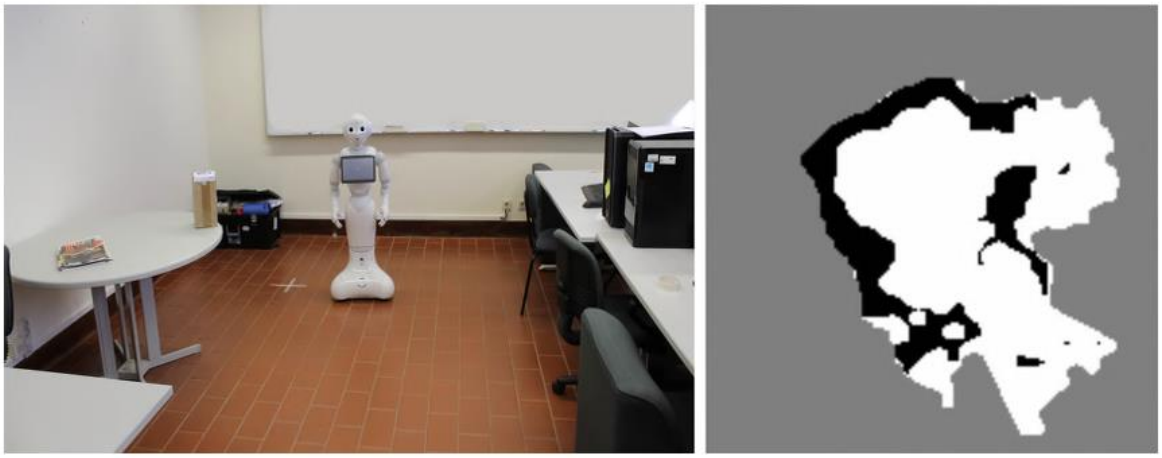


Figure 28 - Pepper mapping environment, real environment (left) and represented map (right).

Advantages of Wheeled Mobility: A very important benefit of wheeled robots is that they are energy-effective. Wheeled mobility calls for less energy than mobility with legged robots, and the batteries are also used for longer periods, thus enhancing their working life. They are thus economical for uses like customer service, security, and delivery assistance. Wheeled robots are also stable because they constantly remain in touch with the ground. This reduces mechanical wear and stress, hence making them outlast complex legged systems. The low mechanical complexity also suggests low maintenance cost and high reliability.

Disadvantages of Wheeled Mobility: Despite their efficacy, wheeled robots are stymied by stairs, uneven ground, and other barriers. They can only travel on even surfaces and therefore are not appropriate for application where mobility in homes with stairs, uneven ground, or outdoor environments is necessary. Moreover, wheeled robots lack human-like mobility and, in applications that expect more natural movements in use, this can cause a reduction in their social acceptance. In comparison to legged robots that are able to change their posture and stride, wheeled robots appear to be less intuitive or more robot-like and, this might indeed affect the quality of HRI.

- **Legged mobility:**

Legged companion robots have been designed for environments more inclined toward human use with its free and easy movement on stairs, rough ground, and crowded areas. Social legged robots are mainly bipedal to mimic human walking. In some designs, quadrupedal structures are introduced for stability. An example of this is Honda's large humanoid robot ASIMO

depicted in figure 30, that performs naturally human-like walking, ascend stairs, and engage socially. Sony Aibo, a four-legged robot, has found applications in companionship, using locomotion to express emotion.

Figure 29 - ASIMO robot as an example of legged social robot

Out of 106 social robots, 63 were legged, which affirms that legged mobility is the trend for interactive applications [1]. 64% of the legged robots were bipedal as shown in figure 28, affirming a trend towards human-like gait patterns.

Advantages of Legged Mobility: Legged robots attain a more natural and socially acceptable type of mobility and thus are most suitable for applications where human interaction is most significant. The fact that they can climb stairs, traverse rough terrain, and walk on indoor floors with obstacles renders them more capable in a greater number of situations than wheeled robots. Also, legged robots promote human-robot interaction through the imitation of the human gait, causing trust and ease in social interaction. They offer a higher capability of adaptive mobilization and step across obstacles to give path to gait modification in realtime.

Drawbacks of Legged Mobility: They are more complex and power-consuming compared to wheeled robots. Their mechanical design entails a more sophisticated way of balancing, thus consuming more power and shortening battery life.

- **Hybrid Mobility:**

It is a type of mobility that combines the wheeled and legged systems to allow them to change from rolling to stepping based on environmental needs. ETH Zurich's ANYmal robot, which has wheels integrated into its legs, can roll along on flat ground and step over obstacles where needed [43] as shown in figure 31.

Advantages of Hybrid Mobility:

The hybrid method provides optimal efficiency on flat ground without giving up the capability for uneven ground and obstacles. In addition, Hybrid robots can execute varied tasks at their best in different environments. True to the meaning of their position between wheeled robots and legged robots, they combine energy efficiency embodied in wheels and the terrain ability of legs and become very highly adapted to certain structured indoor environments and transferring their functionality to dynamic outdoor environments.

Figure 30 - ANYmal robot as an example for hybrid mobile social robot.

Disadvantages of Hybrid Mobility: Still, the mechanical setup of a hybrid robot is quite complex, and to better achieve the transition from wheeled to legged movement will require very advanced control systems. This increases development costs and wears out mechanical components more quickly while taking energy.

Summary of the different locomotion advantages and disadvantages are given in table 12.

Table 15 - Comparison of social robot locomotion types.

Locomotion Type	*Advantages*	*Drawbacks*
Wheeled Robots	*Stable, energy-efficient, cost-effective*	*Only for flat surfaces, lacks human-like motion*
Legged Robots	*Natural movement, improved terrain adaptability*	*Highly energy consuming and complicated mechanics*
Hybrid Robots	*Merges efficiency and adaptability*	*Higher mechanical complexity, high cost*

2.2.7. Localization and Mapping Approaches for Companion Robots

Robust localization and mapping are necessary to enable companion robots to navigate autonomously in dynamic spaces. Such systems allow robots to move around, avoid collisions, and interact safely in human-occupied spaces. More robust localization methods are pertinent to be designed for companion robots since they must endure indoor space dynamics while being adherent to social culture and human activity. During the review, we cover a number of developments in localization and mapping methods that are helping to enhance robot performance, from fusion-based systems to affordance-based models.

One of the notable improvements in robot indoor localization is the fusion of Laser SLAM and Visual SLAM, which is elaborated by [44]. Conventional SLAM methods possess scale uncertainty for monocular vision and mis-localization for repetitive structures within the scene. The method circumvents these limitations by registering laser-based and monocular camera-based SLAM trajectories in order to enhance pose estimation and localize within an error of less than 5%. This approach performs better than individual SLAM models under different

testing conditions. As observed from Figure 32, the system architecture fuses laser and visual information to realize improved localization. Figure 33 presents the test environment (6.8m × 11.8m) and maps created with G-mapping SLAM and ORB-SLAM2.

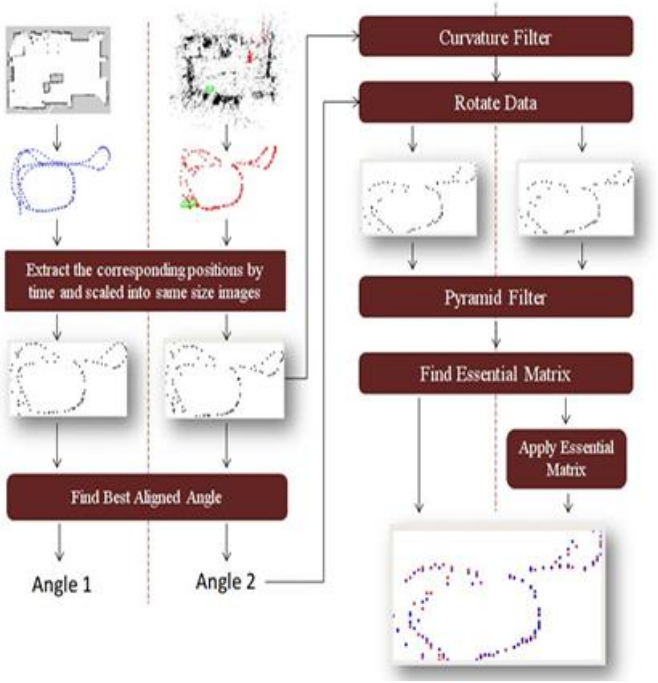


Figure 31 - The architecture of the fusion process is illustrated in the figure. First, data from the laser-based and monocular camera-based SLAM systems are collected. The output (shown in the bottom right side) has blue dots representing data processed under Gmapping (laser-based SLAM), while red dots indicate data processed under ORB-SLAM2 (monocular camera-based SLAM). Alignment, filtering, and essential matrix computation are the steps included in the process [44].

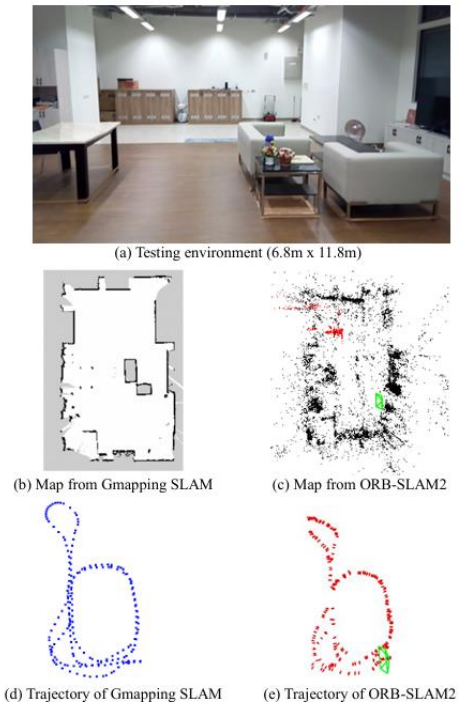


Figure 32 - The testing environment is shown in (a), and the subsequent maps are generated by Gmapping SLAM (b) and ORB-SLAM2 (c). Figures (d) and (e) illustrate the trajectories followed by the two SLAM systems: in blue for Gmapping SLAM and in red for ORB-SLAM2 [44].

In mobile settings, where the standard Adaptive Monte Carlo Localization (AMCL) can be interrupted by moving obstacles, this fusion-based SLAM framework significantly enhances pose estimation and trajectory alignment.

Figure 34 demonstrates the model's correction of mislocalization errors and convergence of diverged AMCL estimates, and Figure 35 demonstrates the ultimate fusion outcomes with overlapped maps and aligned trajectories.

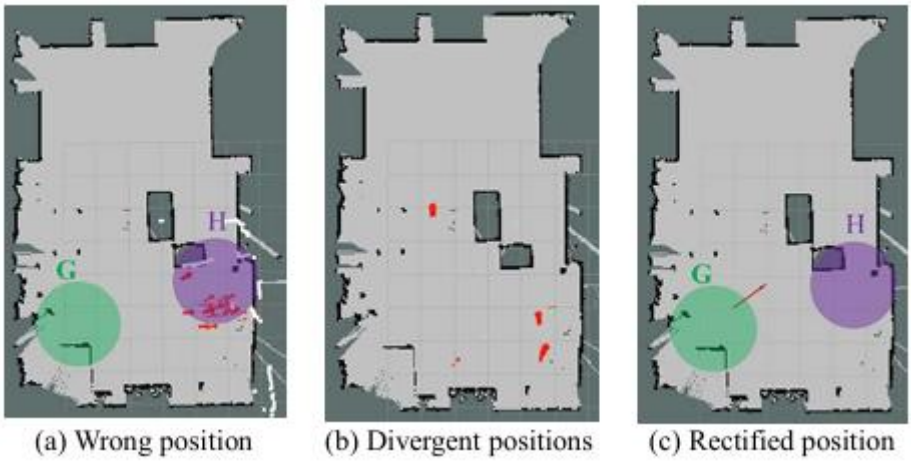


Figure 33 - The results of the experimental verification as conducted. Panel (a) shows an example of how AMCL converges to an incorrect position due to object geometry similarity and panel (b) illustrates a problem of perceiving multiple positions for a location due to moving persons. The problem is solved by our system, as shown in panel (c), which indicates the corrected location [44].

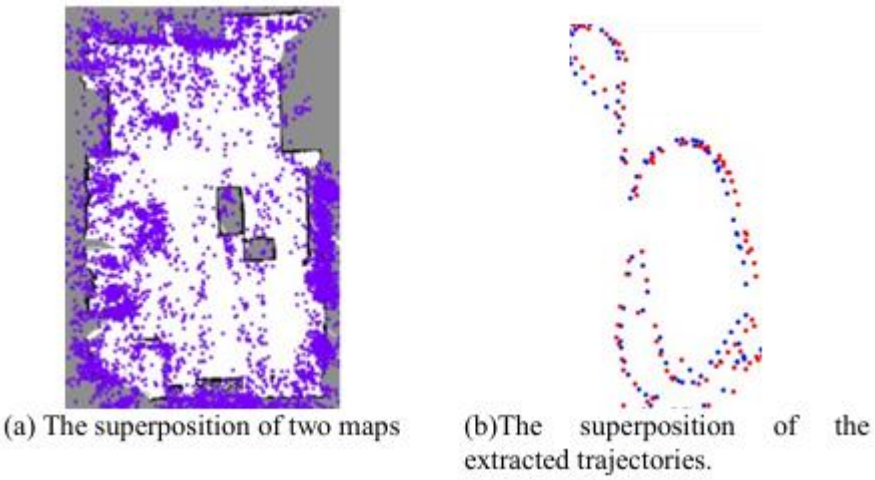


Figure 34 - (a) shows map fusion, which indicates the area of fusion not just over the trajectory area but over the map as a whole. (b) represents the overlay of the extracted trajectories, where blue dots correspond to the Gmapping algorithm's trajectory and red dots are for the ORB-SLAM2 algorithm's trajectory [44].

Tests with a Pioneer 3-DX robot (equipped with Hokuyo laser and monocular camera) achieved the most accurate localization, whereas a Pepper robot, having lower-resolution sensors, saw significant improvement with the fusion model (Figure 36).

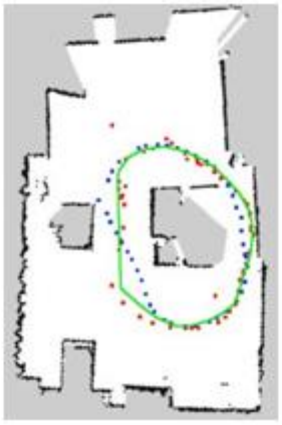


Figure 35 - The trajectory denoted by blue dots is from the Gmapping algorithm; red dots indicate the trajectory from the proposed algorithm. Green loops denote the ground truth trajectory, established before the experiment [44].

The system performance was evaluated with the help of Table 13, verifying the correctness of scaling, which assures the minimization of relative localization errors for different test trajectories. While the approach enhances robustness, it comes at the cost of demanding high-quality laser data, and its deployment in resource-constrained environments becomes difficult, suggesting that further optimizations on sensor fusion and adaptive scaling techniques are needed.

Table 16 - Results for verifying the scaling accuracy of localization [44].

Path	Real distance (m)	Distance in ORB-SLAM2 map (unknown unit)	Distance after applying G_{LV} to the position in ORB-SLAM2 map (m)	The distance ratio with and without applying G_{LV} on ORB-SLAM2	Relative error (%)
AB	1.53	0.1985	1.48	7.46	3.14
AD	2.92	0.3739	2.86	7.65	2.04
AE	6.71	0.8824	6.82	7.73	1.64
BF	4.19	0.5244	4.00	7.62	4.61
CD	2.63	0.5314	2.63	4.95	0.00
FD	4.00	0.5146	4.15	8.07	3.84

An improvement over the SLAM fusion approach, [45] introduced RGB-D Visual SLAM, which strengthens localization by depth sensing augmented with feature-based mapping. A more viable option than conventional LiDAR-based SLAM is provided by the system via the ability of real-time 3D mapping and precise pose estimation. In comparison to 2D LiDAR, which relies solely on geometric features, RGB-D cameras provide depth perception, which also aids in interpreting the scene. System architecture, as indicated by Figure 37, illustrates how visual data and depth are integrated for mapping

In indoor structured environments, the RGB-D Visual SLAM system is of better performance than conventional LiDAR-based systems, as demonstrated by experimental results in Figure 38.

Figure 36 - The proposed setup for the experimental platform [45].

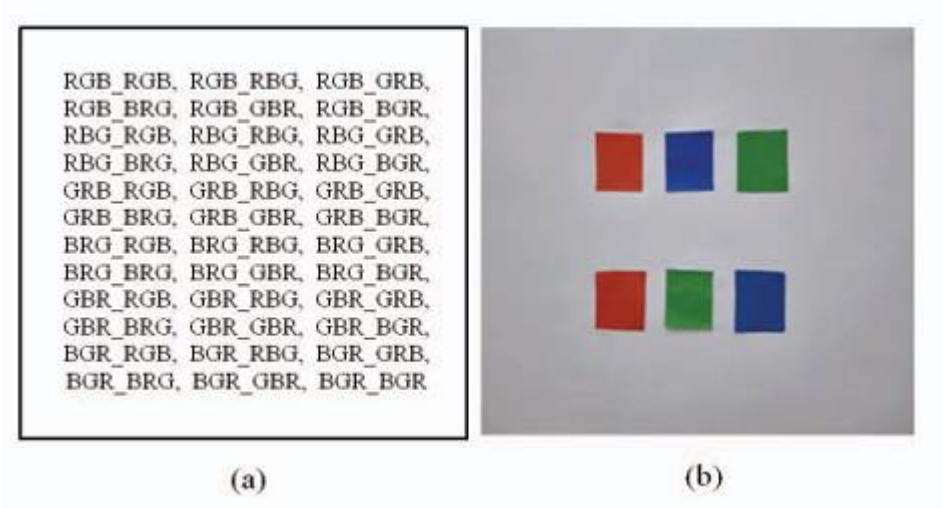


Figure 37 - The experimental setup for the scene [45].

Thus, RGB-D SLAM continues to be sensitive, and with environmental conditions such as lighting and noise interference by sensors, these further degrade the measurements of depth in low-textured scenes. But notwithstanding the shortcomings discussed, this advanced navigation system renders scene understanding quite high and localization ability very promising; all these qualities make it most preferable to be adopted in indoor robot navigation.

The second major advancement is provided by [46], who proposed a multi-vision localization system for enhancing service robot navigation by fusing global visual localization and onboard vision. The system is unlike conventional AGV-based systems that are guide line dependent. Conversely, the multi-vision system provides free, guide-free navigation via Visual SLAM (V-SLAM). Visual odometry is utilized for estimating movement, and 3D point cloud mapping is used for precise localization in the system. Figure 39 illustrates the system architecture, in which it is described how global video surveillance is combined with local depth sensing for real-time positioning enhancement.

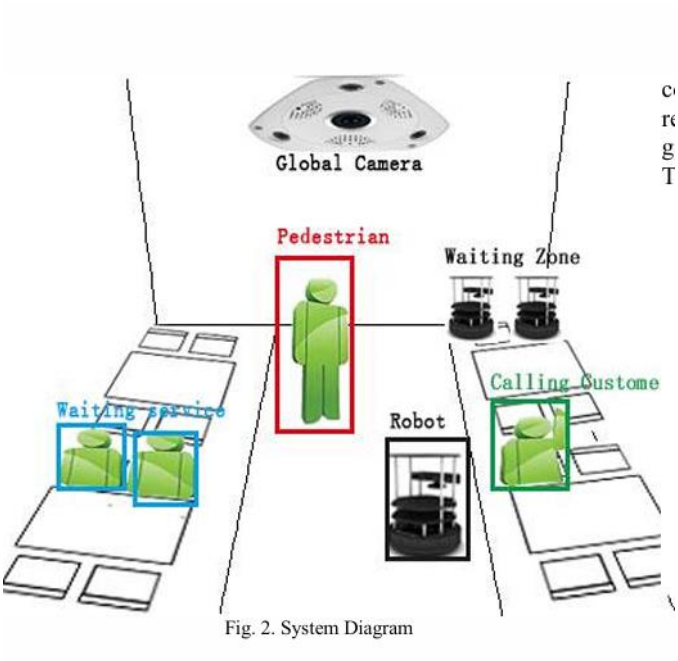


Figure 38 - System Depiction [46].

Additionally, Figure 40 illustrates the V-SLAM process where real-time images are matched to an a priori environment map to improve the accuracy of localization.

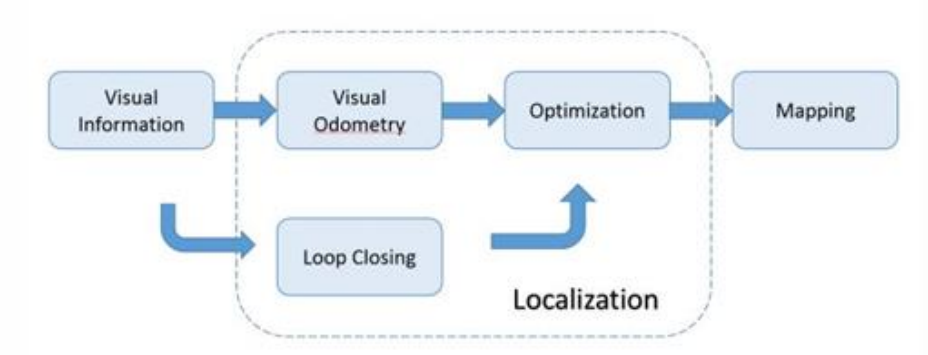


Figure 39 – Depiction of V-SLAM Processes [46].

The system works in a two-stage process of global rough estimation and local precision refinement. Global surveillance cameras first give a rough position estimate through SIFT (Scale-Invariant Feature Transform) feature extraction. Onboard depth sensors then refine the estimate for real-time tuning, as illustrated in Figure 41. The hybrid global-local localization system lowers computational complexity while increasing efficiency and flexibility and is thus suitable for dynamic service environments like restaurants and public areas.

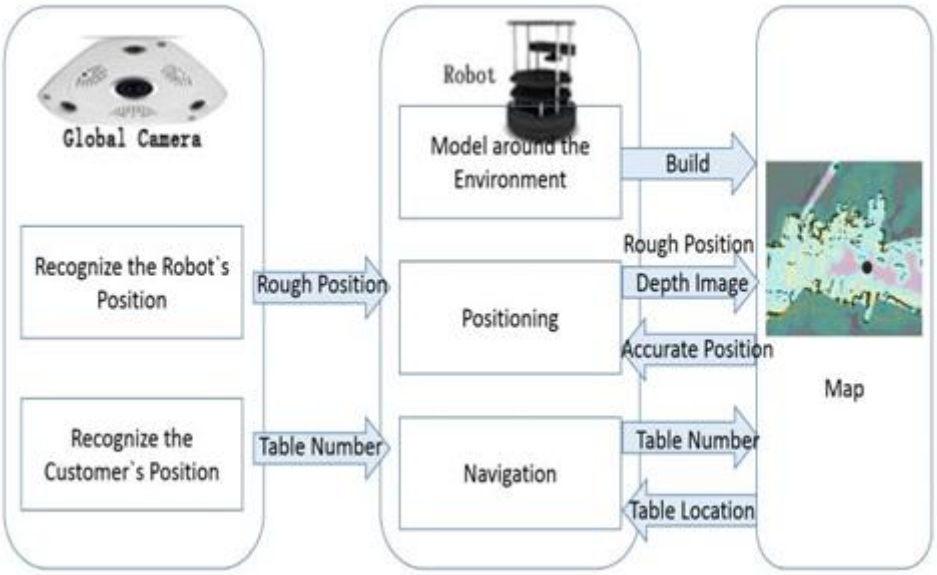


Figure 40 - Localization Diagram [46].

Improving localization techniques, [47] suggested a system that combines Adaptive Monte Carlo Localization (AMCL) with 2D occupancy grid maps and 3D Normal Distributions Transform (NDT) maps. The combination improves navigation in structured environments such as homes, whose layout could be altered often. AMCL uses probabilistic particle filtering to estimate the robot's pose from sensor data, and NDT mapping enhances spatial representation so that robots can navigate in real-world environments. The system was successfully applied to the Robot Companion for home assistance and showed its feasibility for home environments where the robot must get around by itself while interacting with humans.

Lastly, [48] proposed a very novel affordance-based localization model that combines scene semantics, object recognition, and human interaction awareness. In contrast to conventional SLAM-based approaches that mostly rely on geometric features, the model adds four hierarchical layers for enhancing context-aware localization. Figure 42 shows the multi-layered framework, and Figure 43 shows how robots utilize scene inference in detecting functional

spaces from recognized objects. The system consists of a static map layer for spatial context, an object layer to identify objects in real-time through YOLOv3, a scene layer to classify space based on the arrangement of objects therein (e.g., TV and sofa indicating a living room), and an event layer which analyzes human interaction to optimize navigation in real-time.

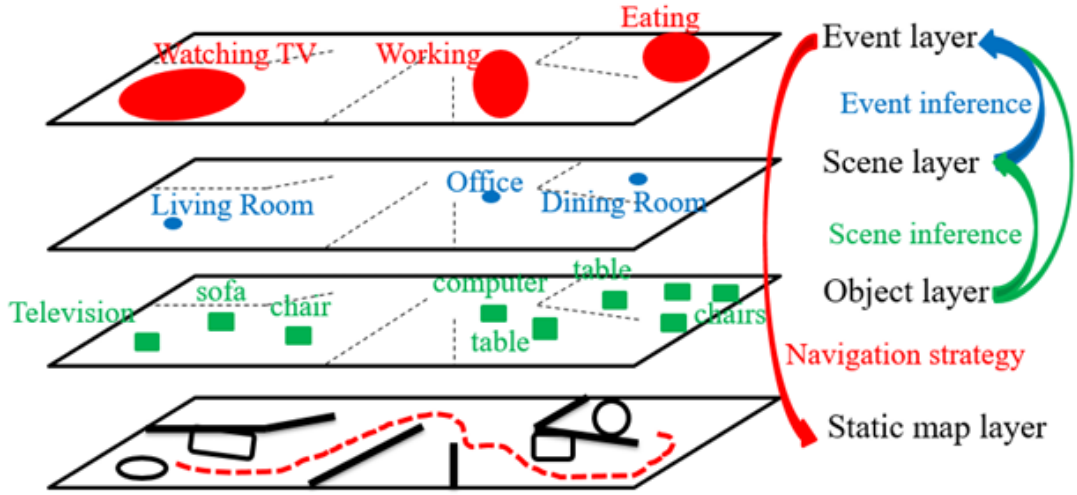


Figure 41 - The multilayer environmental affordance map proposed architecture [48].

This affordance-based model provides several benefits, including real-time flexibility and better human-robot interaction. The system reduces public space disruption with dynamic localization updates based on human activity and environmental dynamics. Figure 43 shows the robot adjusting its navigation path to avoid areas that are occupied, enabling socially compliant movement. Furthermore, Table 14 demonstrates event inference results, validating the model in inferring levels of human engagement. But it is constrained, for example, relying on YOLOv3 object detection precision that may result in errors in scene inference.

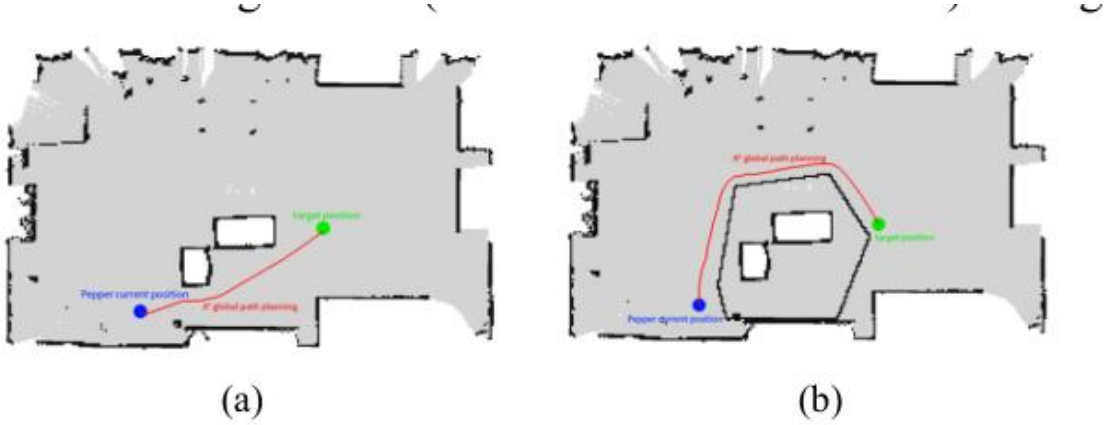


Figure 42 - Illustrates how the mobile robot maneuvers from its current location (blue point) to the target position (green point). (a) shows the absolute shortest global path computed using the static map. (b) Shows the adapted global path, which has also taken [48].

Table 17 - Event inference results [48].

	Skeleton Detection				**Object Layer**		
							
	stand	*eat*	*talk*	*work*	*sleep*	*watch TV*	*Result*
P(event,scene)	**0.33**	**0.33**	**0.33**	<0.01	<0.01	<0.01	-
P(event,object)	< 0.01	**0.5**	<0.01	0.25	<0.01	0.25	eat
P(event,skeleton)	< 0.01	<0.01	<0.01	**0.5**	<0.01	**0.5**	work/ watch TV
P(event)	< 0.01	**0.94**	<0.01	<0.01	<0.01	0.06	**eat**
Ground Truth							**eat**

Second, sensor noise during lighting changes will diminish mapping stability, as can be seen in Figure 44 with the robot updating movement commands according to perceived human

actions. In spite of these issues, this affordance-based localization method significantly improves indoor navigation through additional environmental semantics along with situational awareness in real time regarding humans

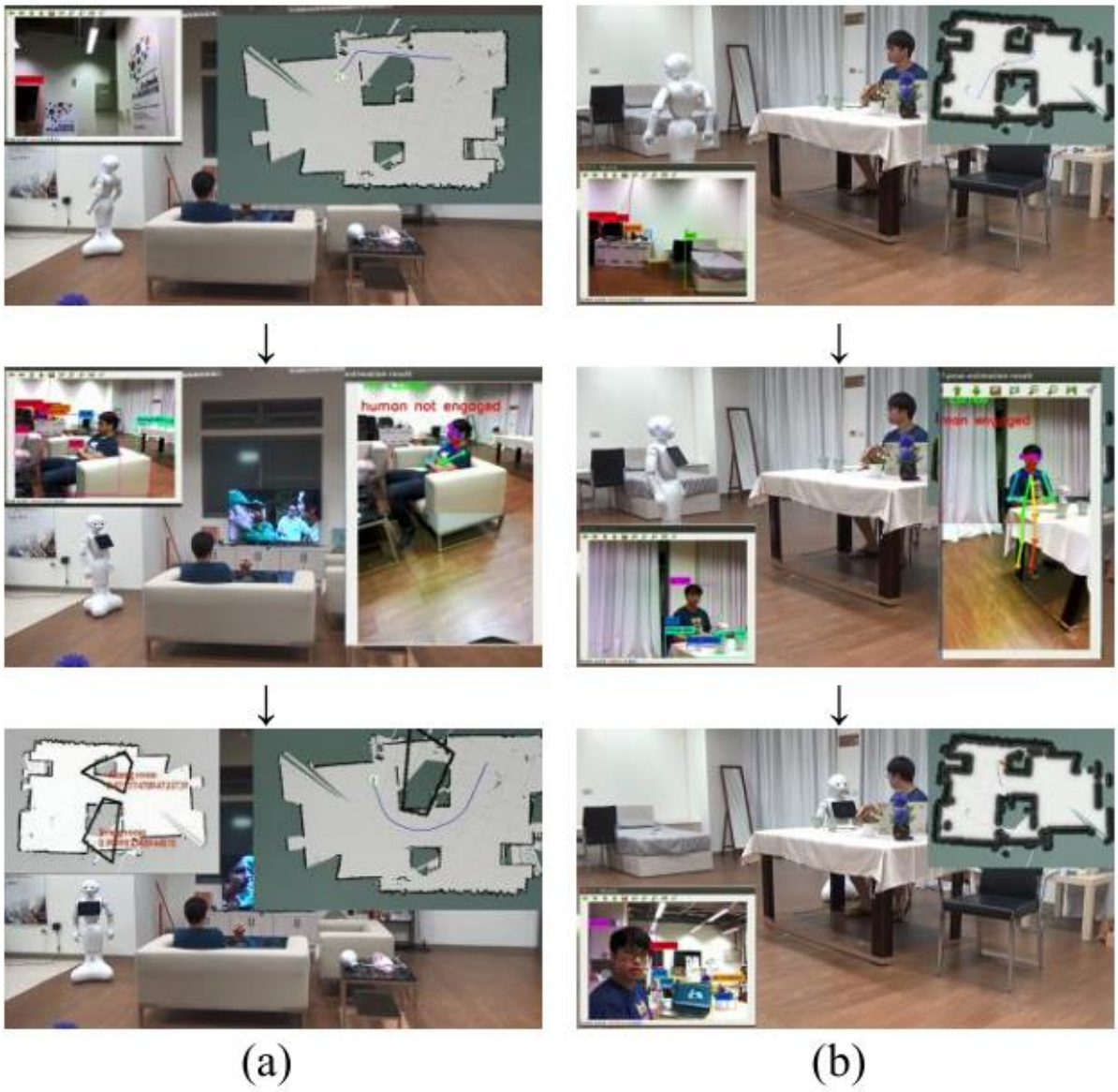


Figure 43 - (a) In this first scenario, the one shown in the left column, Pepper travels toward an office while not disturbing a person who is watching television. (b) This second scenario (right column) demonstrates how Pepper diverts from its intended navigation trajectory to come in front of a person for possible enjoyment of further interaction [48].

In summary, all these localization methods are strides towards making it possible for companion robots to move through complex, human-centered environments. Both the fusion-based SLAM by [44] and the RGB-D Visual SLAM by [45] are first efforts at improving navigation accuracy through the fusion of laser and visual data. The multi-vision system is recommend by [46] to offer flexibility, since it permits navigation without guides. ONDRT-

amcl integration addresses safe navigation in highly organized environments according to [47]. The technique of affordance-based models by [48] is the most matured offering for doing socially aware navigation dynamically adapted to both human activity and changes in surroundings. As the techniques improve, these companion robots will be better at interacting in a more natural and more independent way within human spaces, improving the overall experience of human-robot interaction.

2.2.8. Socially Aware Robot Navigation

Social robot navigation is a vital area of research that aims to enable robots to navigate in concert with humans in a shared environment. Socially-aware navigation aims to enable robots to navigate human-populated environments in a manner that respects social norms, human comfort, and personal space. Early work in the field focused on straightforward methods for avoiding physical obstacles and following pre-specified paths. As robot autonomy and human-robot interaction (HRI) capabilities were developed, however, it became important for robots to take into account also the social dynamics of the environment. The articles surveyed here cover a number of contributions to socially aware navigation, ranging from simple obstacle avoidance to more advanced models that respond to human behavior and social norms.

One of the first socially-aware navigation papers was that of [49], where they presented an unsupervised learning method founded on relative motion prototypes (RMPs) learned from real human motion data. The method produces dynamic cost maps to guide the robot in a socially normative motion in a way that optimizes task efficiency. Hierarchical clustering and an asymmetric Dynamic Time Warping (aDTW) algorithm are used by the system for grouping similar human motion sequences into prototypes. It worked better than the Proxemics-based baseline in that it gave smoother and more natural trajectories as shown in figure 45 with less redundant turning and regard for human comfort zones.

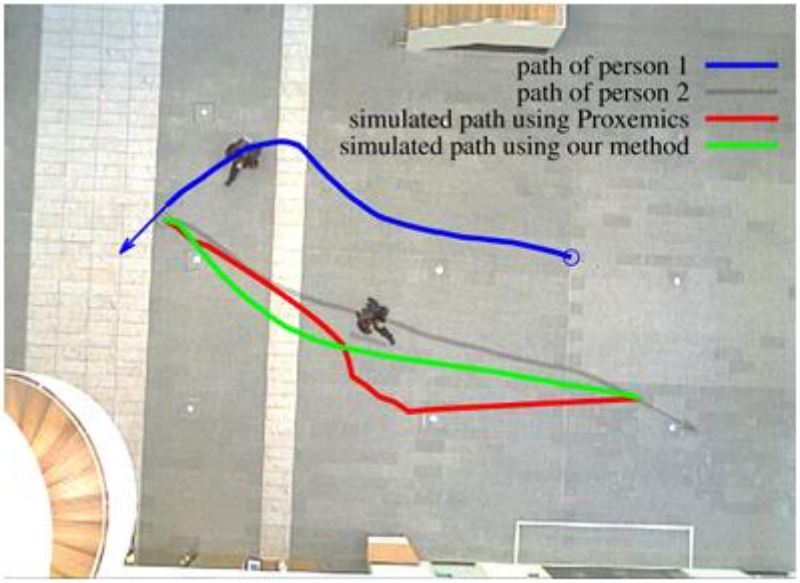


Figure 44 - There are two pedestrians (blue and gray) that have been avoiding each other. In the figure, the learned relative motion prototype (green) mimics the trajectory traced by a real person (gray), while the Proxemics-based path (red) has been needlessly diverted and stretched. The result of the learned method is a navigation path that is more human-like and efficient [49].

In their evaluation, the authors observed significant gains over the baseline both objectively and subjectively, with the system reducing path length by 55% and generating paths that were closer to human paths visually. Such results show learning of socially-aware navigation behavior from real-world data and a more natural, more efficient robot motion planning model for environments shared with humans. This work paved the way for more sophisticated models that involved incorporating human actions within the decision-making of the robot.

Later on, [50] suggested a probabilistic socially compliant mobile robot navigation by modeling human pedestrian motion as a mixture distribution. The approach captures both discrete navigation choices, i.e., left or right turns, and continuous deviations along human paths. For the effective prediction of pedestrian movement in dynamic environments, the authors employed Hamiltonian Markov Chain Monte Carlo sampling. The outcomes indicated that this model illustrated in figure 46 could capture the stochastic nature of individuals' movement in office buildings, confirming its usefulness in practice.

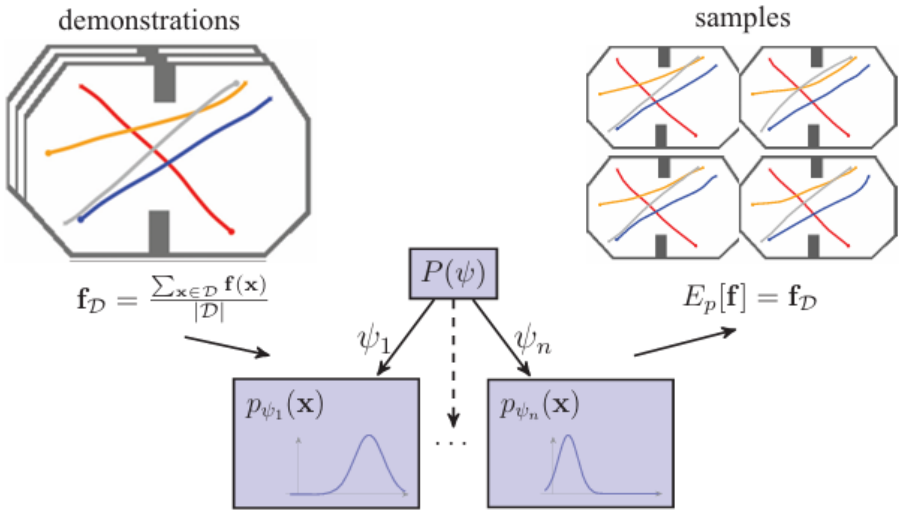


Figure 45 - A model of human cooperative navigation behavior is learned from a mixture distribution over composite trajectories with this method. The model encompasses both discrete and continuous aspects of the behavior. Besides, it generalizes this ability in new situations while allowing the drawing of samples of trajectories representing the stochastic nature of natural navigation [50].

The capacity of the system to generalize to novel situations and learn human actions distinguishes it from other conventional models. Besides, [50] employed a broad evaluation with cross-validation and a Turing test in determining the human-likeness and precision of the predictions of the model. The outcomes were that the behavior of the robot is rated as more human-like when controlled using the learned model compared to other models, including the social forces model and reciprocal velocity obstacles as presented in figure 47. The method worked particularly well in dynamic and dense environments where pedestrian movement is

less predictable. The success of the model in such conditions highlighted its potential in enhancing human-robot interaction in complicated scenarios.

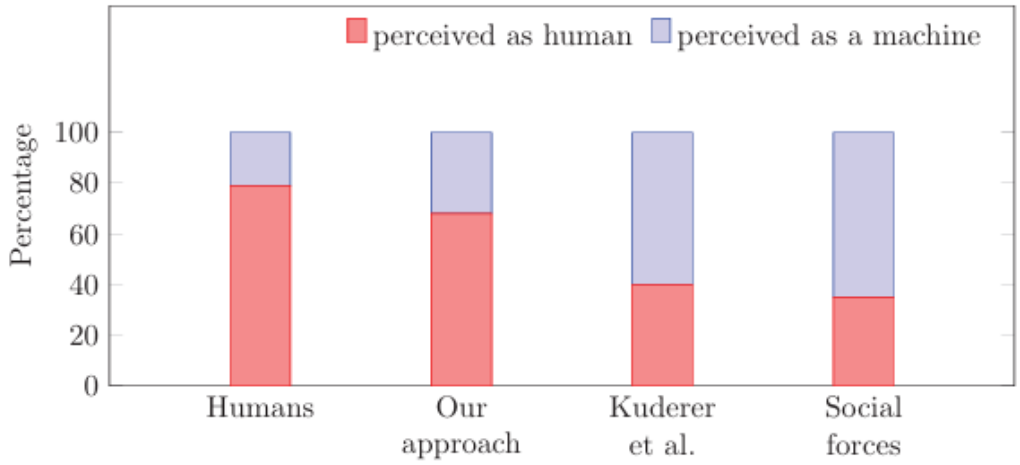


Figure 46 - The study was used to investigate whether behaviors triggered by our method, [51] Kuderer's method, and social forces model [52] are perceived to be human-like. Results suggest that our method was perceived to be most human-like; that is, the percentage of times it was perceived as human was higher than the other model approaches [50].

In 2016, [53] presented a novel Bayesian Inverse Reinforcement Learning (BIRL) based approach to model socially normative robot navigation behaviors. Unlike traditional methods founded upon hard-coded rules or over-restrictive assumptions about human behavior, BIRL allows robots to learn from expert demonstrations on a graph-based map that naturally encapsulates task-specific constraints. The plan was experimented in three settings: lobby, hallway, and intersection, and the results indicated that socially normative behavior—i.e., polite and friendly navigation—resulted in longer path lengths and greater maneuvering but enhanced mission success and reduced disruptions to social relationships. An example in the lobby environment showed that polite behavior prevented intruding on people's space, while the friendly behavior guided the robot toward individuals which can be seen in figure 48.

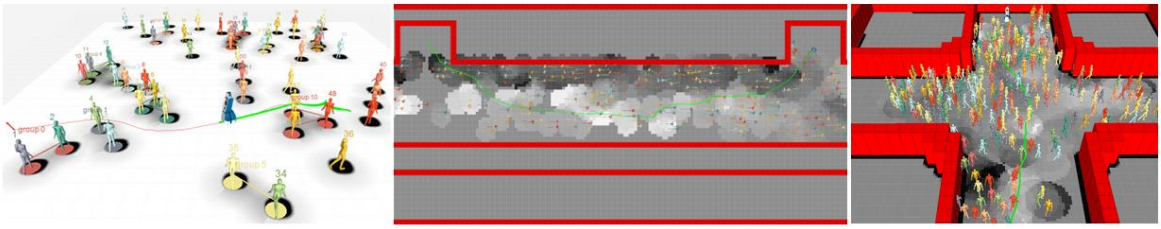


Figure 47 - Shows a number of experimental setups: left, a lobby with 400 people moving in two flows while the robot avoids opposing flow over a long path; center, hall with 50 people and 18 pairwise relationships, depicting the robot's navigation based on social interactions; and right, intersection with 280 people where the robot performs slipstream navigation by following individuals with similar bearing. The planned path by the robot is indicated in red, whereas

the executed trajectory is in green, while costmaps (gray) show learned behaviors with dark regions indicating high costs [53].

The system was also integrated into the robot Daryl, which conditioned its navigation by not interrupting an individual in a conversation without otherwise invading personal space clearly evident in figure 49. This illustrated that the robot was able to generalize from a simple set of features to highly elaborate socially normative behaviors. The BIRL approach showed its usefulness in real-world situations by enabling the robot to navigate in a socially aware manner while optimizing mission completion.

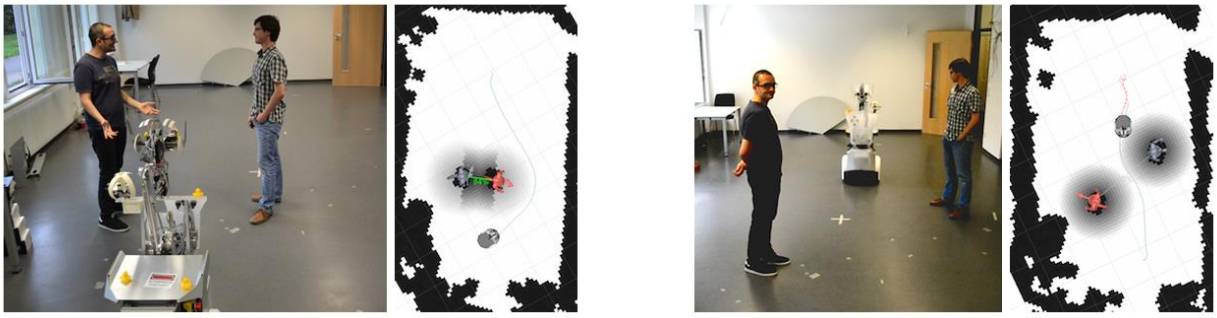


Figure 48 - The figure presents two sets of experiments—top for unconstrained environments and bottom for urban settings at the Barcelona Robot Lab. The first column displays the environment layout, with paths for the robot in green, indicating its movement across different pedestrian densities. The second column shows performance comparisons between the Proxemics approach (black), SFM without prediction (red), and SFM with prediction (green). The third column presents bar diagrams of goal achievement percentages across different pedestrian densities, highlighting the effectiveness of each method in varying crowd conditions [53].

The paper [53] took another path of socially aware robot navigation with the application of the Social Force Model (SFM) to offer human-aware navigation in crowded spaces. The authors calibrated robot-human interaction parameters, which were robot-specific for the Tibi robot, for the robot to be able to socially interact more effectively with humans in dynamic city environments. It was evaluated through simulations and experiments in the real world, where it performed more effectively in groups of pedestrians by avoiding collisions and replicating the motion of people around it as in figure 50.

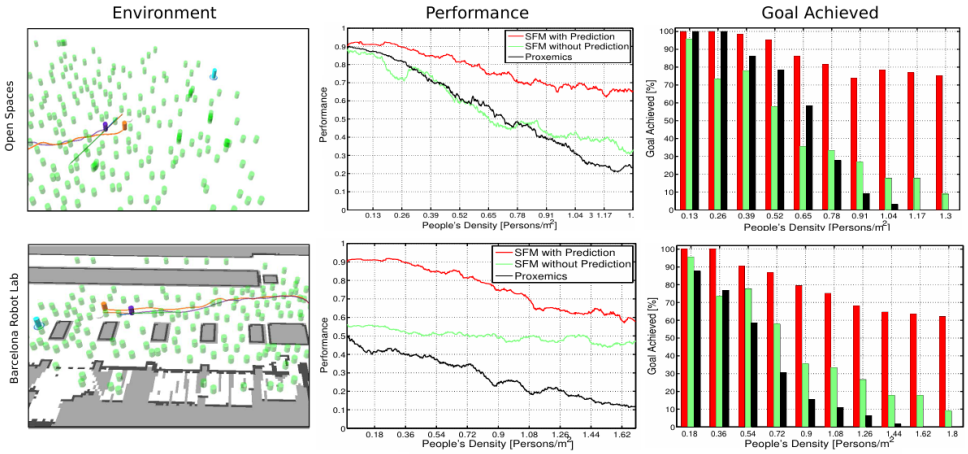


Figure 49 – Depicts two experimental arrangements: the top is for unconstrained environments whereas the bottom is meant for urban settings tested at the Barcelona Robot Lab. The first column shows the environmental layout with the robot paths in green showing its trajectory across different pedestrian densities. Second, a performance comparision among the Proxemics approach (black), SFM without prediction (red), and SFM with prediction (green) is performed. Finally, in the third column, bar charts show the differnt goal achieves across pedestrian densities, revealing an effectiveness schematic of each algorithm against different crowd conditions [53].

A feature that stood out in this work was that it set a measure for proxemics, such that the robot maintained human comfort distances while interacting. This enabled the robot to be able to quantify its capability of remaining within socially acceptable limits, modulating its behavior according to human reaction in real time. The system also illustrated that the robot learned from human reaction, with its navigation strategy changing over time. Success of the system is also shown in figure 51 using real-world experimental results, where the robot succeeds in achieving its goals while adhering to socially acceptable distances as in figure 52.

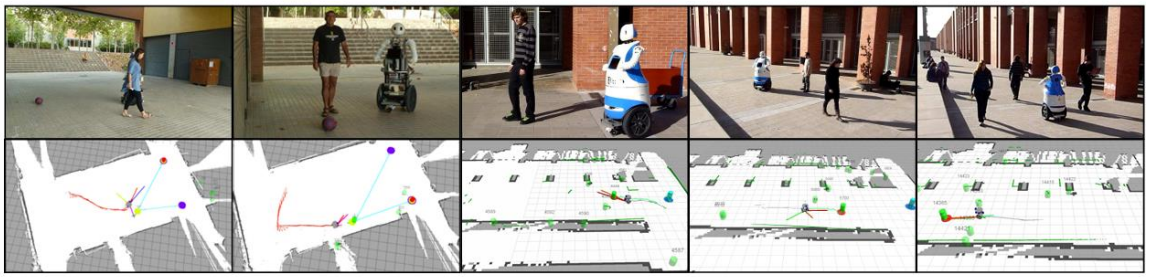


Figure 50 - This illustration shows how the robot Dabo is actually used in real-life experimentation. The upper part depicts Dabo accompanying a person toward a particular goal while following the movement and path of that person. The lower part shows a screenshot of this very same scene represented on the system interface with highlighted robot's trajectory along with a map of the environment. All this is augmented by the implementation of 3D mapping and visual feedback, which improves navigation performance of the system [53].

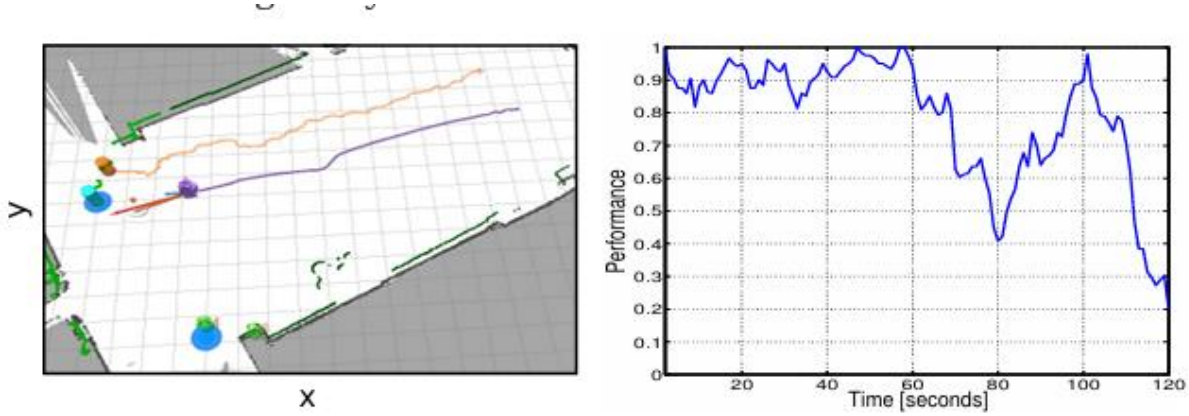


Figure 51 - The accomplishment that trail and performance in experimental data collected has shown figure. On the left, show trajectories of robot and volunteer with different colors in the environment in which different paths taken. Performance obtained during the experiment is visualized in the temporal surface course on the right-hand side, where the performance metric is given on the Y-axis and time (in seconds) on the X-axis. This chart shows the robot's navigation achievement in doing the task. [53].

Socially aware navigation was also promoted by [54], proposing an adaptive spatial density function-based system to separate individuals based on their social activity and mark no-go areas for robot navigation.

Space affordances, or areas around objects that humans naturally interact with, are also part of this framework. The combination of both these modules helped the robot walk through crowded human spaces and keep out of personal spaces. The robot was also able to prevent human-object interactions that are discomforting. The performance of this system was verified in both virtual and real world settings, in which the robot was able to avoid discomforting humans successfully and keep a proper distance as the scenario in Figure 53.

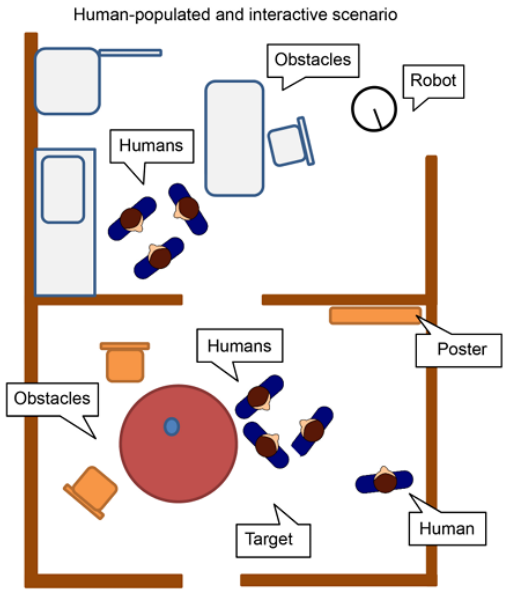


Figure 52 – Shows the experimental setup for the method detailed in this paper. The robot is operating in an environment populated with humans and obstacles, and interactive features such as a target and a poster. The robot must find the optimal path through all this while avoiding causing distress to any of the nearby humans [54].

Moreover, the system's path planning kept the robot away from personal spaces and areas of activity around objects, which rendered it socially more compliant. The contrast with conventional navigation techniques demonstrated the system's superiority in "average minimum distances" and reduced "personal space intrusions" shown figure 54. The findings validate the system for application in navigation through social dynamic spaces and its possible application to real human environments for enhancing robot performance.

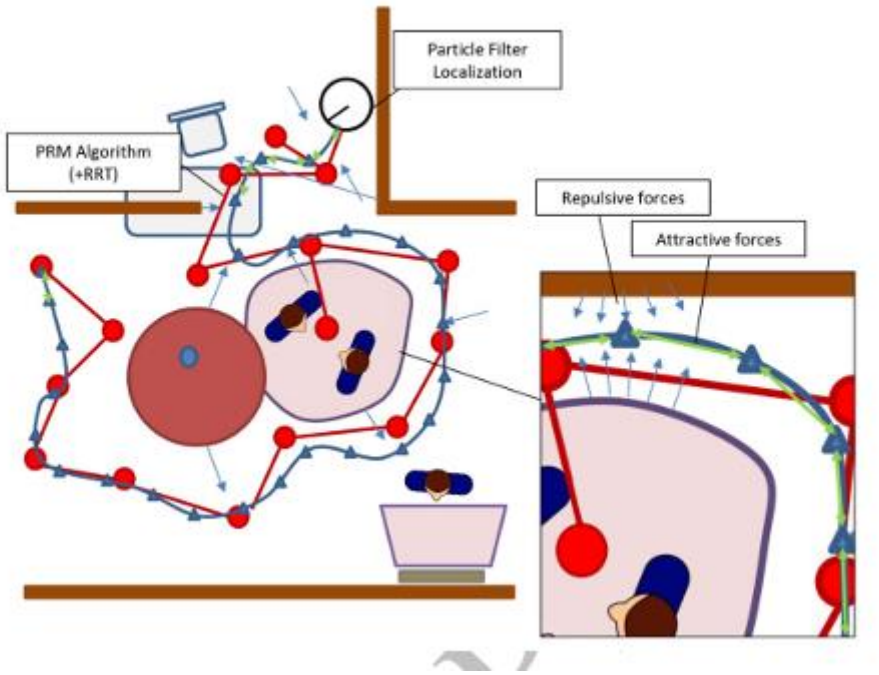


Figure 53 – The path stands represented by the blue continuous line (Δ) as it appears from the final social path. Further, path planners display Gt in red along with the set of forces acting upon the robot. The diagram here demonstrates the repulsive-attractive forces that help the robot maneuver through the environment. For assisting in the completion of path planning and navigation, a particle filter localization as well as a PRM-RRT combination is being used [54].

Most recently, [43] presented a socially aware path-planning framework that computes path costs dynamically to prevent invasions of personal space and object affordances. Social constraints are incorporated into classical navigation maps show in figure 55 so that robots choose paths that make individuals less uncomfortable. Experiments validated that socially aware path-following robots alleviated discomfort and traveled more predictably in shared spaces, traveling more smoothly and steering clear of sensitive regions illustrated figure 56.

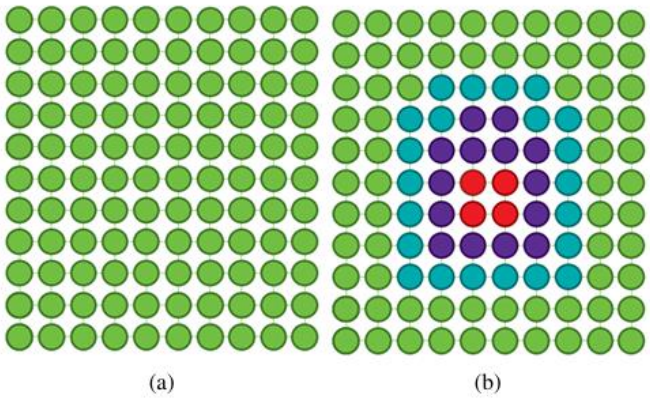


Figure 54 - The picture above reveals graph based grid mapping: it (a) shows the original version of the free space graph while it (b) denotes the final version of the graph on top of which the social interaction space is added, indicated by a gradient of colors.[43].

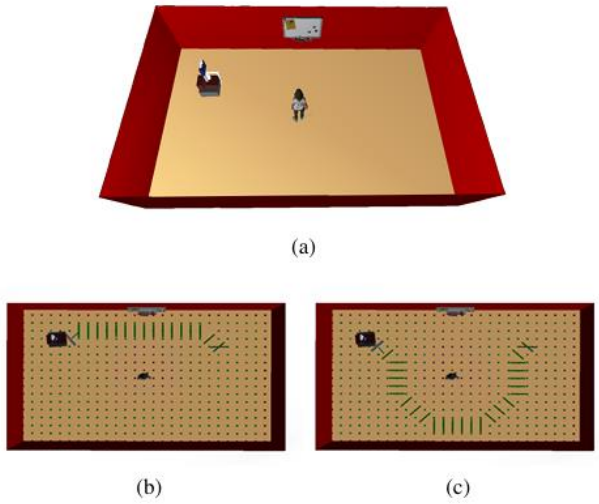


Figure 55 - The figure shows an interactive condition: (a) the original environment is shown, (b) navigation showing how to do it without space affordance, and (c) navigation showing how to navigate with space affordances, where the robot modifies its trajectory around the human [55].

However, the system's accuracy is still dependent on precise human detection and environmental modeling, as errors in estimating personal spaces can affect navigation efficiency. In spite of this limitation, the method substantially improves human-robot interaction and trust in realistic situations. The system is a robust solution to socially-aware navigation and is therefore deployable in numerous applications wherein human comfort and social compliance are imperative.

Collectively, these works represent a progression to the creation of reusable socially-aware robot navigation systems that provide increasingly sophisticated solutions to the problem of human-robot interaction in dynamic environments. Early approaches like those in [6] and [7] focused on learning motion models from human motion data and modeling pedestrian motion

trajectories. As the research later evolved, though, works integrated more sophisticated models like Bayesian Inverse Reinforcement Learning [8] and spatial adaptive density functions [9] for facilitating more real-time and situation-aware navigation. These methods enhanced navigation in the robot in terms of human comfort and social compliance and represent key advances in socially-aware navigation research. As sensor fusion and real-time processing continue to get more advanced, future systems will become more adaptive, which will further improve the performance of robots in human-centric environments.

2.2.9. Ethical Issues of Companion Robots

Though there are many benefits provided by companion robots, their popularity creates ethical along with practical issues that must be considered seriously.

Emotional Dependence and Attachment: Irrespective of the emotional support provided by companion robots, their users might become over-dependent or highly attached to them. Long-term interaction and its psychological influence need to be understood for their ethical deployment at short notice.

- User over-dependency issues can result from the user becoming attached to companion robots because they believe the robots are humans [55].
- Elderly users often view Paro as a real pet, which poses ethical issues such as deception [24].

Privacy and Data Security: The more information about the user companions gather and engage with to facilitate more advanced interactivity the more privacy and security concerns they raise. Safety will only be guaranteed once ethical decisions have been made concerning informed consent, protection of user information, and adherence to laws and policies.Stronger regulatory policies will help ensure the protection of data in an adaptive robotic system [56].

These considerations call for interdisciplinary discourse on the ethical deployment of companion robots, placing emphasis on transparency, informed consent, and user autonomy.

Companion robots constitute a transformative advancement in HRI, demonstrating empirical efficacy in healthcare, education, and social integration. While anthropomorphic and zoomorphic robots are the focus of the current research, functional and caricatured designs also keep on developing. However, challenges such as emotional overdependence and privacy concerns require ongoing scrutiny. Future research should emphasize long-term impact assessments and ethical frameworks that will make the deployment of companion robotics sustainable and responsible.

2.3. Conclusions

Whisper AI definitely ranks higher in terms of accuracy and speed but also brings more to the table as it is meant to be used in cases of human-robot interaction. Second in line is Google ASR but with a higher word-error rate (WER). VOSK works offline, while Sphinx and Deep Speech perform poorly.

ChatGPT-4 Turbo offers a lot more in reasoning, coding and medical inquiries. Compared to Gemini Pro ChatGPT 4 Turbo offers some better features than While Gemini Pro is the best when it comes to translation, in terms of accuracy the model is poorly scored. In QA tasks Microsoft Copilot becomes competitive but just takes the backseat compared to ChatGPT-4

Effective HRI with respect to the assistance of children with ASD and elder individuals must have a simple, appealing, and engaging emotion display system. Several methods were studied, such as using LED arrays, LCD screens, cartoon symbols, and dynamic facial movements, for visualizing emotions in companion robots.

Among the choices, it is the LCD screens that has the capability to display different customizable facial expressions, versatile GUI integration, improved interaction and even usability towards research and development. LCD screen emotive, with a visible external representation of emotions by the user, would help greatly improve engagement, trust, and experience, thus creating the best conditions for a realistic and effective HRI system.

As mentioned above, from the first robotic companion robot to the sophisticated ones nowadays, emotionally expressive systems ought to be built for human interactions; therefore, the information provided above about companion robots demonstrates their remarkable development as time has passed. More realistic and responsive interactions between humans and robots have become possible through the advancements in chatbots and Automatic Speech Recognition (ASR) technologies. Additionally, robots could express themselves by utilization of LCD screens for emotional display, which improves the user engagement and builds deeper attachment, enhancing the HRI. These developments emphasize the increasing importance of companion robots in a variety of applications, from healthcare aids to personal assistance, as companion robots continue to incorporate complex AI driven emotional expression and communication.

Therefore, as was previously mentioned, the study of companion robot's mobility, flexibility, and locomotion highlights the crucial tier of an efficient moving system for flawless indoor navigation and human robot interaction HRI. Furthermore, a variety of locomotion strategies

and types, including wheel and legged design, have been studied to increase the robot's stability and adaptability in southern actions. Moreover, by combining these technological developments, the future of companion robots would be elevated, highlighting the importance of the robot's mobility, emotional intelligence, and effortless communication. As research in fields are in continuous development, future companion robots ought to become more sensitive, flexible, and capable of providing meaningful companionship and assistance in daily life.

Chapter 3 : Mechanical Design

This chapter investigates the mechanical design process of the companion robot with a focus on developing a structure that satisfies aesthetic, function, and durability. The chapter discusses all phases of the design from concept sketches to simulations that can confirm the final design will satisfactorily fulfill operational & safety considerations.

3.1. Introduction

Mechanical design is mainly one of the factors that describe the capabilities and functionality of the companion robot. An efficient mechanical framework would provide stability, smooth operation, allow assembly and disassembly with ease, and also ensure low maintenance even with a modernized design. The framework also aims at seamless integration of electronic components, mobility and interaction optimization, and reliable long-term use in the real world.

The main goals for achieving best design in our opinion was the following:

- Create a friendly shaped design suitable for users of different ages and feels natural in homes and workspaces.
- Ensuring stability and durability, the robot remain balanced and operate smoothly.
- Achieve easier assembly and disassembly by using heat inserters which help in future maintenance and troubleshooting.
- Integrate screen and sensors without making the robot looks too mechanical.

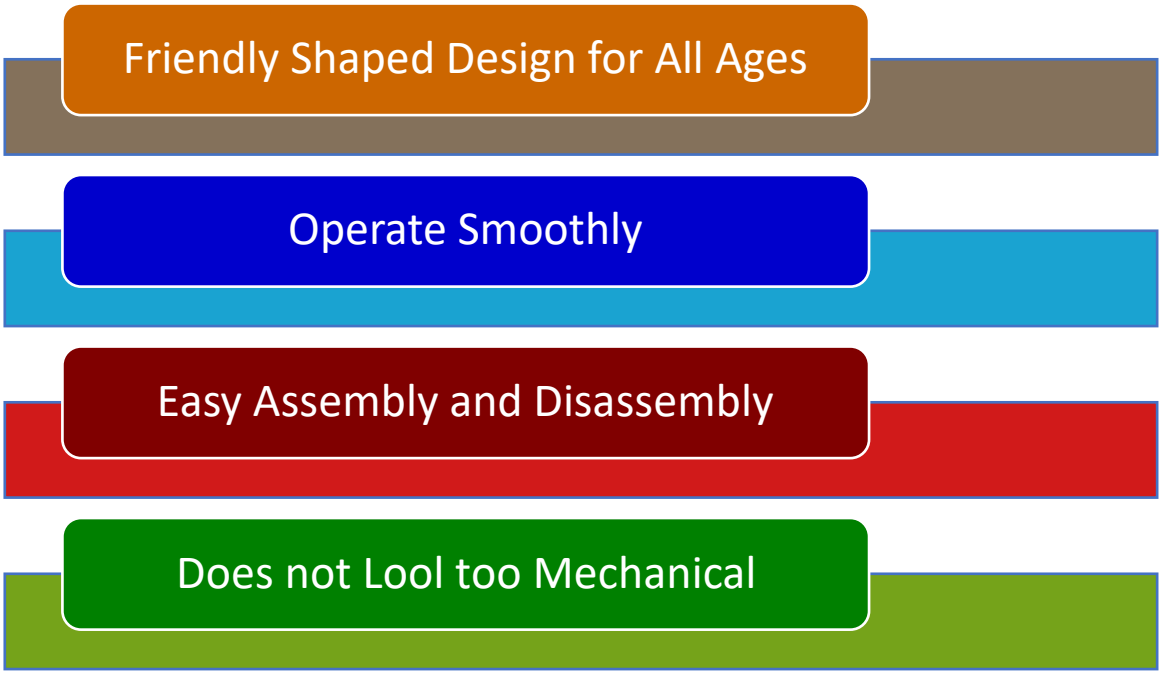


Figure 56 Mechanical Design Goals

However, a lot of considerations regarding aesthetics was needed, aesthetics as well as design-strength considerations. This robot had to look good but should also be durable and solid. Besides that, we have materials that can withstand wear and tear while not adding a lot of weight.

How design process and steps were done:

1. Brainstorming, sketching and discussing different designs to select the most appropriate one.
2. Creating 3-D Model design on CAD software such as SolidWorks to simulate internal components weight and space.
3. Selecting appropriate materials to achieve both solid stable and good-looking robot
4. Testing wheels and observe its movement and stability to make sure the robot will move smoothly
5. Prototyping and 3-D printing the body

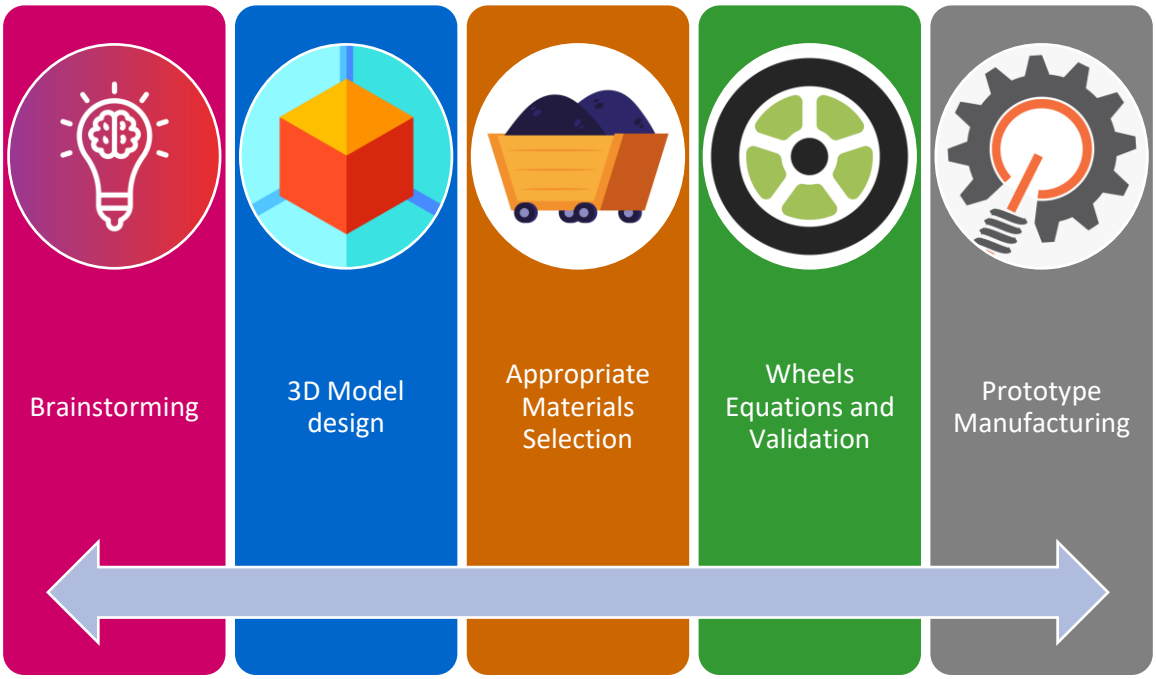


Figure 57 Robot Design Process and Steps

The chapter is divided into Motor Design, Wheel Calculations, Weight Calculations, Internal Design and External Design.

3.2. Conceptual Design

To ensure an optimal functionality and safety, the companion robot's conceptual design was thoughtfully developed with several significant aspects in consideration. Therefore, as illustrated in section 2.2.1 above, many designs were considered; however, the optimal choice was the circular-shaped base. Henceforth, the circular-shaped robot had been chosen, as it would preferably provide effortless movement and navigation indoors. In addition, while taking safety into consideration, the robot would be safer to operate in various circumstances due to its circular shape which aids in sharp edges elimination, which lessens the possibility of accidents and injuries. Furthermore, the robot's entire surface area is reduced by obtaining the circular base design, which enhances movement efficiency and manoeuvrability in restricted environments.

The robots compact and medium sized dimensions achieved an equilibrium between portability and functionality. This size selection ensures the robot's sustained mobility, rigidity, and lightweight design which facilitates the transportation of the robot when necessary. Moreover, these qualities also improved the robot's entire usage and versatility in a variety of contexts,

whether it is stationary or mobile for varied activities. The combination of these design characteristics covers practical concerns like mobility and safety while also fulfilling the functional needs of providing solid and smooth navigation.

3.3. Detailed Design

3.3.1. Motors specifications and Weight constrains

The companion differential drive robot motors were carefully selected based on the complete system weight, target speed, required torque and power required to make the robot durable. The used software to size the motors efficiently without wasting energy was made by ROBOTS SHOP community helped to determine the optimal specifications that the robot will require for maximum performance and effectiveness.

The overall weight of this robot was calculated by summing the masses of the single components, as listed in Table 15. An approximate weight was determined to be around 8 kg, weighing most essential electronic and mechanical components, such as the Arduino Mega, Jetson Nano, IMU, LiDAR, power supplies, motor drivers, display, outer cover, base plates, and structural fixings.

The total weight is thus significant for calculating the amount of torque and power needed by the drive motors. The design of the system guarantees that the motors that have been selected will bear the corresponding load while ensuring movement on different types of terrains.

Table 18 Component Weight Distribution

Component	Weight (kg)
2 Arduino Mega	0.10
Jetson Nano	0.15
Battery	1
IMU	0.0005
Power Bank	1.5
Lidar	0.19
2 Motor Drivers	0.06
Mic	0.5

Speaker	0.52
Screen	0.497
Outer Body	2.5
Interior Body	1.2
Rods and Fixtures	0.37
Total weight	**7.957 ≈ 8 Kg**

Motor selection criteria

After calculating the total weight, to achieve the desired movements, the following constrains, and parameters were used to calculate the motors specifications:

- Total mass: 8 kg
- Number of drive motors: 2
- Radius of drive wheel: 0.425 m
- Robot Velocity: 1 m/s
- Maximum incline: 20 degrees
- Supply voltage: 12 V
- Desired acceleration: 0.2 m/s2
- Desired operating time: 60 mins

By applying these constraints and parameters the software output for each drive motor was the following:

- Angular Velocity: 91.08 rev/min
- Torque: 11.151 kg-cm
- Total Power: 21.878 W
- Maximum current: 1.8232 A

As shown in results, for the robot to operate efficiently the output torque should be at least 11.151 kg-cm and the revolution should be between 90 RPM -100 RPM.

Selected motor Specifications

- Nominal voltage: 12 V
- Free-run speed at 97 RPM
- Free-run current at 0.09A
- Rated speed at 80 RPM
- Rated current at 0.3A
- Rated load torque at 3.5 kg.cm
- Rated load power 3.6W
- Stall current at 2 A
- Stall torque at 13 kg.cm
- Gear ratio: 1:87

The selected motor for the differential drive robot is JGY-370-1285 Miniature Worm Gear Motor taken to account weighing, torque calculation, and the power required. The selected 12V DC gear motor with 97 RPM and 13 kg.cm torque provides for reliable performance and efficiency.

3.3.2. Internal Design

The internal skeletal frame of the companion robot adheres to a modular design philosophy, whereby a series of stacked plates create the main support for major electronic and mechanical sub-assemblies as shown in figure 59. The plates are spaced by spacers illustrated in figure 60, which are mounted to the bottom plate, creating a structurally sound and clutter-free internal layout. This modular structure provides not only the assembly procedure in a straightforward manner, but also access to all components, upgrading, and maintaining. A skeleton as such favors a companion robot as it can have an exceptionally lightweight yet robust skeleton, through which multiple sensors, actuators, and processing modules can easily be fed without any other extra burden.

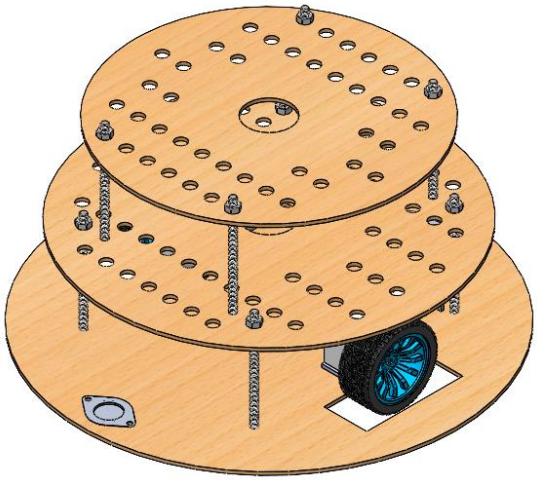


Figure 58 - Modular internal design of companion robot.

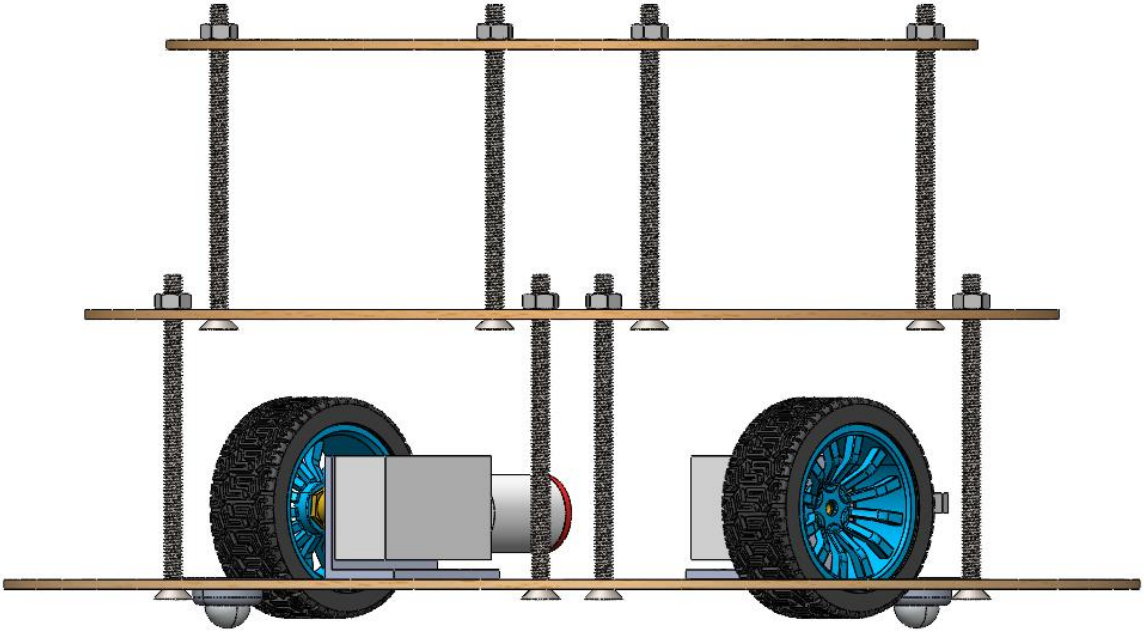


Figure 59 - Plates assembled using spacers.

One aspect of this skeletal structure is the gradual decrease in plate size as they go upwards, lending the robot an overall tapering or oval configuration. This geometric structure aids in the aesthetics and form factor since it maximizes the use of space without compromising a low center of gravity. The variable spacing between plates further enhances the design's flexibility

since it can be configured differently based on the target application or component configuration. Regardless of whether additional battery packs, more processing units, or better cooling systems are accommodated, the variable plate spacing future-proofs the robot and makes it compatible with various upgrades and revisions.

Also, the design ensures optimal distribution of weight because heavier units, such as the motors and power supply, stay at lower levels, thus, increasing stability. Spacers improve structural integrity, allowing clearance for routing cables, cooling air, and neatly securing delicate electronics without crowding. This thoughtful design makes robots more resistant to mechanical stresses with a clean and advanced internal setup. The stacking plate modularity, therefore, gives an additional advantage of design scalability for any possible expansions or modifications intended with minimal design effort.

The technical drawing provided illustrated in figure 61 of the internal skeleton of the companion robot; it presents front, side, and top views, with all major dimensions that were considered vital for concerned structural integrity and movement. The separation distance between the wheels is 224.60 mm, obtaining a stable and well-balanced base for movement. This spacing has been optimized to obtain an adequate balance between manoeuvrability and support since it will allow the robot to navigate efficiently while maintaining control during turns. The total height of the internal structure is 169.00 mm, which was derived from the number of modular plates stacked. This height allows for the appropriate amount of vertical clearance between the levels for the placement of various components ensuring easy access for maintenance, wiring, and integration of sensors and electronics.

The 350.00 mm diameter circular base of the robot enables a compact and sufficiently large footprint for carrying along internal components, while still giving an agile touch in an indoor environment. The design itself fully accounts for the caster wheel mounting, which includes a precisely located hole of Ø25.40 mm for this purpose; this hole placement guarantees smooth load and stability, avoiding any unwanted tilting and ensuring that the robot moves in balance. Structured layering and thoughtfully spaced criteria altogether lead to an efficient modular robotic platform towards nimble and reliable movement for future adaptiveness.

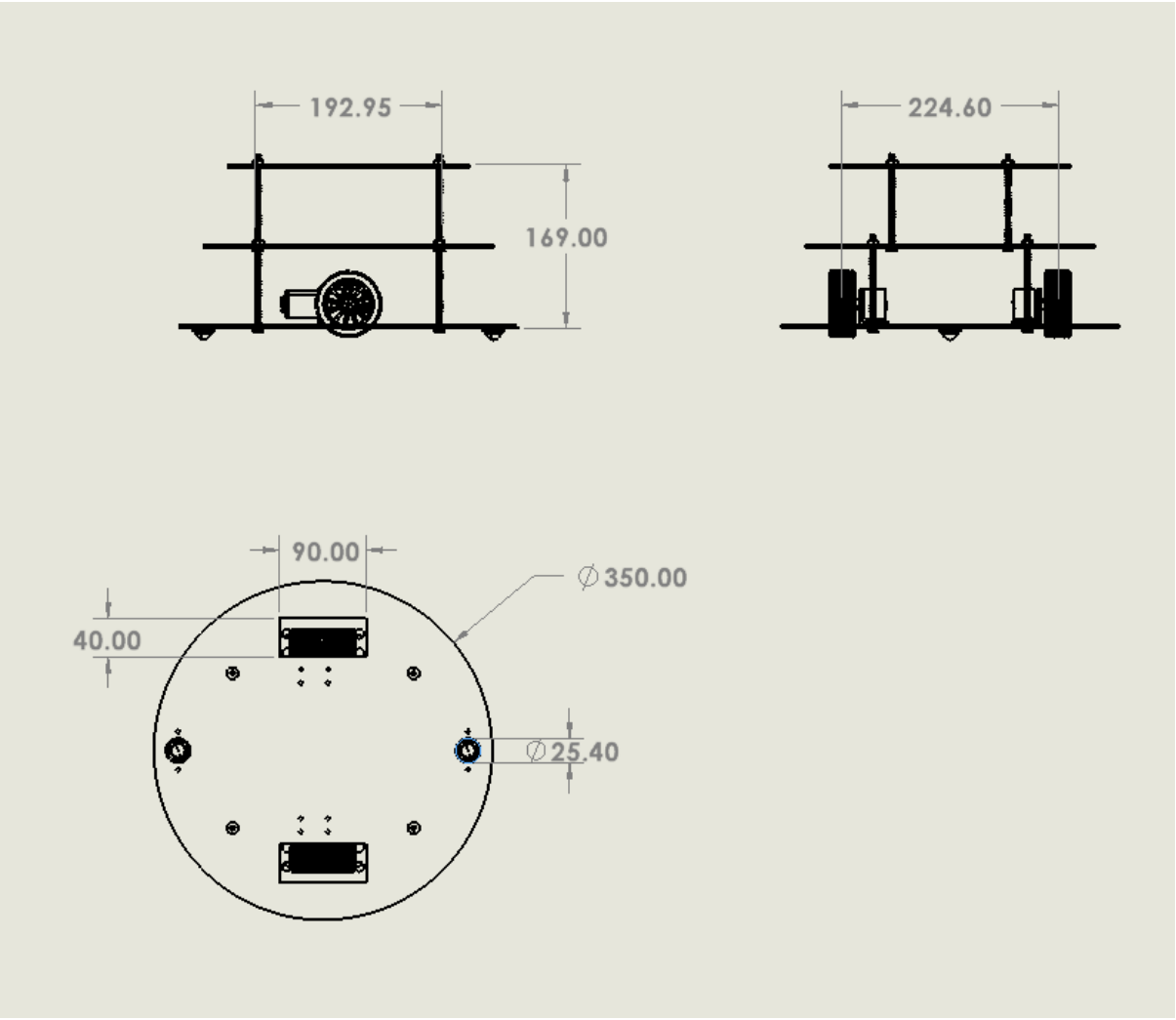


Figure 60 - Assembly drawing of internal design

3.3.3. Outer Design

The external design of the companion robot is a mix of modern aesthetics, functionality and easy maintenance. Every aspect of its form has been considered carefully to create a companion robot that is both visually appealing and practical for everyday use. The companion robot has a sleek, modern and minimalist design. It features curved body that improves both aesthetics and functionality. The outer shell is cut to give an inviting, friendly appearance which makes it ideal for use in home and professional environments. The robot is designed with no sharp edges to make it safer for all users of different ages.

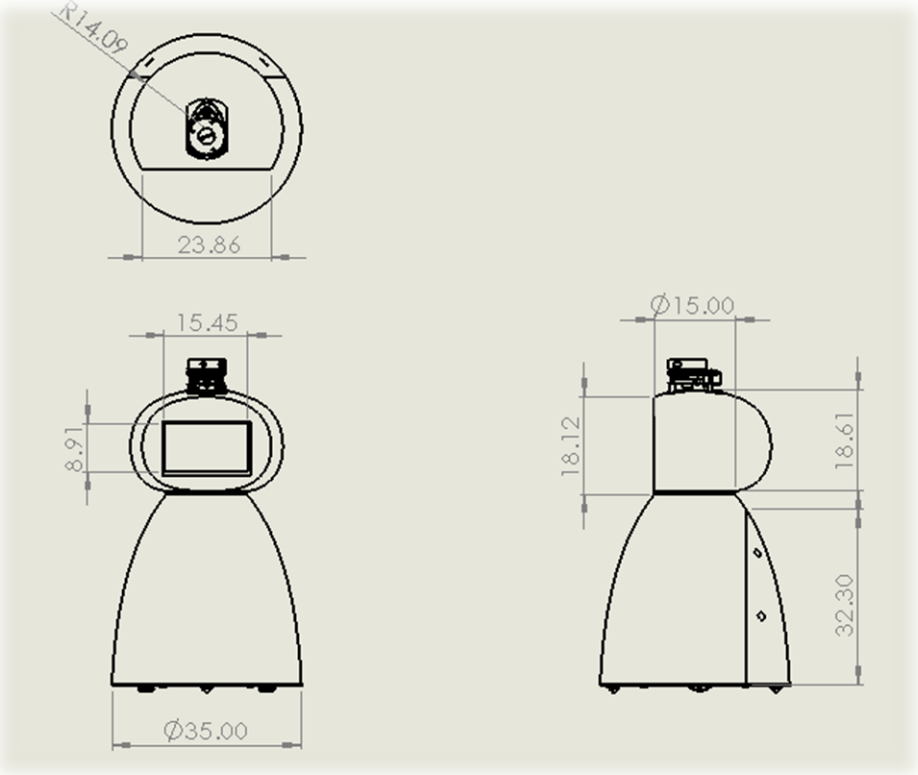


Figure 61 Drawing of the Robot

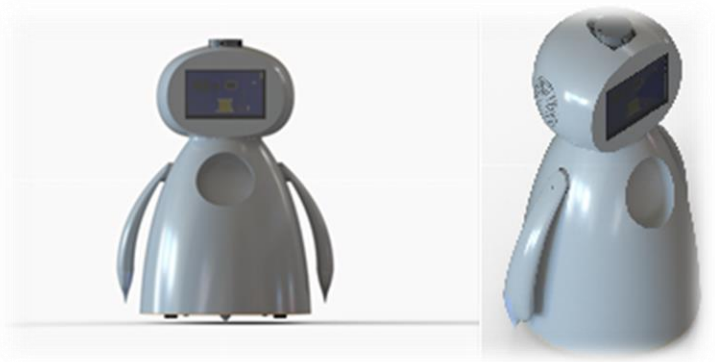


Figure 62 Project Outer Design

The robot head is rounded and compact, it features a rectangular display on the front. This screen serves as the main interaction interface. The display screen is considered the primary graphical user interface (GUI) and serves also as a facial interaction with different emotional expressions and status updated. The smooth edges and slightly curved screen placement contribute to a more natural and engaging user experience. On top of the head is a LiDAR.

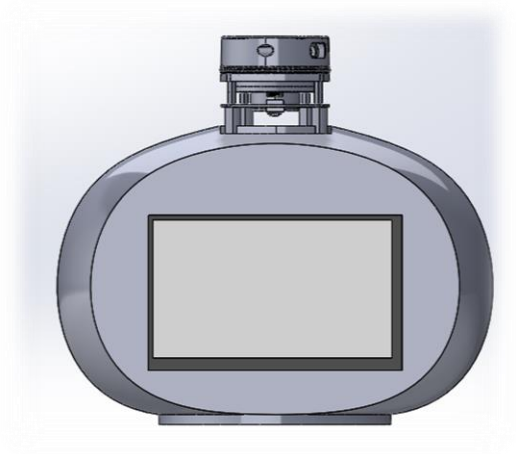


Figure 63 Robot Head Design

To ensure easy assembly, disassembly and troubleshooting heat inserters were integrated into the robot body with considering hole diameter tolerance, wall thickness and hole size referring to data sheet of heat inserters. These heat-inserters allow for a modular construction, allowing easy access to internal components. This approach substantially lowers maintenance time and simplifies part replacement when necessary. The base of the robot is integrated into the design with small caster wheels providing smooth motion.

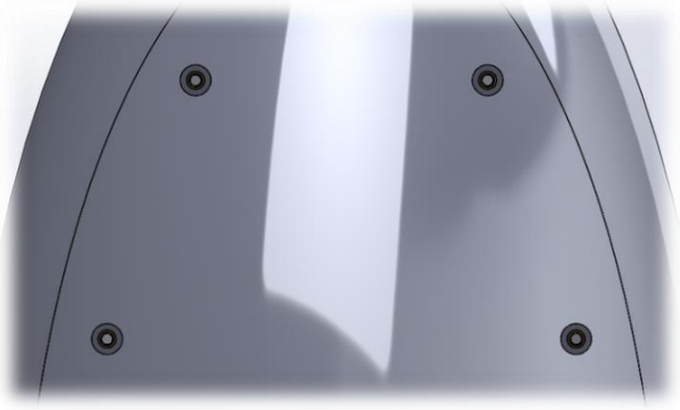


Figure 64 Heat Inserters Position in the Back of the Robot Body

The external design of our companion robot is a perfect combination between aesthetics, functionality and ease-of-use. Its curved and minimalist form enhances user interaction.

3.4. Stress and Finite Element Analysis

The Finite Element Analysis (FEA) is critical for confirming the structural soundness and mechanical reliability of IRIS companion robot's components under expected loads. The basic FEA considers the worst-case design situations. All key elements of the robot's mechanical subsystems, including the Head, Body, Arms, and Base, must function safely and as intended in the established use cases. In accordance with industry standards, design to withstand fail-safe load conditions without crushing or excessive deformation while user interaction retains stability and reliability attributes.

The design loads applied for this analysis were based on the technically accurate weight of the parts. Relevant safety factors were introduced to incorporate unknowns, dynamic contingencies, and variabilities of the material. Notably, the head can carry a load of the screen (0.497 kg), LiDAR sensor (0.19 kg), and speakers (0.3 kg), which is equal to around 0.987 kg. The total weight, with a safety factor of 3 will determine our equivalent design load (30 N). Design loads confirm that the head structure is sufficiently strong to maintain stability and reliability, while maintaining a tolerance for unexpected shocks and dynamic limitations. The same construct holds true for the arms, with the load being able to sustain up to 1.5 kg with safety factor applied in our equivalent design load of 17 N.

Material properties were imported from ANSYS's engineering database and limited to project specific material data. The basic material used in assembly is Polylactic Acid (PLA) with 30% natural fibers, based on sustainability, intermediate strength, and lightweight. The primary material properties considered include density, Young's modulus, Poisson's ratio, and yield strength in order to accurately simulate the mechanical response between the processed components to the design loads and boundary conditions.

This section included a mechanical analysis of the four main components, the Head, the Body, the Arms, and the Base, of the IRIS robot. The components were examined using FEA modeling software for total deformation, equivalent stress (von-Mises), and equivalent total strain under the respective design loads and boundary conditions, to determine if the overall design is a good mechanical design and to ensure all parts have sufficient safety factors to

remain a safe operating condition and to provide some mechanical robustness to the interactive and assistive capabilities of the companion robot.

3.4.1. Material Properties

The natural fiber reinforced PLA has adequate stiffness and reasonable strength needed to support the overall lightweight components of the IRIS robot. A density of 1.3×10^{-6} kg/mm^3 contributes to a lightweight robot. Young's Modulus of 5255 MPa suggests reasonable resistance to elastic deformation due to an applied load. The Poisson's Ratio of 0.3899 suggests that, unconstrained, the PLA with natural fiber reinforcement will expand laterally under axial stress. This is important to note for determining stress distribution.

A Bulk Modulus of 7954.9 MPa demonstrates possible resistance to uniform compression for when the components experience multi-directional forces (e.g. forces a boundary condition rather than a displacement). The Shear Modulus of 1890.4 MPa will represent the material's potential resistances to shearing forces that are respectively relevant to the arms as it is plausible that the arm's internal torsional stresses are due to directed loads. The material has a Tensile Ultimate Strength of 701.55 MPa and a Tensile Yield Strength of 68.41 Mpa, which demonstrates the material should reliably perform when used within the confines of the expected loading conditions.

The Isotropic Secant Coefficient of Thermal Expansion of 0.000109 1/°C, which is not in focus of this study, would mean that change of dimension under these temperature ranges are minor. Overall, this PLA composite provides a satisfactory state of low bulk density, moderate stiffness and strength levels to meet the structural performance requirements of the IRIS robot. The addition of natural fibers in the PLA composite material choice, contributes to the overall life cycle of the project to be environmentally responsible by reducing reliance on petroleum-based polymers. The FEA results show that all structures evaluated perform against failure modes well within known safe limits, thus demonstrating elongation or deflection of the materials will be suited for the loading conditions of this project.

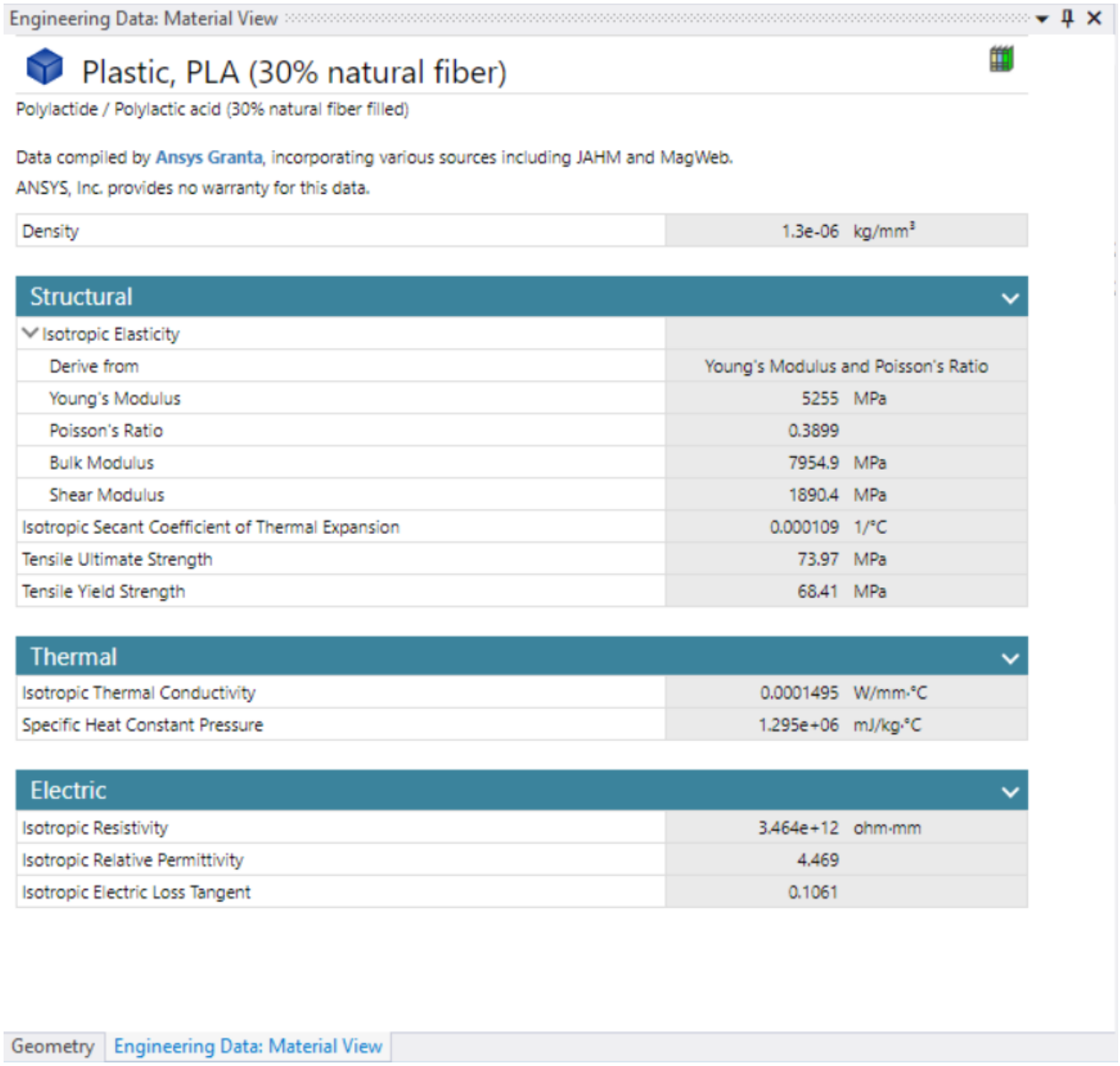


Figure 65 Material Properties of PLA

3.4.2. Simulation Setup

Finite element analysis (FEA) of the mechanical structure of the IRIS companion robot was conducted using ANSYS 2024 R2, a powerful and comprehensive simulation package that performs high-fidelity structural analysis. The software will include advanced meshing capabilities and solver options that enabled torque, stress, strain and deformation evaluation across the Head, Body, Arms and Base components in operational loading conditions that are realistic. Material properties designated to all components are aligned with the PLA (30% natural fiber filled) configuration outlined in Section 5.7.2.

3.4.3. Meshing Strategy

An identical meshing approach was completed for all the components to facilitate an accurate and stable numerical solution, while monitoring computational assets. For all three-dimensional parts (Head, Body, and Arms) of the IRIS, SOLID185 tetrahedral solid elements were used. The flexibility of tetrahedral solid elements allows for excellent approximations of stress contrasts and distortions within complex shapes like the IRIS's, while maintaining numerical efficiency. Mesh quality was verified using conventional aspects with aspects such as aspect ratio and skewness produced in ANSYS. Checks made sure that the aspects achieved agreement with quality thresholds to minimize the risk for numerical instabilities.

In the Head assembly (see Figure 66), a relatively fine mesh was used, particularly in areas that would be anticipated to have stress concentrations such as the screen mounting area and sensor cutouts in order to get localized stress and deformations correctly. Areas where it is expected that lower stress gradients are present were less dense to improve computational time with minimal impact on accuracy.

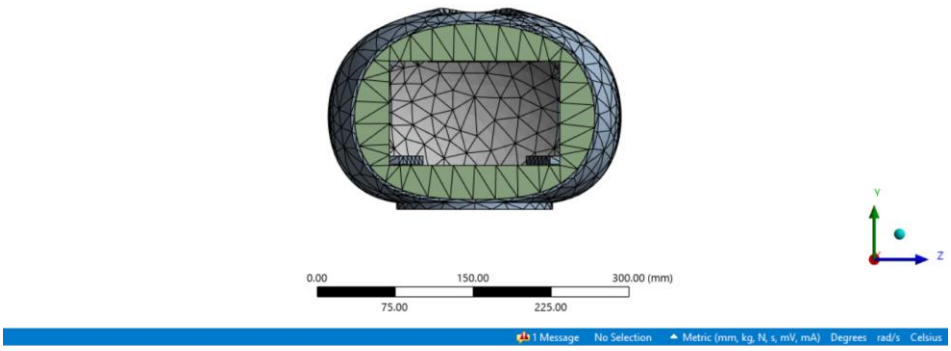


Figure 66 Meshing of Head

With the Body component (see Figure 67), a similar meshing strategy was implemented, although additional mesh refinement was included along the upper ring of the head structure where it meets up with the body. The refinements were required to accurately simulate the load transfer from the head to body and to apply and assess the changes to any stress concentrations.

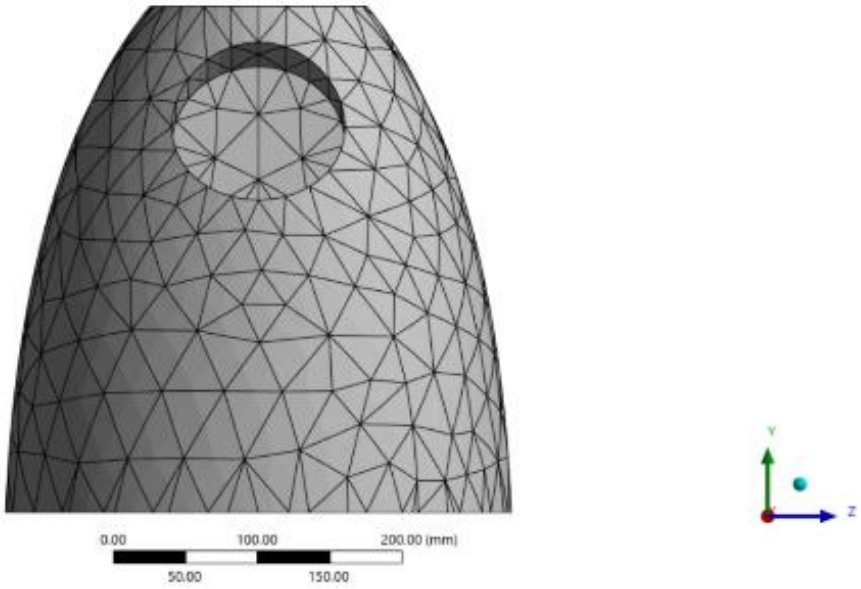


Figure 67 Meshing of Body

In the case of the Arms (see Figure 68), the meshing strategy was to capture bending behavior and stress distribution through the length of curved profile of the arm structure. Without consideration of local stresses, a finer mesh density was applied at the free end of the arm where loads are anticipated to produce more stress and deformation.

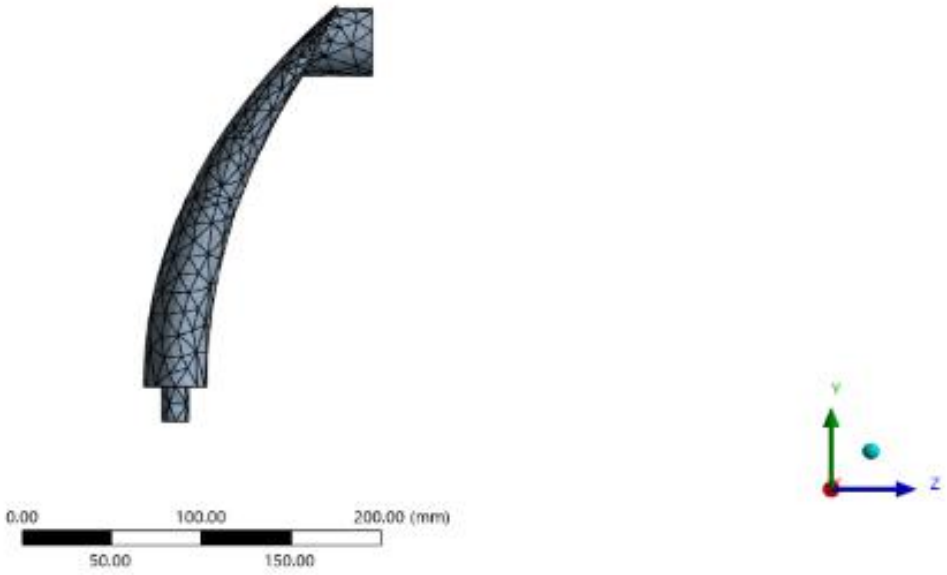


Figure 68 Meshing of Arm

The Base (see Figure 69) was modeled with two-dimensional triangular shell/surface elements that are suitable for thin, flat structures with large cutouts. The mesh was reasonably refined in the area of the geometric discontinuities (rectangular and circular holes) to effectively capture stress concentration areas. The quality of the mesh was checked to ensure that element distortion and skewness were within suitable constraints.

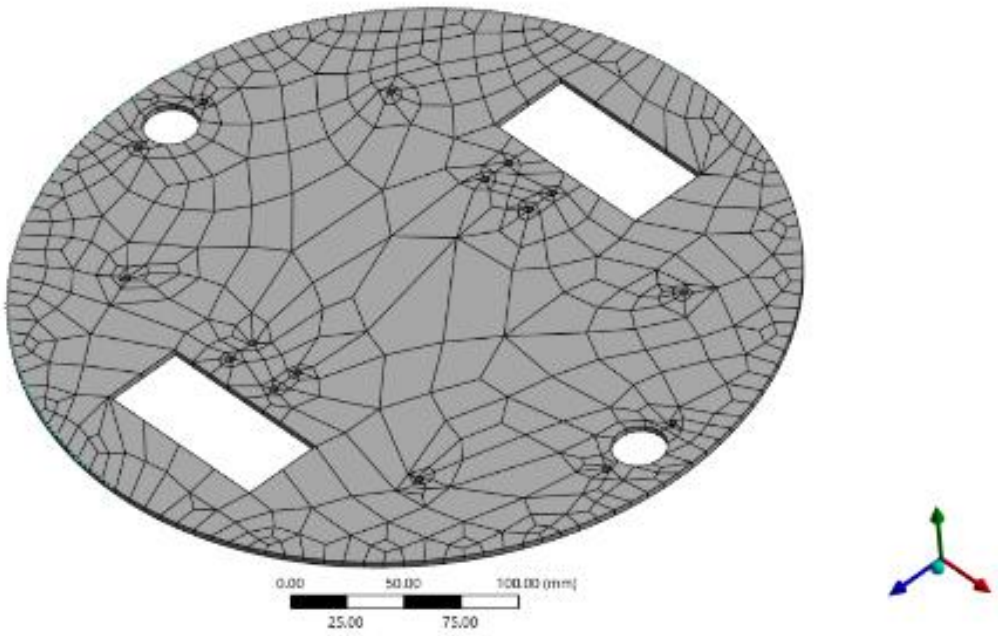


Figure 69 Meshing of Base

3.4.4. Boundary Conditions and Loading Scenarios

The boundary conditions were defined to simulate the realistic mechanical constraints that would be present in the operational environment of the robot. The Head's lower face was bounded in all translational and rotational degrees of freedom, which correspond to a rigid mounting of the Head to the torso. The entire weight of the screen (0.497 kg), LiDAR sensor (0.19 kg), and speakers (0.3 kg) is 30 N. Since the engineer's 'safety factor' would be a conservative 3, an arbitrary load was placed at the screen mount interface. The load was distributed at four places (7.5 N, 3 N, 3 N and 3 N) at what were assumed to be realistic points of load transference through the screen housing (Figure 70).

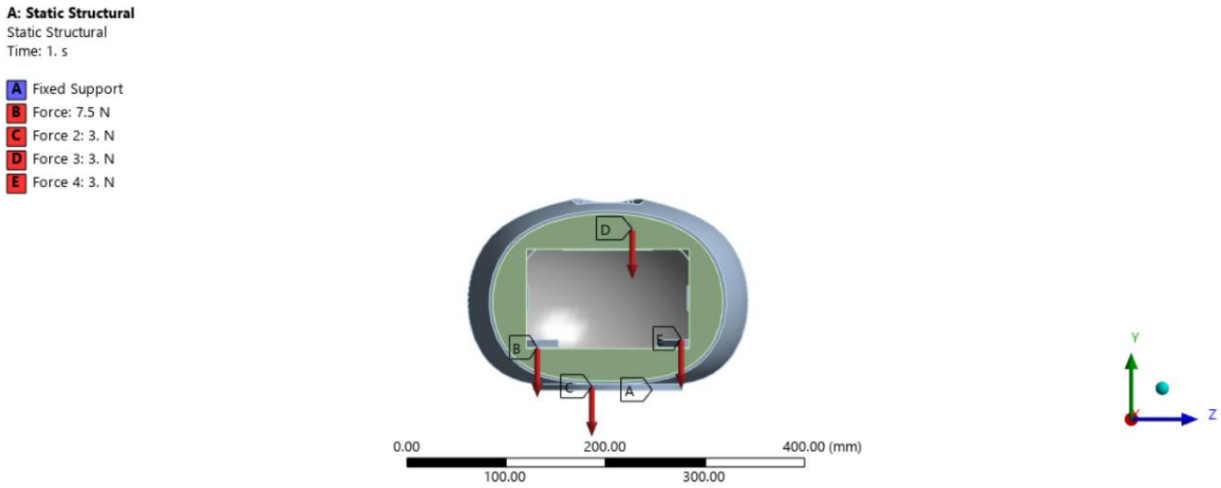


Figure 70 Loading and Boundary Conditions of Head

The Body structure was modeled with a fixed support located at the lower ring, to represent its connection to the base assembly. A vertical load of 30 N was applied at the upper opening of the body structure representing the total load from the head assembly, including the safety factor mentioned previously (refer to Figure 71).

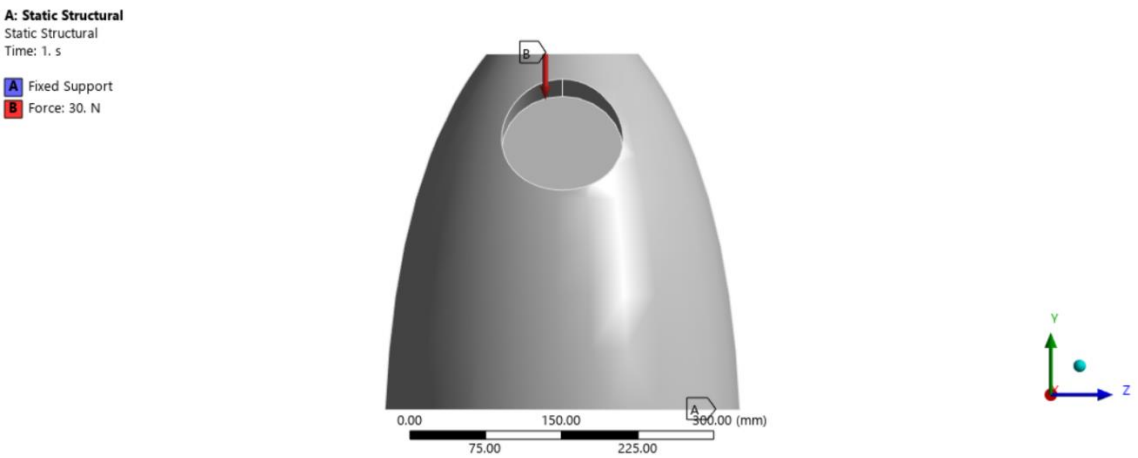


Figure 71 Loading and Boundary Conditions of Body

In the case of the Arms, a fixed support was applied at the base of each arm to also simulate its connection to the torso. A downward load of 17 N was applied at the free end of the arm representing the design load capacity with safety factors applied (refer to Figure 72).

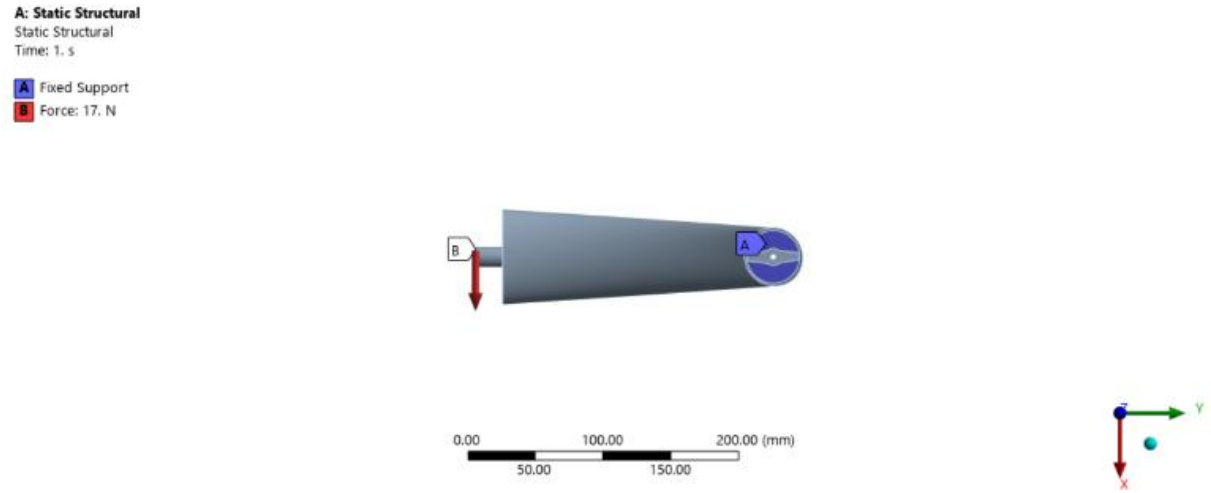


Figure 72 Loading and Boundary Conditions of Arm

Finally, the Base is modeled with fixed supports applied at multiple locations along the perimeter to simulate bolted connections to the chassis. A concentrated vertical load of 93 N was applied at the central location of the base representing the combined load of the upper assemblies (head, body, arms, and internal electronics) including a conservative safety factor of three (3) (refer to Figure 73).

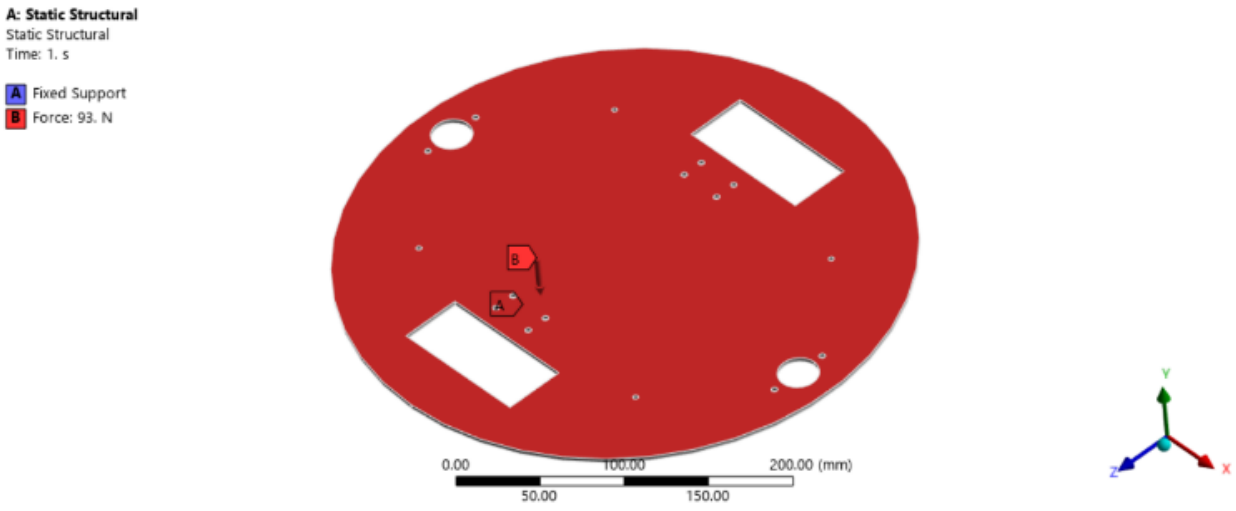


Figure 73 Loading and Boundary Conditions of Base

The calculations of the design load are based on the actual mass of the robots components, but the loads are scaled by a safety factor of 3 to cover dynamic loads, tolerances for manufacturing and any unforeseen operational loads. The head assembly load is calculated as screen (0.497 kg) + LiDAR (0.19 kg) + speakers (0.3 kg) which has a total mass of approximately 0.987 kg, including a safety factor of 3 results in an effective load of 30 N (0.987 kg × 9.81 m/s² × 3).

The arms, with an estimated mass of 0.57 kg would assume a design load of 17 N (0.57 kg × 9.81 m/s² × 3). The upper assemblies, which included the head, arms and have electronics, have an estimated mass of about 3.1 kg which corresponded to a design load of 93 N (3.1 kg × 9.81 m/s² × 3) to be applied to the base.

These load calculations and their points of application, ensured the simulations were a reflection of operational performance in a realistic environment when accounting for safety factors of design.

3.4.5. Finite Element Analysis Results and Discussion

This section presents and discusses the results of the finite element analysis (FEA) applied to the mechanical structure of the IRIS companion robot. The FEA analysis includes the total deformation (total def.), equivalent (von-Mises) stress (Total stress), and equivalent total strain (Total strain) for each of the largest four components: Head, Body, Arms, and Base. This information is critical to validate the mechanical integrity of a design subjected to operational loads and to allow not only functionality but also safety in real-life situations. All of the aforementioned simulations were carried out utilizing ANSYS 2024 R2 and implementing the material properties of PLA (30% natural fiber) as described in Section 5.7.2.

a) Head

Figure 74 shows the total deformation in the head structure assuming an applied load of 30 N distributed on the screen mounting region. The maximum total def. recorded is 0.00628 mm and is occurring around the screen mounting points where the load is applied. This deformation is extremely small relative to the total overall geometry of the head structure, indicating excellent stiffness and dimensional stability under operational loads.

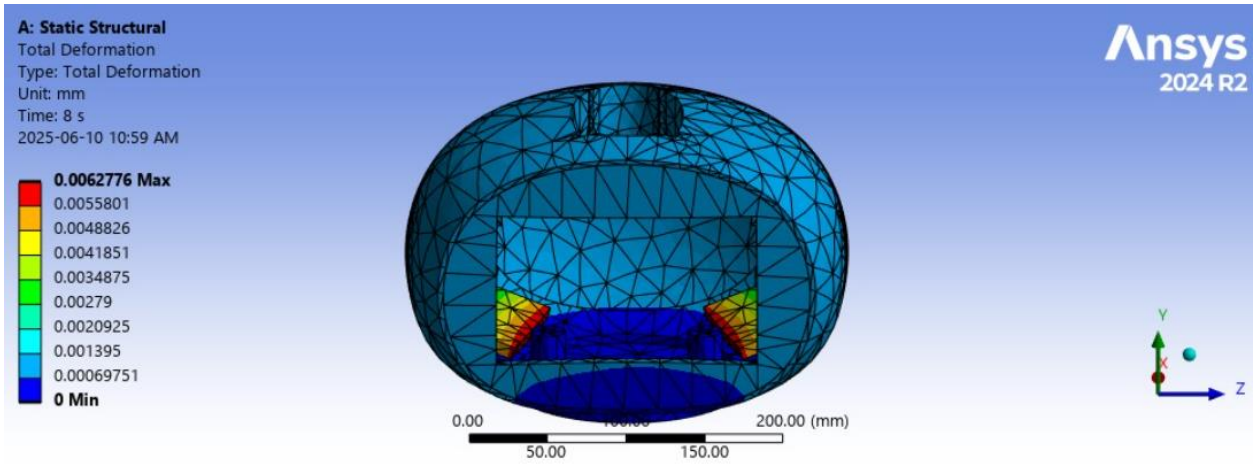


Figure 74: Head Total Deformation

The equivalent von-Mises stress results, illustrated in figure 75, show a maximum stress of 0.071067MPa occurring in-determinately near the screen mounting region. This is consistent with the stress concentration due to load application, and it is a reasonable order of magnitude consistent with the expected mechanical behavior of the structure. The maximum stress is fairly small given that the yield strength of PLA is 68.41MPa, so the maximum stress is smaller than the elastic limit (and remains elastic). A maximum of 0.071067MPa represents approximately 0.1% of the yield strength of the material.

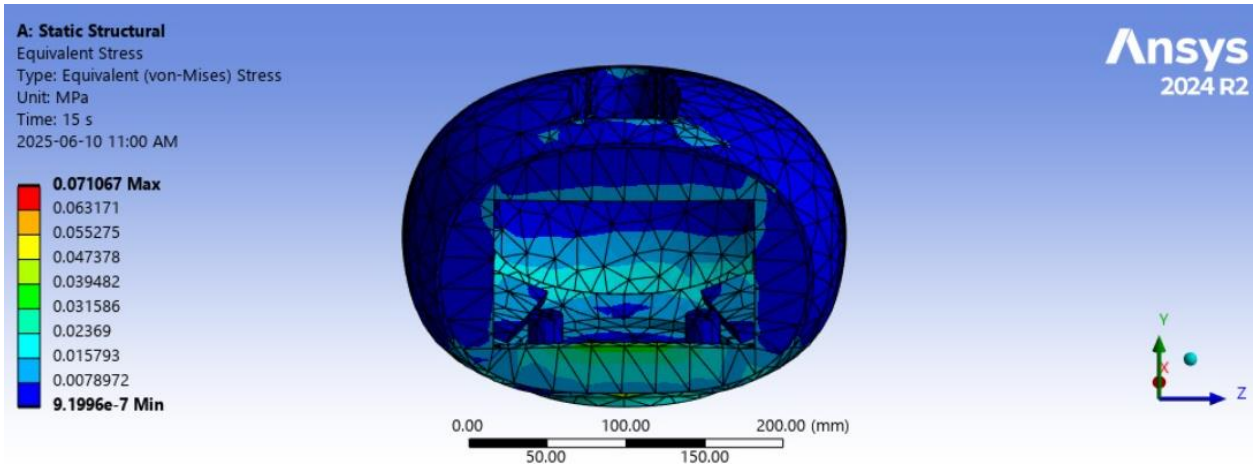


Figure 75: Head Equivalent (von-Mises) Stress

The maximum equivalent total strain (shown in figure 76) in the head structure is 1.4388×10^{-5} mm/mm, it generally occurs in-determinately near the screen mounting region, or 0.0014%. Since the typical elongation at yield for PLA is ~6-8%, this total strain value is significantly smaller, and the consistency of the low strain over the entire structure indicates that the head design is sound, and should not deform excessively under normal operational conditions.

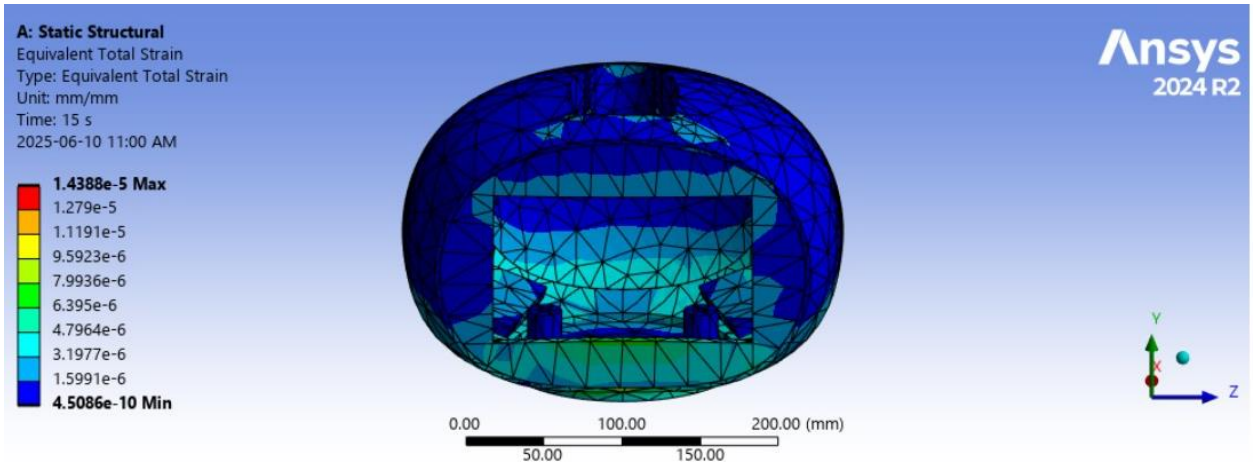


Figure 76: Head Equivalent Total Strain

If we assume the maximum von-Mises stress value (0.071067MPa) and if we know the yield strength of PLA (68.41MPa) we can estimate the minimum factor of safety.

$$\text{Safety Factor} = \frac{\text{Yield Strength}}{\text{Max Stress}} = \frac{68.41}{0.071067} \approx 963$$

- Given this high safety factor, it is clear that the head structure is designed with a substantial margin of safety, ensuring reliable and safe performance for a reasonable range of unintended loads.

b) Body

Figure 77 shows the total deformation of the body structure under a vertical load of 30 N applied at the upper opening. The maximum deformation is 0.0087224 mm, mainly concentrated around the top ring where the head connects. The low magnitude of deformation indicates the structural stiffness of the body, and its ability to support the load transferred from the head assembly.

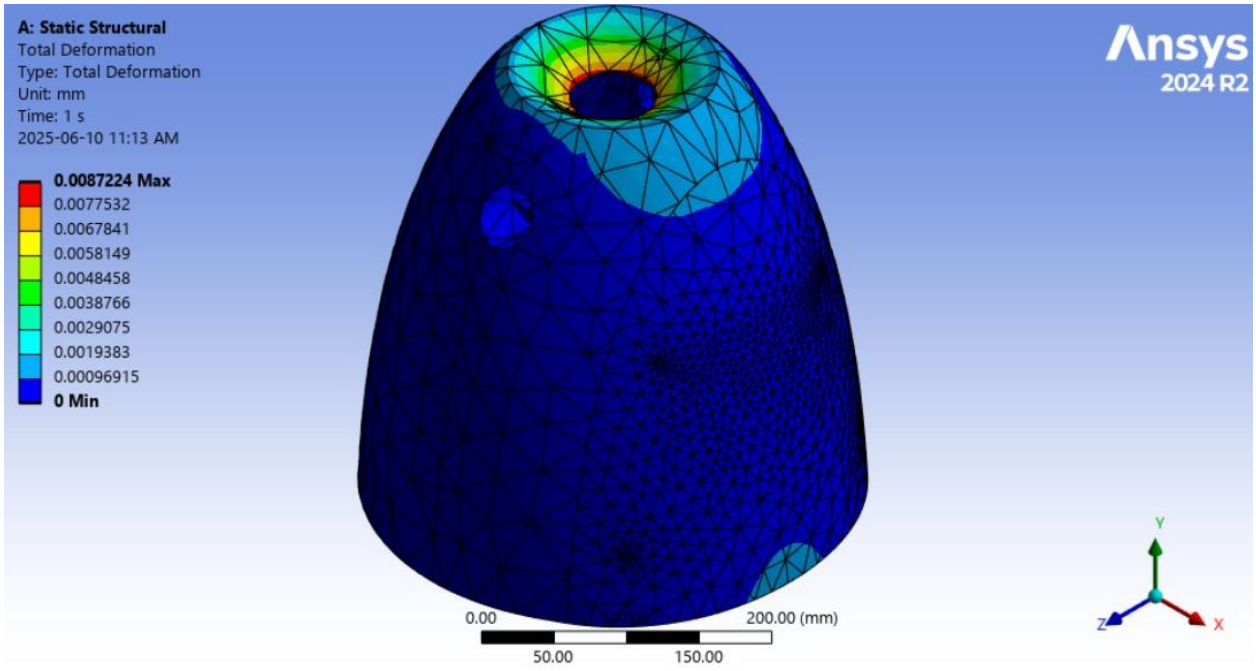


Figure 77: Body Total Deformation

The equivalent (von-Mises) stress results shown in Figure 78 have a maximum of 0.1576 MPa, located at the upper opening where the load is applied. This stress is about 0.23% of the yield strength of PLA (68.41 MPa), confirming the body structure is well within safe operational limits and is not likely to yield or fail under the load applied.

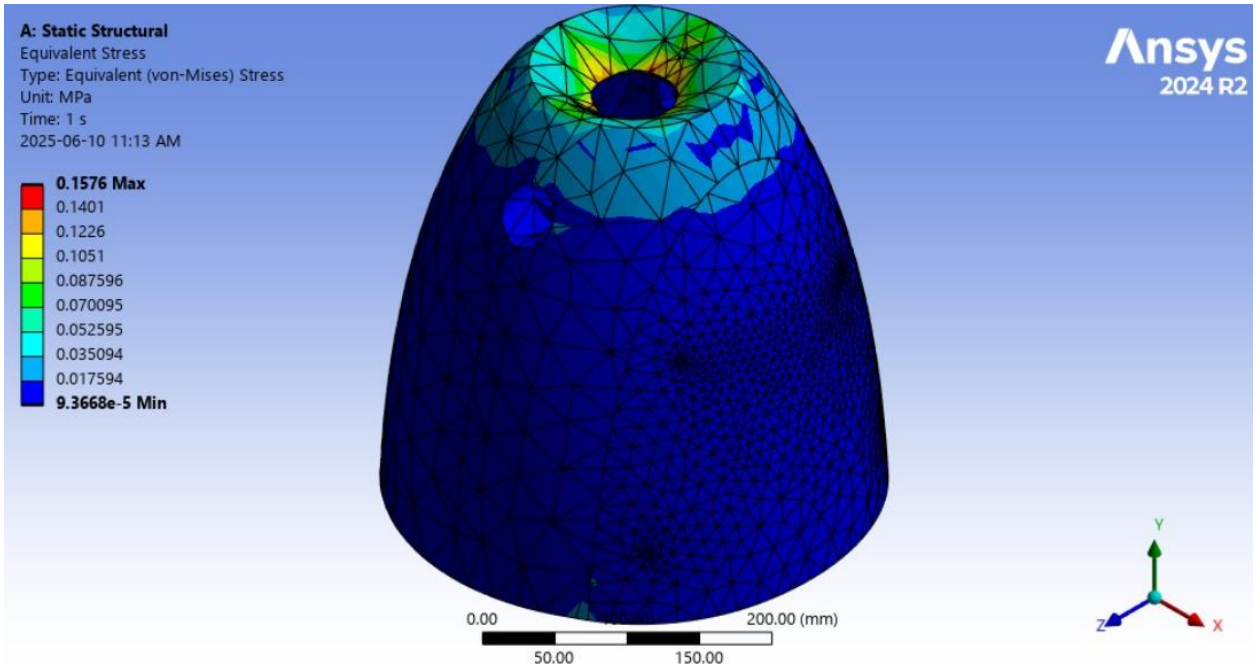


Figure 78: Body Equivalent (von-Mises) Stress

The equivalent total strain distribution shown in Figure 79 has a maximum strain of 3.2711 × 10^{-5} mm/mm (0.0033% strain). This is comfortably within the yield strain of PLA indicating that the body structure is still in the elastic stage and can take operational load without permanent damage.

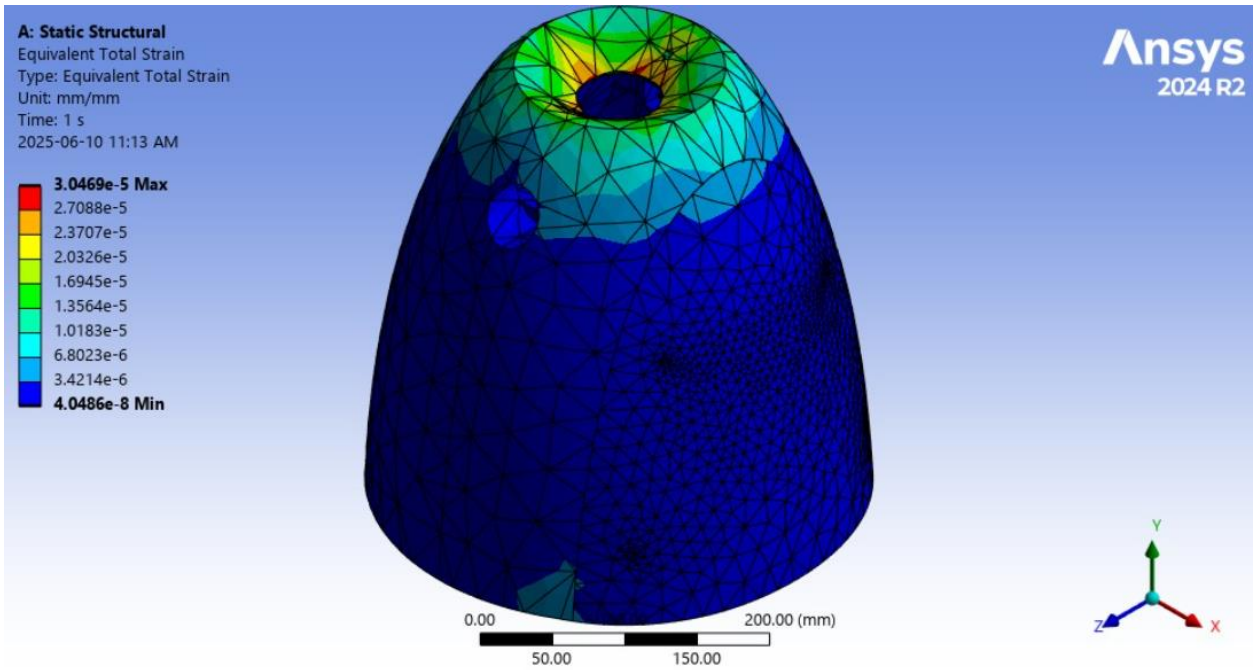


Figure 79: Body Equivalent Total Strain

The measurements indicate that loads applied to the head are effectively transferred through the body structure's upper portion and into the fixed lower supporting portion. The stress

distributions have a seamless gradient, indicating that there are no sudden high-stress regions (i.e. stress concentrations) that may ultimately lead to fatigue or cracking in the future. Therefore, the body structure is in good condition, and will support loads as expected.

c) Arms

The arm structure resulted in a maximum deformation of 0.5318 mm at the free end which is consistent with the free end where the 17 N load is applied (Figure 80). This is a reasonable deformation considering the lightweight nature of the arms and the load that was applied. The deformation distribution correlated well with the bending behavior of the arm due to vertical loading.

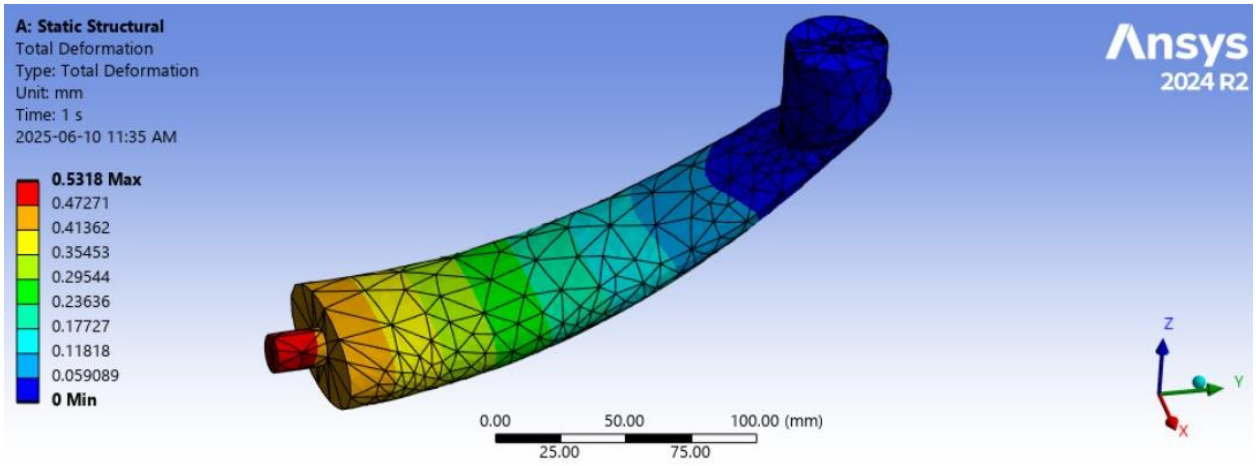


Figure 80: Arms Total Deformation

The equivalent von-Mises stress results are shown in Figure 81. The maximum value for equivalent von-Mises stress is 3.2467 MPa, located near the free end of the arm, as predicted, at the base of the arm where it will be expected to have its greatest stress concentration. This figure corresponds to ~ 4.7% of the PLA yield strength, indicating that the arms have an ample safety factor, and the chances of material failure during standard use and the limitations of the testing procedure are low.

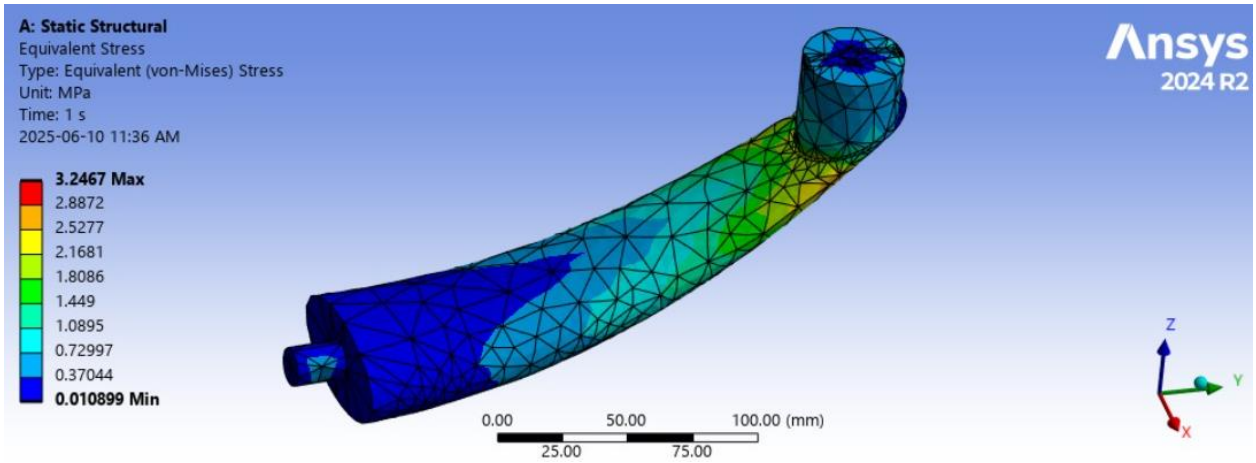


Figure 81: Arms Equivalent (von-Mises) Stress

The equivalent total strain information given in Figure 82 provides a maximum strain of 0.00062196 mm/mm (0.0622% strain) which is well below the PLA yield strain indicating that the arms operate in the elastic region, meaning they can be relied upon, and will not lose their mechanical integrity over time.

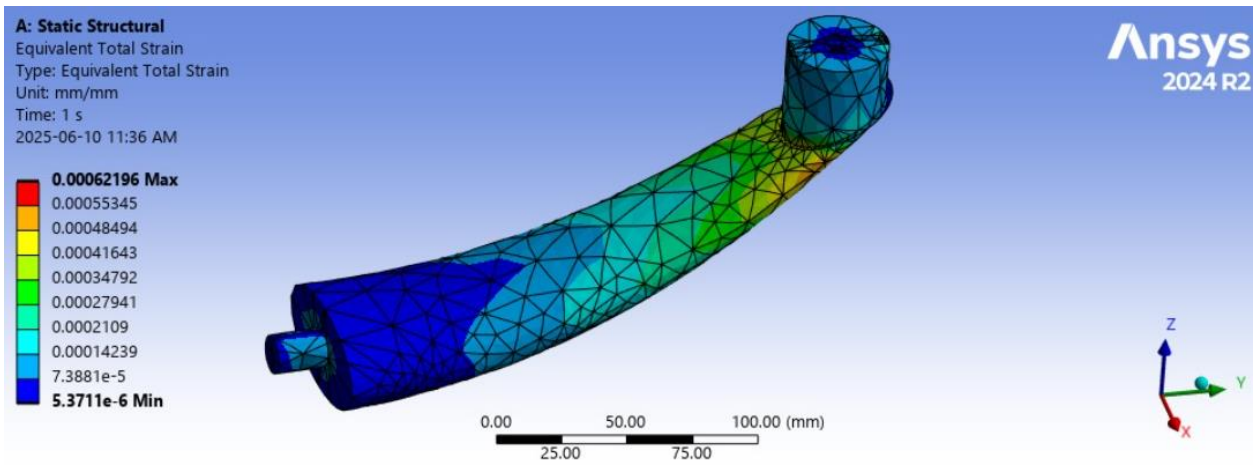


Figure 82: Arms Equivalent Total Strain

In light of the arms flexible design, and the moderately large observed deformation, we recommend that dynamic effects - such as vibrations, as well as hitting potential impact loads - need to be considered in future designs. However, the current design demonstrates adequate performance under static loading conditions, ensuring that the arms are mechanically robust and capable of supporting expressive gestures without risk of failure.

Dynamic Load Factors Considering the arms' relatively flexible design and apparent deformation, it is suggested that dynamic effects (vibrations and impact loads) are also included in future designs. Ultimately, the branches have satisfactorily exhibited their performance

under static loads and are mechanically resilient and capable of supporting expressive gestures under reasonable conditions without risk of failure.

d) Base

A load of 93 N has been applied to the base. As shown in Figure 83, the maximum deformation of the base structure is 9.4067×10^{-7} mm, which is essentially insignificant. The deformations are significantly minor confirming that the base is extremely stiff and can adequately support the upper structure without affecting its form or alignment.

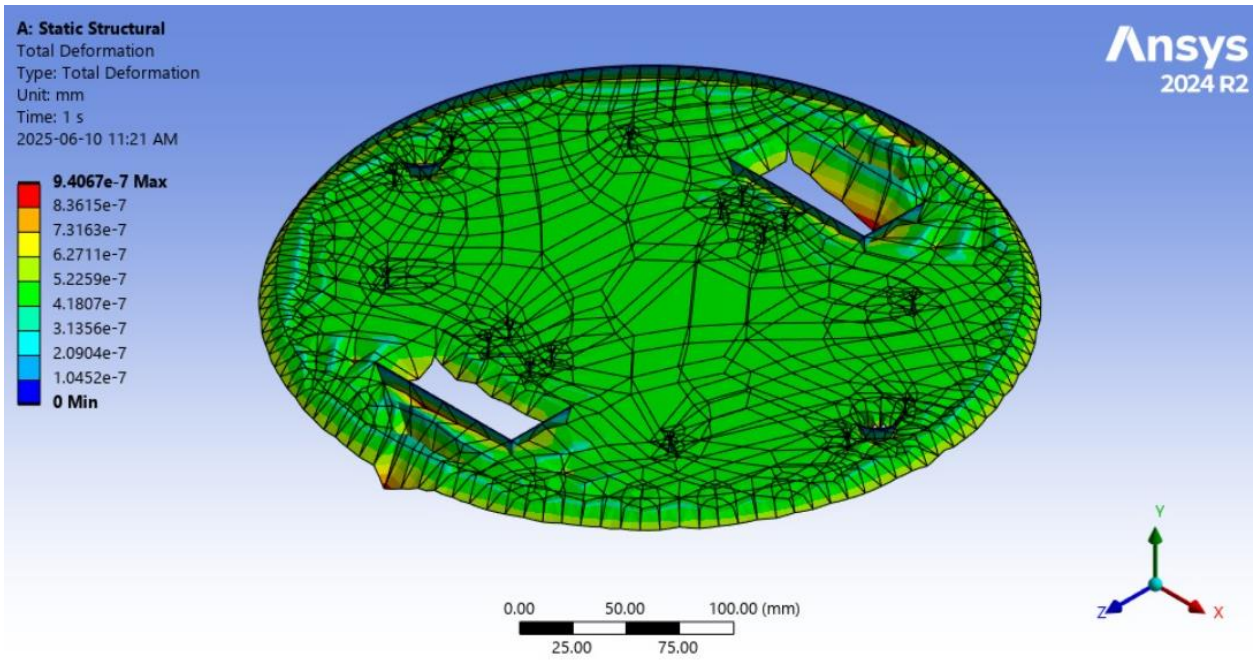


Figure 83: Base Total Deformation

Equivalent von-Mises stress results show a maximum stress of 0.0012192 MPa (see Figure 84), less than 0.002% of PLA yield strength, clearly demonstrating that the base structure is extremely strong and can distribute load properly and safely well below any mechanical limits.

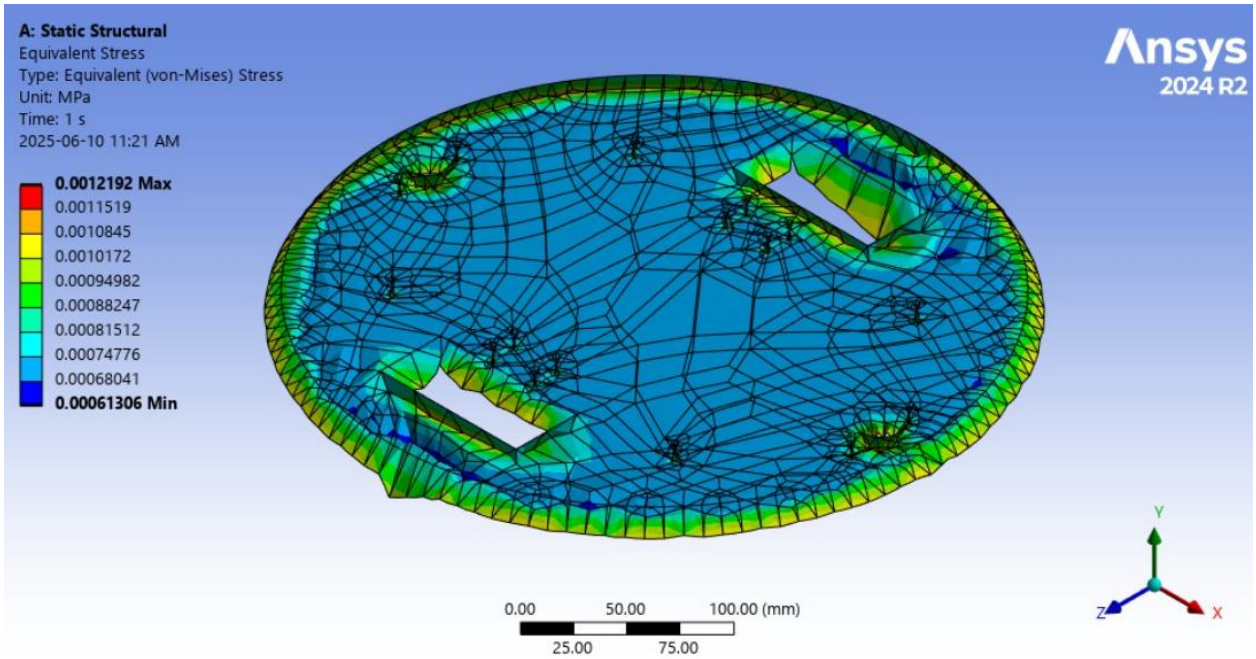


Figure 84: Base Equivalent (von-Mises) Stress

The equivalent total strain data show (see Figure 85) a maximum strain of 1.961×10^{-7} mm/mm or 0.0000196% strain. Again, this strain is extremely low, which shows that the base structure is still comfortably in the elastic region with operation exposure to operational loads. As a result, the robot base configuration should not undergo any stress or mechanical failure through operational service.

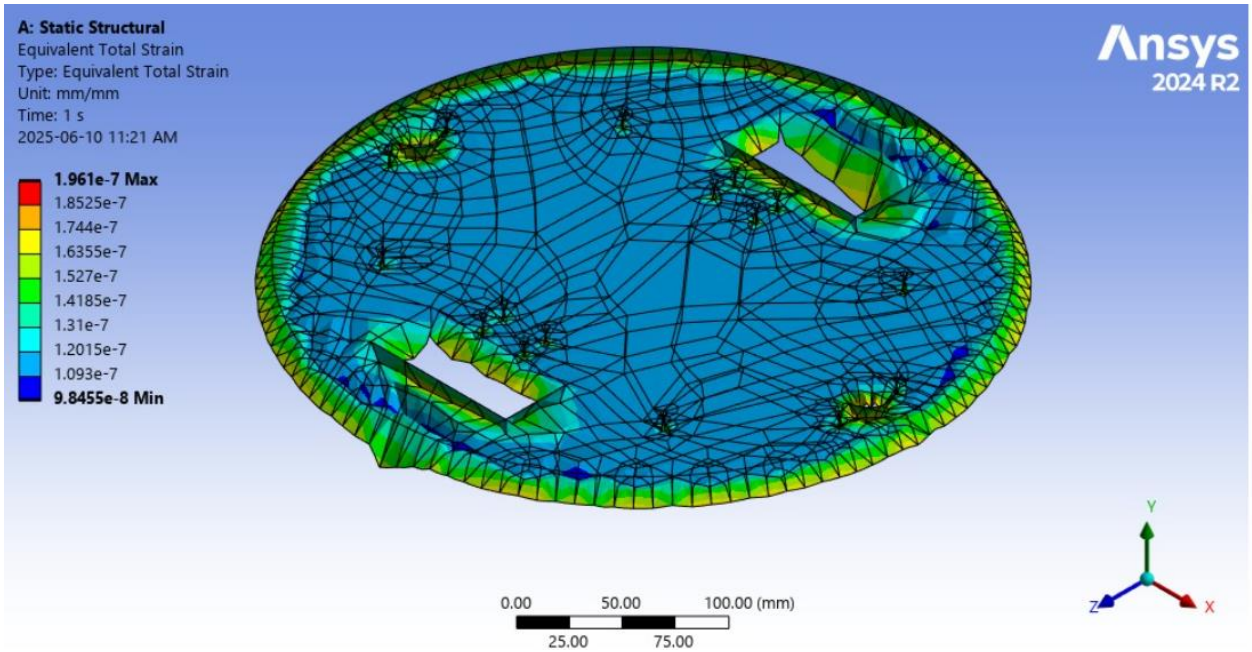


Figure 85: Base Equivalent Total Strain

The stress and strain results indicate a uniform distribution of the applied load with respect to the base structure geometry and a successful transfer of the upper assembly load to the assembly supports. The very limited deformation and stress means that the base structure will remain stable under working conditions when a robot assembly mass is positioned on top of it and provide a reliable operating platform for the robot. Stability is critical to overall mechanical stability of the IRIS companion robot to avoid tipping the robot or oscillation movements while in motion.

Finite Element Analysis (FEA) of the mechanical structure of the IRIS companion robot (head, body, arms and base) shows a robust and reliable design that meets the functional and safety specifications discussed in previous sections. Each part was tested under conservative load cases and used appropriate safety factors in assuming structure will remain intact during use.

The results indicate that all respective components function well within the yield strength limit of chosen scaffolding material (PLA finger material with a yield strength of 68.41 MPa), and the von-Mises stress values were determined to be low. For instance, the highest stress occurs in the arms section, with the maximum stress having a value of 3.2467 MPa, which equates to just shy of 4.7% of the yield strength. Similarly, all deformation values remained acceptable, which means well below values which would degrade mechanical stability, or lead to material fatigue.

Overall, the mechanical design of the IRIS companion robot is structurally strong and ready for the field as all components analyzed have strains and stresses within the confines of the material limits. This assessment validates the design, is a great starting place for future optimization, and development.

3.5. Manufacturing and Fabrication

The IRIS robot hardware housing and structural elements were manufactured using additive manufacturing methods with emphasis on Fused Deposition Modeling (FDM). Figure 86a illustrates the slicing using Cura software, where the CAD design of the head structure is sliced to convert the design to toolpath layer by layer instructions for 3D printing.

The image illustrates the breakdown of individual segments and the buildup of necessary supports, as required when printing complex geometries that have overhang and complex features.

After the digital preparation the printing was performed on a desktop FDM printer as shown in Figure 86b.The image illustrates the deposition process of PLA filament with natural fiber reinforcement on the build plate. The FDM printing process builds parts in layers and in essence builds each part strand by strand of filament. This takes place layer by layer building the part to represent the digitized design, or ideal, with each pass of filament making the part come to reality.

The PLA material can be effectively designed with sufficient mechanical abilities to ensure the components in consideration have desirable mechanical strength, which was ensured by life predictions through FEA assessments in previous sections of the report, while additionally being lightweight aiding the mobility of the robot.

Figure 86c shows the final assembly of the IRIS robot. The 3D printed parts (including the head, torso, and arms), along with the electronics and interactivity, are combined in Figure 86c to show the completed assembly. The additive manufacturing process allowed not only for rapid prototyping and design iteration but also for complex geometries which other manufacturing processes would find difficult and costly to reproduce. The results highlight the benefits of additive manufacturing on a functional, ergonomic, and aesthetically pleasing design toward improving the robot's movement or operation and user acceptance of the robot.

a) b) c)

Figure 86: Depiction of phases of manufacturing. a) design on slicer, b) outer body being printed, c) fully assembled robot.

3.6. Conclusion

Last but not least, the mechanical design of the companion robot has been achieved successfully as a lightweight and circular based structure, first, smooth navigation, and safety indoor movement optimization. As mentioned before, the circular-base of the robot was initially chosen for its smooth edges, reducing the risk of accidental impacts; in addition to safety, the circular base offers the robot a high mobility and a high manoeuvrability while autonomously moving and to maintain less space covered by the robot. In addition, the material

selected was contributed to maintain a yield strength below 20 MPa, which ensures that the base would survive any operational loads. The finite element analysis FEMA stress results demonstrates that the applied forces do not exceed the safety limitations of the material, which assures the mechanical design's effectiveness.

Moreover, the robot would be built by the utilization of additive manufacturing, ensuring a sustainable manufacturing process bye material waste reduction, precise fabrication, and the ease of customization. Therefore, design has successfully met the primary objectives as being a lightweight and compact robot, yet a robust and efficient one. Henceforth, limitations and challenges such as the weight optimization while maintaining and efficient localization and navigation system Where is solved due to the material selection and the structural enhancement applied on the room with mechanical design. Furthermore, the stress analysis results above illustrate that the robot could overcome any unexpected load without failure or sudden impact, ensuring long-term durability. Additionally, the mounting points and cutouts in the robot were essential components for excess weight reduction while maintaining the strength and the functionality of the robot.

Finally, despite these results above, limitations may exist, such any minor deformation of the robot's inner or outer parts under high impacts. Therefore, future enhancements could be applied, such as further material investigation, reinforced structural elements, or weight reduction through the optimization of topology. Moreover, experimental techniques would be applied in real-world simulations for ensuing an effective and flawless performance of the robot.

Chapter 4 : Electric and Electronic Design

Electrical and electronics form the structure of the IRIS companion robot that consists of mechanical motion, sensory, and intelligent feedback. IRIS is created for several uses, specifically to assist older adults and individuals on the autism spectrum.

IRIS utilizes the subsystems to allow for safe, emotionally intelligent, and socially interactive performance. It consists of (1) Arms System, (2) AI System, (3) IoT Monitoring System, and (4) Autonomous Navigation System. The Arms system has servo joints driven by an Arduino Mega and PCA9685 driver to achieve human motion. The AI System is configured to connect to a Microphone, Speakers, and an OLED Display, powered by a Jetson Nano or Raspberry Pi in such a way that it can allow for expressive, emotionally aware communication.

The IoT Monitoring System is based on an ESP8266, and can be connected to the current battery unit's regulated 9V supply, and is able to connect multiple sensors for temperature, humidity, air quality sensors, and flame sensors. The Autonomous Navigation System is composed of a Raspberry Pi for computing intensity and coordination, a Teensy microcontroller, an IMU, and LiDAR Sensors, powered again, by a regulated, reinforced controlled 12V battery supply. The features above form IRIS's capability to locomote and display communicative behaviours reliably.

Together, these subsystems create a dynamic, adaptable platform. In the following sections, each system is analyzed in detail, exploring design considerations and integration strategies that make IRIS a safe and emotionally engaging companion robot.

4.1. Design Methodology and System Overview

A schematic presentation in Figure 87 of the electrical and electronics design for IRIS was created using Fritzing software. Fritzing is a public service and open-source software that allows individuals to create schematics with incredible detail, as well as printed circuit boards (PCBs). Each subsystem (Arms, AI, IoT, Navigation) was also developed in a modular fashion to support integration and troubleshooting.

The systems of IRIS have been outlined as four primary subsystems:

- **Arms System**: The arms subsystem uses an Arduino Mega and a PCA9685 to allow for two high torque servo motors controlled through the use of a dedicated battery pack, for the arms to operate as specified.

- **AI System**: The AI system uses a Jetson Nano or Raspberry Pi 4 to run speech processing as well as sentiment processing, and user interface contained in a microphone, OLED, and speaker.
- **IoT Monitoring System**: The IoT system consists of an ESP8266 NodeMCU connected to a DHT11, MQ-2, and flame sensors, all controlled through a voltage circuit for stable and safe operation.
- **Autonomous Navigation System**: The system consists of the main computer (Raspberry Pi 4), a teensy microcontroller for hardware connectors, two JGY-370-1285 motors through 2 dual H-Bridge controllers, an MPU6050 IMU, LiDAR sensor (if possible) all powered through a 12V battery pack to maneuver in order to increase interactivity.

By ensuring each subsystem effectively plays its own part towards the operational desired outcomes of IRIS this modular design is maximized. Figure 4.1 shows the overall integrated architecture of IRIS, which highlights how these subsystems integrate and seamlessly operate as one.

More specifically, IRIS integrates high-performance computer systems which also serve AI and navigation, but include Jetson Nano and Raspberry Pi 4 CPUs, respectively. The actuators are faithfully controlled by PCA9685 and dual H-Bridge motor drivers. The sensors used are DHT11, MQ-2, flame sensor, MPU6050 IMU and LiDAR. In addition fixed power supplies and voltage regulators give reliability to the entire system and will protect the aforementioned components.

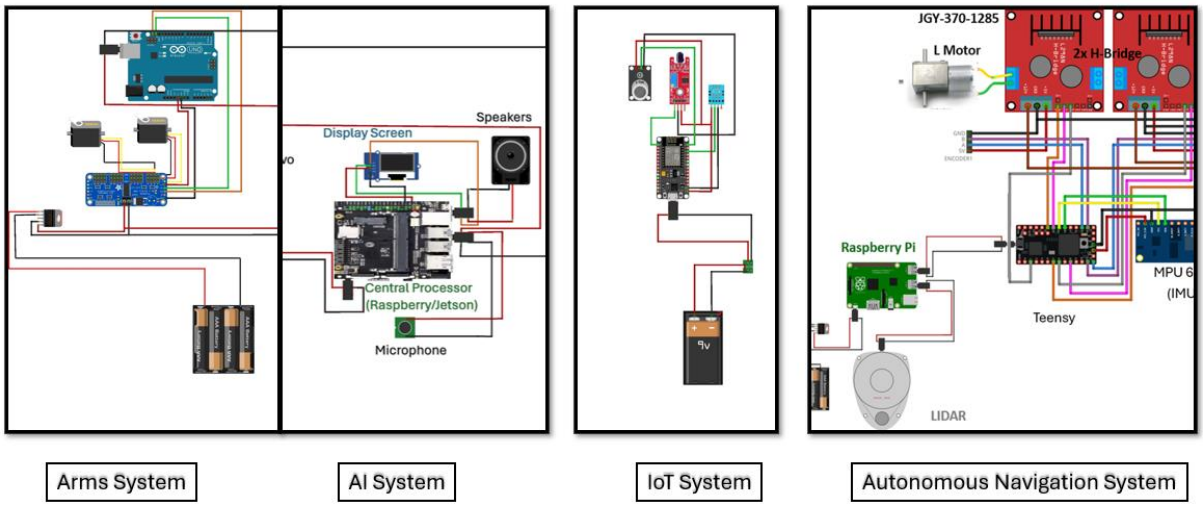


Figure 87 - Wholistic Diagram of Electricand Electronic Circuit

Figure 88: High-level overview IRIS electrical and electronics architecture, showing the integrated Arms System, AI System, IoT Monitoring System, and the Autonomous Navigation System.

4.2. Results and Analysis

4.2.1. Power Supply Strategy

The development of a safe power strategy is significant to IRIS because it will have to combine the multiple subsystems, as it will have to incorporate Arms, AI, IoT Monitoring, and Autonomous Navigation. Each of these systems has voltage and current profiles that are different, so it is critical that a safe power strategy is developed to ensure the systems work correctly.

- Arms System

The arms system employs an Arduino Mega 2560, PCA9685 PWM driver, and two high-torque servo motors. A 6 volt power supply (4xAA batteries) supplies sufficient current to facilitate dynamic gestures, while the Arduino control is separated with a logic voltage to prevent a voltage drop, and impact proper control of the HW.

- AI System

The AI System is powered by a regulated 5 V, 3 A supply source for the Jetson Nano or Raspberry Pi 4 which allows for peak loads occurring during speech processing and sentiment analysis. The OLED display, microhpone, and speakers each draw power from their own 5 V GPIO pin, with suitable grounding to avoid interference and distortion when processing and recognizing speech.

- IoT Monitoring System

Both the ESP8266 NodeMCU and the sensors (DHT11, MQ-2, flame sensor) required a 5 V and a 3.3 V supply. A 9 V battery was stepped down to the required voltage (LM7805 or AMS1117) and then regulated to supply 3.3 V for the NodeMCU. The grounds were all bridged to a single node. This ensured that all components shared a common ground path for a stable voltage reference.

- Autonomous Navigation System

This sub system is powered by two JGY-370-1285 worm gear motors. The battery pack is 12 V, and each motor is driven by an H-Bridge driver rated for peak currents (instantaneous) of 5 A. The Raspberry Pi power is separate from the battery because of potential motor voltage spikes, instead powered via USB-C cable. The MPU6050 IMU sensor and LiDAR sensors were

powered from the GPIO pins of the Raspberry Pi, which also incorporated decoupling capacitors to act as noise filters against the motors.se.

- System Integration and Communication

Grounding is carefully controlled across the subsystems to keep loops out and also to keep logic levels stable. The subsystems used two types of decoupling capacitors (100 nF and 10 µF) to prevent transient noise as well as short circuits in voltage levels.

This maintains subsystem separation while ensuring consistent reliable performances, safety, and the functions of IRIS there are in operating situations that could overlap in real world environments.

4.2.2. Arms System Circuit

The Arms System of the IRIS robot is intended to use physically expressive gestures to facilitate human-robot interaction with users (especially when considering older adults or users with Autism Spectrum Disorder). The Arms System allows the robot to perform fundamental gestures such as waving, greeting, and transmitting basic body language, which can further enrich emotional and social interactions for individuals with users.

The hardware implementation of the Arms System is ultimately based on the Arduino Mega 2560, which provides more GPIO pins than other boards on the market and also provides very good support for the implementation of PWM motor control. The implementation is supplemented with a PCA9685 PWM driver module, which allows up to 16 channels to be precisely and simultaneously controlled. This is critical for controlling a large number of servo motors. In this case, two high-torque MG996R servo motors are actuated to allow the arms of the system rotate and pivot with adequate torque and smooth, human-like motion to accommodate varying load conditions.

The arms circuit (illustrated in Figure 89) is a complete circuit showing the specific wiring and connections of the Arms System. The Arduino Mega communicates to the PCA9685 via the I2C bus. The servos are powered through a separate 6 V supply from a 4xAA battery pack. Because of the exclusive power supply, this circuit limited the risk of voltage fluctuations possibly interfering with logic level. The servos were attached to the PCA9685 to add stability and accuracy in control but also to minimize any electrical noise that would otherwise affect the main control system.

The Adafruit PWM Servo Driver Library is utilized for our servo motion control, allowing for precise angle control and the ability to adjust gestures in real-time based on user input or an AI-detected sentiment; any gesture can be stored in the Arduino's memory, thus allowing for sequences of gestures to be pre-programmed and automatically selected. The stored gesture can also be transitioned to smoothly and in sync with the AI system's dialogue and sentiment analysis.

Performance testing showed that the servos were able to achieve complete range motion (0 degrees - 180 degrees) with a latency of about 300 ms, which is reasonably closely aligned with the conversational pace of the AI system. Under steady-state operation, the servos consumed 1.8 A, with the peak instantaneous current nearly reached transiently at 2.5 A under rapid gestures. Continuous repeated usage resulted in no overheating or degraded performance from the system due to the dedicated power supply and effective decoupling techniques.

In conclusion, the Arms System connects seamlessly with the AI module through a serial communication link, enabling both gesture and speech output to be synchronized for a coherent, emotionally-desirable experience for the user. The arms circuit allows for strong functioning and safe power management, meeting the requirements outlined in the design challenge for interactive social robotics.

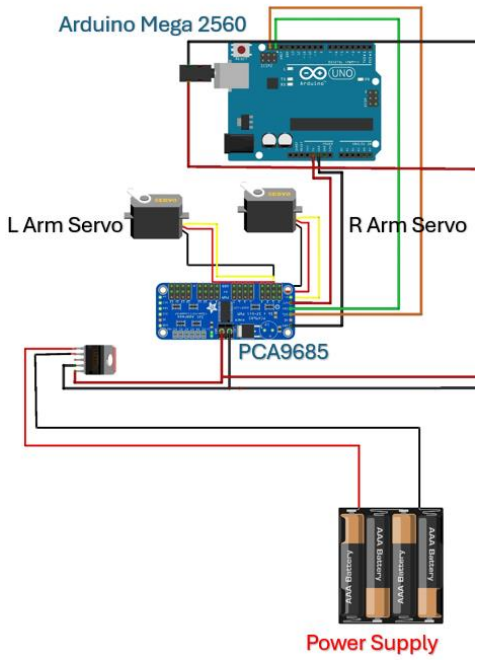


Figure 89: Detailed circuit diagram for the Arms System, showing the incorporated Arduino Mega 2560, PCA9685 driver, high-torque servo motors, and independent power supply.

4.2.3. AI System Electric Circuit

The intelligent interaction component is the AI System integrated in IRIS, which can apply real-time speech processing, perform sentiment analysis, and engage in real-time dialog generation during natural human-robot interaction. The sub-component uses a NVIDIA Jetson Nano device to provide this functionality based on its computational ability and ability to save energy.

The AI System outlines in the components and connections of the circuit in Figure 91 that allows for interactive, expressive responses. The Jetson Nano serves as the central processing unit connected to the audio inputs microphones, OLED display screen, and a speaker unit. The microphone takes in live audio which is subsequently processed to perform speech-to-text functionality using the whisper ASR mode which has also been implemented locally or through cloud service. Sentiment takes place through an embedded Python pipeline that considers user sentiments and provides audio-visual feedback corresponding to emotions.

The OLED display screen, which communicates via I2C, will provide text string responses, emoticons, or system status messages to incentivize user interaction via multimodal feedback. The refresh rate of the screen is around 30 Hz which allows for the smooth refresh of continuously changing content. The speaker unit is hard wired to the Jetson Nano's audio output and the built-in amplifier makes sure audio quality is maintained, even in loud environments. The amplifier has a power rating of 3 W, which should have enough power for small-room spaces without distortion.

The power management within the AI System is centralized and controlled by the main power supply, which is the same supply as the Arms System utilizes. The AI System will be powered by a regulated 5 Volt, 3 Amp supply to the Jetson Nano, and is able to be maintained as stable during peak computation loads during speech processing/inference, both of which are considered maximum processing states. The power consumption of the AI System ranges from 1.5 Amps whilst idling to 2.7 Amps during peak processing states. Thermal management of the CPU is conducted using a small heatsink and fan assembly and is able to keep CPU temperatures below 65°C even during extended periods of usage.

Using a customized ROS node architecture for software integration provides the AI module with the ability to send and receive messages via serial communication with the Arms System that will allow the arms to perform synchronized gestures based on emotion or spoken phrases that can enhance the user experience even further. Furthermore, the AI module also

communicates with the autonomous navigation system to manage the interactions between conversation and movement and ensures that the robot can still engage in meaningful interactions while in motion.

Overall, the AI System combines advanced electronics and computing intelligence to embody IRIS' human-ike interaction and companionship extended through authentic conversation and emotional cognizance.

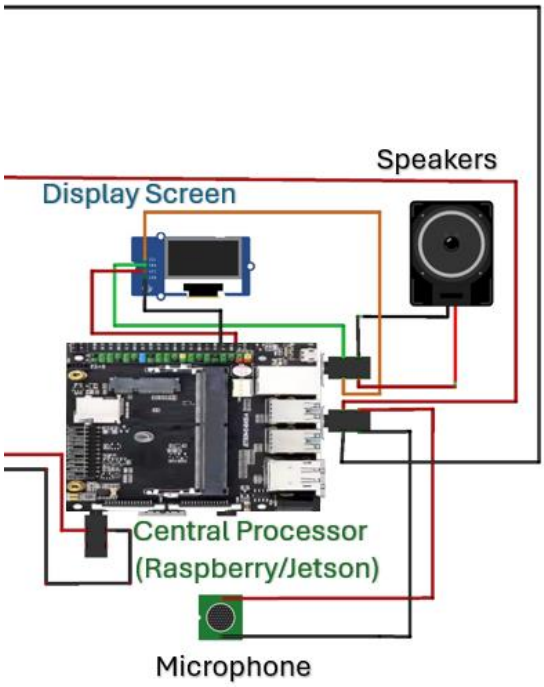


Figure 90: Detailed circuit diagram for the AI System, showing the incorporated central processor (Raspberry Pi, Jetson Nano), display screen, speaker, and microphone.

4.2.4. IoT Monitoring Circuit

The IoT Monitoring System is an essential feature of IRIS that concentrates on environmental awareness, and collecting real-time information; the IoT subsystem is responsible for increasing robot awareness by continually monitoring environmental parameters for safe operation in context. The IoT subsystem is centered around the issue of ESP8266 NodeMCU microcontroller.

The NodeMCU was chosen for the IoT subsystem as it is lightweight, has on-board WiFi and is easy to program; it is intended to be the coordinating processor for the IoT subsystem, and has the ability to not only read data from many sensors, but to communicate to the main processor by WiFi. The IoT Monitoring System connects three main sensors (Figure 92):

a. **MQ-2 Gas Sensor**: The MQ-2 Gas Sensor is a suitable sensor to detect smoke and some combustible gases. The MQ-2 Gas Sensor is interfaced to the ESP8266 using an analog input pin. To attenuate the effect of voltage fluctuations and buffer the output voltage down to a reasonable value for the analogue to digital converter within the ESP8266, a voltage divider configuration (on of the resistors has a resistance of 1.075 kΩ and one of the resistors has a resistance of 2.15 kΩ) was used.

b. **DHT11 Temperature and Humidity Sensor**: This digital sensor will be used to monitor room temperature and relative humidity. As a digital sensor, the DHT11 is powered from the VIN pin (5 V) of the NodeMCU, producing reliable readings. The DATA line connects to GPIO4 (D2), as defined in the flexible robotic arm control code, to obtain readings of ambient conditions. The DATA pin is connected to VCC through a 10 kΩ pull-up resistor in order to eliminate unwanted signals from environmental random signals.

c. **Flame Sensor**: The flame sensor detects flame/fire, or high temperatures that indicate a flame. It is set to send a digital output signal to D3 (GPIO0) on the ESP82666, ensuring that, if fire alerts or safety measures are necessary, the system is ready to act rapidly.

a) b) c)

Figure 91: The main sensors part of the IoT Monitoring System (a) MQ-2 Gas Sensor; (b) Flame Sensor; (c) DHT11 Temperature and Humidity Sensor.

A voltage regulator module is used to convert the voltage from the primary 9 V battery source to the voltages required for operation of the ESP8266 and any sensors that are connected to the ESP8266. By using a module for regulation, we can ensure all components are within safe voltage specifications, i.e., protected from voltage damage as well offering more reliable performance with the sensors.

All of the grounds within this sub-system are tied together to a common reference point, which is important for noise immunity and signal integrity between the different sensors and the main microcontroller.

The IoT Monitoring System shown in Figure 93 is executed at a power consumption of less than 500 mA fully loaded with sensors, where the ESP8266 (+mcu power is approximately 70 mA during active measurement from the various sensors This allows for extended operation of the battery without substantial drain during continuous operation necessary for continual monitoring while in deployments.

For software, the ESP8266 uses the Arduino IDE to be programmed and used its onboard WiFi stack to connect and send data to the central server. The data collected can be used for a number of things including real time, reactive monitoring, alerts, and high level adjustments by the master controller.

In conclusion, the IoT Monitoring System provides an essential platform for the IRIS architecture by integrating environmental sensors with a powerful microcontroller platform to allow the robot to sense and communicate with their environment naturally and autonomously.

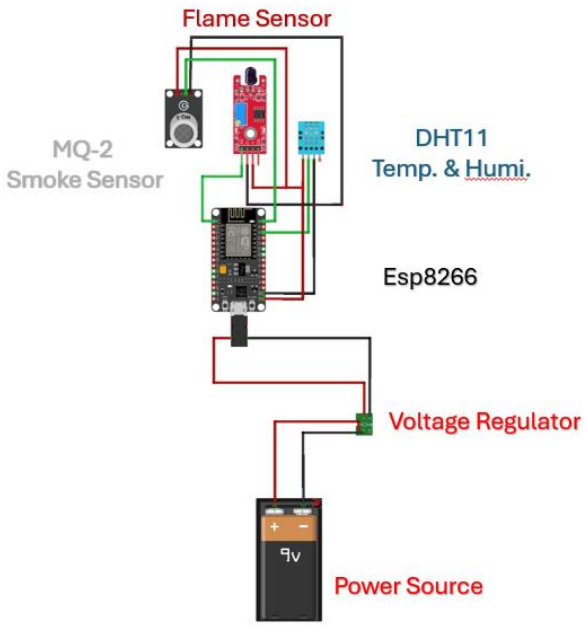


Figure 92: Detailed circuit diagram for the IoT Monitoring System, showing the incorporated ESP8266 NodeMCU, DHT11 temperature and humidity sensor, MQ-2 smoke, flame sensor, 9V battery or voltage regulator.

4.2.5. Autonomous Navigation System

The Autonomous Navigation System serves as the foundation of the IRIS platform's ability to navigate complex indoor spaces. It utilizes an integrated system of sensors, motor commands, and real-time data processing to allow the robot to respond to a changing environment, and operate safely. The component wiring diagram in Figure 94 denotes how the devices are connected, while also outlining the modular and interconnected structure of the system.

At the centre of the Autonomous Navigation System is the Raspberry Pi computing board, which carries out the higher-level navigation algorithms, enables communication with the Robot Operating System (ROS) component, and collects data and commands from various sensors. It has a direct connection to the Teensy microcontroller, which performs many of the low-level real-time commands that enable motor control, such as reading and processing encoder signals and communicating with the motion driver.

The system runs two JGY-370-1285 worm gear motors that have high torque; each motor runs through its own L298N H-Bridge module. Because there are two H-Bridges, the each type of motor can run independently both forwards and backwards. This design supports the differential drive that allows for better mobility. All the motors get their power from a 12 V battery pack, which is regulated to ensure a stable output voltage and current under sustained use.

An MPU6050 Inertial Measurement Unit (IMU) is connected to the Teensy to improve the robot's orientation estimation and orientation tracking. The IMU provides six axes of measurement (three axes of accelerometer data, and three axes of gyroscope data) which allows for accurate real-time state estimation. Additionally, each motor includes encoders that increase the robot's ability to keep position and speed control.

An added layer of safety, and for user interaction, was added by including a Safety Button between the battery power supply and the main circuit. The Safety Button will kill the power to the entire system in the case of an emergency, improving operational safety in uncertain or movement-prone environments.

The robot includes a 2D LiDAR which interfaces with a Raspberry Pi for environmental interaction. The 2D LiDAR will do real-time scans of the robot's surroundings in order to detect obstacles and to use mapping and localization through a ROS framework. The high-resolution

point cloud data from the LiDAR is vital for generating dynamic maps and providing safe and economically intelligent paths for navigation.

The Autonomous Navigation System is powered by its own battery module. This module provides power to all subsystems including Raspberry Pi, Teensy, IMU, LiDAR, and motor drivers. By separating power sources, we minimize switching power supply possibility of voltage fluctuations and assure that each component operates with the specific voltage ranges specified for each subsystem. Segregation of power to all subsystems provides better reliability and stability of the total system.

All system connections are completed with care and each subsystem wiring is color coded for ease of maintenance and troubleshooting. The ground (0V) connections across the system are also managed to ensure all interconnected subsystems have a common voltage reference to minimize electrical noise and help to assure signal integrity in between subsystems.

This subsystem successfully combines mechanical actuation (the motors), sensor information (real-time feedback from IMU and LiDAR), and processing intelligence (from Raspberry Pi/Teensy) into a coherent autonomous capability. The modular design provided monitored state data and facilitates easy upgrades and scaling, as well as integration into the IRIS platform's robotic subsystems.

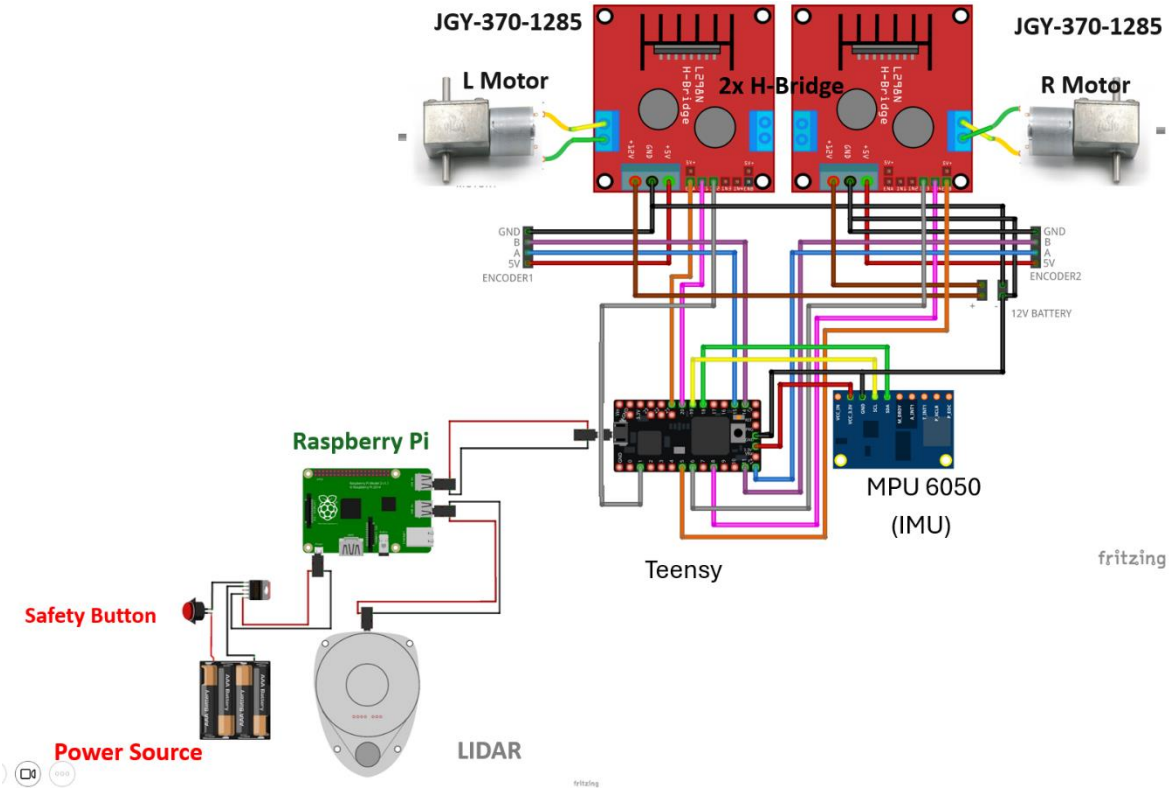


Figure 93: Detailed circuit diagram of the Autonomous Navigation System showing integration of the Raspberry Pi, Teensy, JGY-370-1285 motors, dual H-Bridge motor drivers, MPU6050 IMU, LiDAR sensor, safety button and power supply module.

4.3. Conclusion

The systems design, electrical, and electronics design of the IRIS platform were thoroughly documented in this chapter, illustrating its modular approach and an advanced amalgamation of critical systems enabling the platform to deliver its unique functionality. Rather each system - Arms, AI, IoT and Autonomous Navigation - were conceived, assembled, and powered as needed to support their assigned functions as a fully-fledged socially assistive robotic companion.

The Arm System utilized the Arduino Mega 2560 and the PCA9685 PWM driver to communicate the dynamic movement of the left and right arm servos, to create effective gesturing that complements their interaction with the user. The AI System used a Raspberry Pi (or Jetson Nano) to manage the voice input selected through the microphone, utilize an AI framework to process the audio signal and provide multi-modal vocal and display output to their user with an expected interaction towards engaging in dialogue.

The IoT Monitoring System was a combination of environmental monitoring sensors based on the ESP8266 microcontroller, with the DHT11 temperature & humidity sensor, MQ-2 gas sensor, and flame sensor to allow for user safety while collecting real-time data on the environment surrounding the user. The System designed with voltage regulation and power management to utilize the provided 9Volt supply to efficiently operate and obtain compactness.

Lastly, the Autonomous Navigation System has developed an elegant mix of hardware and software, using a Raspberry Pi, Teensy microcontroller, two dual L298N H-Bridge motor drivers, an IMU, and a 2D LiDAR to deliver accurate real-time navigation in dynamic indoors environments. An added safety button and regulated power source means the project can be put into reliable use, safely, even in testing conditions.

All of these interrelated systems are principled and cohesive designs that combines safety features, mechanical actuation, sensor inclusion, computational intelligence, and user safety. Furthermore, the modulated electrical architecture of IRIS not only promotes ease of maintenance and debugging; it also facilitates future scalability and technology upgrades to keep the system at the forefront of socially assistive robotics.

Chapter 5 : Control System

The chapter presents control systems concepts associated with the IRIS companion robot, modelling, simulation and performance evaluation. It details how the robots differential drive can assimilate high level commands, execute a specific purposeful movement whilst planning and coordinating its own motion stability.

5.1. Kinematic Modelling of Differential Drive Robots

This section presents the kinematic modeling of differential drive mobile robotic systems as implemented in IRIS. A differential drive mobile robot uses two independent wheels mounted along a common axis for motion. The differential drive mobile robot can move both linearly and rotationally by controlling the relative angular velocities of each wheel. This design allowed IRIS to satisfy the mechanical simplicity indicated above, while also allowing IRIS to exhibit a high degree of maneuverability, a necessary design requirement for navigating dynamic indoor environments.

The fundamental two kinematic equations that will determine the relationship between angular velocities of the wheels and the linear and angular velocities of the robot are given by the equations:

$$v = \frac{R}{2}(\omega_R + \omega_L) \qquad (1)$$

$$\omega = \frac{R}{L}(\omega_R - \omega_L) \qquad (2)$$

Where:

- v is the linear velocity of the center of mass of the robot,
- ω is the angular velocity about the vertical axis of rotation that passes through the center of mass,
- R is the wheel radius, which is an important parameter because it affects the wheels, while in motion, conversion of rotational motion into linear displacement,
- L is the axle length, which will also be the lateral distance between the wheels and has an important role in the robot's capability to make turns,
- ω_R and ω_L are the angular velocities of the right and left wheels, respectively.

These mathematical relationships are critical for converting top-level trajectory planning commands to individual wheel actuator signals. These equations form the basis of motion control algorithms that allow proper path following and maneuverability. Also, the formal definition and use of these relationships are critical in the development of simulation environments that replicate the robot's motions in a wide range of operational settings, allowing rigorous evaluation before deployment.

Table 19 Robot Specifications Relevant to Kinematics

Parameter	Symbol	Value	Notes
Wheel Radius	R	0.0425 m	Derived from 85 mm diameter wheel / 2
Wheel Diameter	D	0.085 m	As per product specification
Ticks per Rotation	N_{ticks}	1044	12 CPR encoder × 87 gear ratio
Axle Length	L	0.2246 m	Empirically measured wheel separation
Max Angular Velocity	Ω_{max}	10.16 rad/s	97 RPM converted: (97 × 2π) / 60

The wheel radius (R) is the key variable when establishing the connection between wheel rotations and linear speed, which relate directly to trajectory, and the control of movement. Axle length (L) relates to turning radius and ease of movement of the robot. These represent important factors in the design of motion planners and controllers. The encoder resolution expressed in ticks per rotation (N_{ticks}) drives the odometric accuracy and indirectly influences the robot's capacity to localize. Finally, the maximum angular velocity (ω_{max}) constrains the robot's operating envelope and informs the controller design by guiding components to avoid actuator saturation and destabilizing operability.

In practice, these parameters (Table 19) need to be accurately configured, continually verified and adjusted as needed to lessen the impact of mechanical wear, encoder drift, and interference by the environment in order to realize the overall efficiency and effectiveness of the robot. The effective collaboration of these robot-specific parameters and the preceding kinematic model form the basis for a reliable, high-accuracy control system capable of operation and robustness in complexity and dynamic environments.

Figure 95 shows the MATLAB Simulink model of an open-loop control system of a two-wheeled differential drive robot system that shows how kinematic modeling concepts can be related to a control-oriented simulation based environment. The figure is indicative of the process of modularization that is taken when developing the encoding processing subsystems, dearest reckoning logic, and motor subsystems. The application of the kinematic equations in the model and the relationship of them into the motor subsystems allows for an opportunity to find the value of the theoretical analysis using simulation, and to experiment to find tuning parameters, before proceeding to develop closed-loop control.

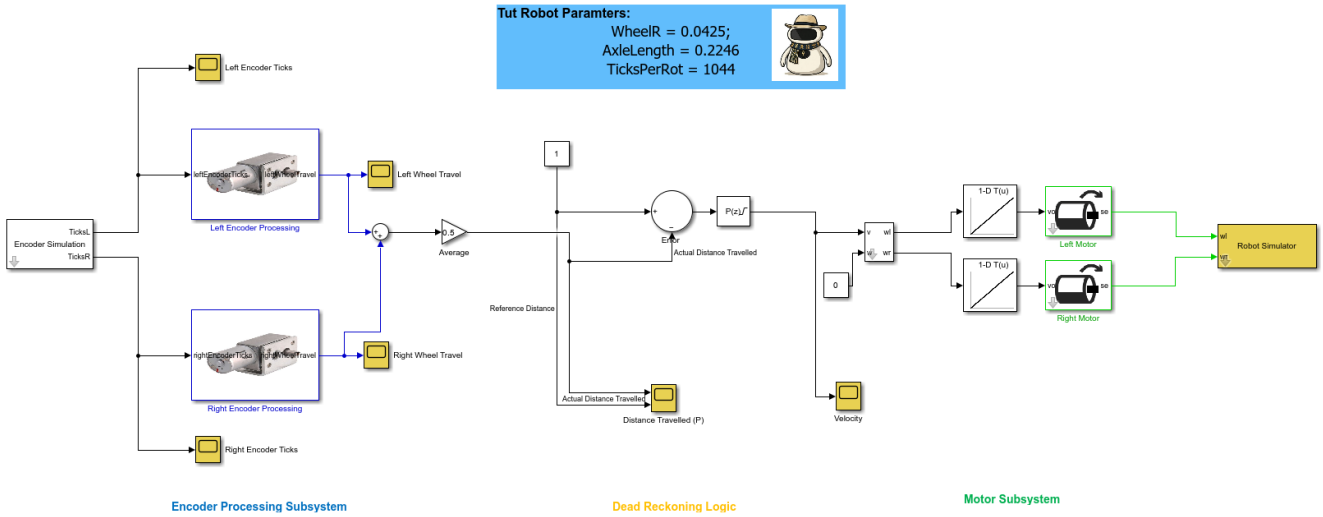


Figure 94: A Simulink open-loop Matlab model for a two-wheeled differential drive robot featuring encoder processing, dead reckoning and motor control subsystems components combined through an entire modular simulation environment.

5.2. Dynamic Modeling and Parameterization

This section represents an in-depth analysis of the dynamic modeling framework necessary for differential drive robotic platforms with both theory and parameterization for accurate control design. The dynamic model of a differential drive robot involves more than just kinematic relationships and involves the inertial, friction and behavior of electromechanical interactions that affect transient and eventual steady-state behavior of the robot.

The fundamental dynamic model that describes the torque required to turn each wheel is written as

$$\tau = J \cdot \alpha + B \cdot \omega \qquad (3)$$

where:

- τ is the torque due to the motor acting on the wheel,
- J is the total moment of inertia of the wheel-motor system,
- α is the angular acceleration of the wheel,
- B is the viscous friction
- ω is the instantaneous angular velocity of the wheel.

The model demonstrates rotational inertia to resist the rate of change of angular velocity and dissipative forces from friction. These are extremely important factors for assessing the dynamic response of the system. This relationship is also useful for implementing more advanced controls such as feedforward compensation and model predictive control.

Table 20: Robot Specification Relevant to Dynamic Modeling

Parameter	Symbol	Value	Notes
Max Angular Velocity	ω_{max}	10.16 rad/s	97 RPM converted: (97 × 2π) / 60
Nominal Voltage	v_{nom}	12 V	JGY-370-1285 motor specification
Stall Torque	τ_{max}	13 kg·cm	JGY-370-1285 motor specification
Rated Load Torque	τ_{rated}	3.5 kg·cm	JGY-370-1285 motor specification
Gear Ratio	GR	87:1	JGY-370-1285 motor specification
Free-run Speed	ω_{free}	97 RPM	At output shaft, from motor specification

It is essential to include each of these parameters in a dynamic characterization of the robot as shown in Table 20. The maximum angular speed (ωmax) describes the limits of the actuators, and defines a major design limitation during the controller design process. The nominal voltage (vnom) describe the electrical potential required to operate the motor and, in addition, defines the torque and speed the actuator can generate depending on the loads applied.

Stall torque (τmax) and rated load torque (τrated) specifically define the allowable loads on the robot and guarantee the actuators are driven within thermal and mechanical limits. The gear ratio (GR) changes the output speed and torque of the motor and affects both the responsiveness of the system to accelerations, and performance in steady state. Finally, the free-run speed

(ωfree) provides insight on the operational characteristics of the drive system at no-load for comparison to the other zero-load performance metrics specified.

In practice, applying these dynamic parameters to simulation frameworks and control designs requires a structured approach which will evaluate the capability of the controllers deployed to deal with disturbances, disturbances in the load, and reality of non-ideal operation. Accurate modeling, and parameter identification, normally through empirical testing and system identification approaches, would provide a more accurate dynamic model and thus increase controller performance and robustness.

5.3. Encoder-Based Odometry Modeling and Implementation

To accurately control mobile robotic platforms, usable odometry, or the ongoing computation of a robot's location relative to its origin, is vital to autonomous navigation approaches and feedback control mechanisms. One way to accomplish this is by fitting rotary encoders to the motor shaft of each wheel, in which the motor encoder count or "tick" is specified for every full rotation in a set amount of "ticks". Generally, the specifications for encoders are found in the manufacturer's datasheet and are consistent for each encoder, making them a calibration constant for any robotic platform.

The total linear distance, or D, traveled by the wheel can be defined mathematically as equation 4:

$$D = \frac{\text{Encoder Ticks}}{N_{\text{ticks}}} \times 2\pi R \qquad (4)$$

where:

- D is the total distance traveled by the wheel (m),
- **Encoder Ticks** represents the accumulated tick count that is recorded by the encoder,
- N_{ticks} the fixed encoder ticks-per-rotation combination,
- R is radius of the wheel (m).

The convenience with which this relationship succinctly defines the transition from discrete encoder measurements to continuous displacement also serves as a basic framework for odometric calculations. The accuracy of this relationship clearly depends on the correct measurement of wheel radius and the correctness of encoder resolution, which must all be determined experimentally and monitored rigorously in order to reduce error induced by wear, deformation or other system artefacts.

To apply this conceptual framework in a control-oriented simulation environment, the encoder tick-to-distance transformation can be executed in Simulink. Figure XX provides a Simulink subsystem implementing this transformation, wherein inputs consisting of the encoder ticks are first multiplied by the circumference of the wheel, $2\pi R$, then divided by N_{ticks}, to produce a continuous estimate of the distance travelled by the wheel. The design is modular, with key blocks and consistent naming—this will allow user transparency, provide ease of debugging,

and provide a methodical testing method to provide consistency between simulation-based outputs and physical performance by the robot.

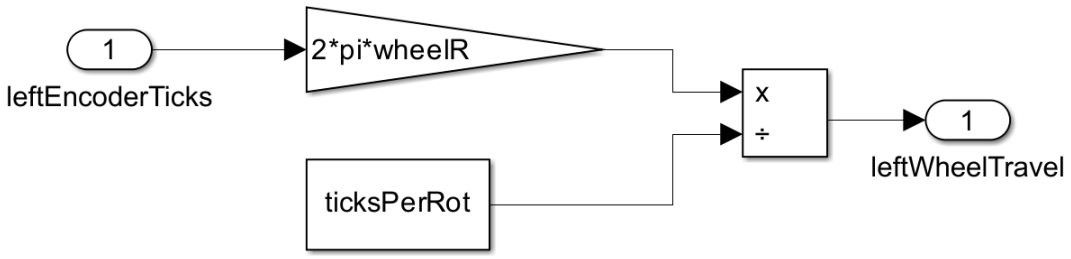


Figure 95: Simulink implementation of the encoder tick-to-distance transformation subsystem, illustrating the multiplication of encoder tick input by wheel circumference and subsequent division by encoder ticks per rotation.

5.3.1. Practical Implementation of Odometry Using Simulink

To make the theoretical odometry model useable in a control-oriented simulation environment means that the entire process of encoder tick-to-distance transformation, and everything else that happens after it, can be configured in Simulink utilizing its modular, graphical programming style. For this implementation, the generation of encoder ticks under constant motion is handled nicely using the Encoder Simulation block that simulates tick rates (or counts) for both the left and right wheels. These tick signals enter left encoder processing and right encoder processing subsystems which calculate the distance traveled for each wheel using gain blocks set to the multiplicative factor $\frac{2\pi R}{N_{ticks}}$. The encoder tick inputs are constantly converted to useful values for displacement.

Then, the Average block averages the travel distance of the two wheels, or average estimated center-of-mass travel distance. The Reference Distance block compares the previous two variables to a target distance (goal) so that the error can be calculated for closed-loop control. The distances, velocities, and other outputs are labeled and organized in well-defined subsystems to encourage model clarity and ease of maintenance. Furthermore, explicitly specifying the values of the parameters (e.g., wheel radius R and encoding resolution N_ticks) aids in ensuring we measure performance consistent with the robot's physical characteristics. Time in the simulation was chosen to ensure multiple rotations of the wheel were tested, to ensure odometry performance can be evaluated over extended operation. This process allows us to verify that the model is plausible and can facilitate a model with higher-level control and

s. The complete Simulink model is provided in Figure 2.3.2, showing the complete Encoder Processing Subsystem, Dead Reckoning Logic and Motor Subsystem used in a full odometry simulation.

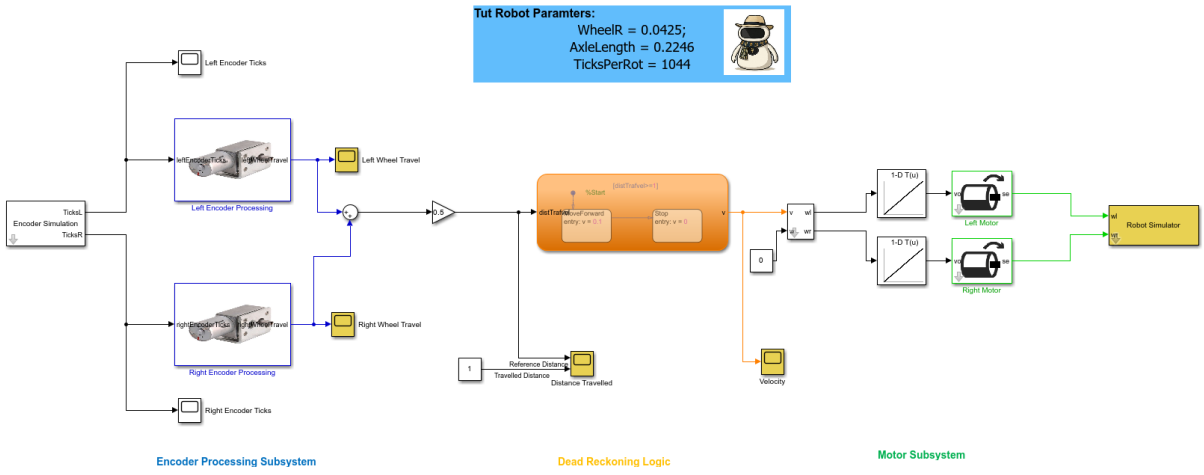


Figure 96: Simulink model of the encoder-based odometry implementation, featuring Encoder Processing, Dead Reckoning Logic, and Motor Subsystems for differential drive robots.

5.3.2. Extending Odometry to Differential Drive Configurations

Most mobile robotic platforms have a differential drive configuration and an odometric model needs to include both the left and right wheel encoders in order to capture the differences in the respective paths lengths that each encoder direction experiences during curved or turning, because each wheel has a different path.

In the Simulink environment, the left wheel odometry model is simply copied to have a right wheel module, each of which estimate distance independently based on encoder ticks. The robots net center-of-mass motion can be computed by averaging the distance traveled by the left and right wheels as in equation 5:

$$D_{avg} = \frac{D_{left} + D_{right}}{2} \qquad (5)$$

where:

- D_{avg} represents the estimated linear displacement of the robot's center of mass (m),
- D_{left} and D_{right} denote the distances computed by the left and right wheel encoders, respectively.

The procedure is modeled in Simulink with an Add block that combines the distances, followed by a Gain block with a factor of 0.5 implementing the average. Therefore, the robot's odometry can be considered reliable regardless of the speed of each wheel or uneven terrain.

A useful test case is simulating a rotation in place, where the same angular speed is applied to one wheel moving forward, and applied to the other wheel, moving it backward. In the case of this movement, the robot's center of mass has net zero linear displacement, but the robot has a pure rotational movement. Having this kind of behavior recorded in the odometry model will help develop realistic control systems to operate successfully in complicated, dynamic environments.

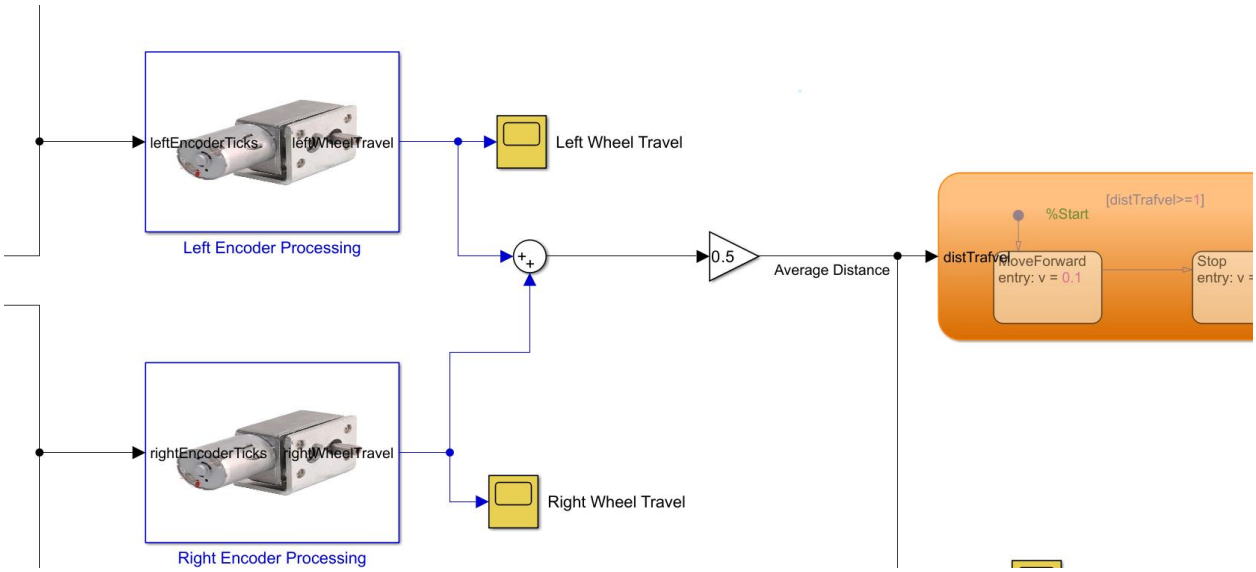


Figure 97: Simulink block diagram of the extended encoder-based odometry model of a differential drive robot, showing left- and right-encoder calculations, averaging logic, and its use with state machine-based movement logic.

By systematically combining prick interpretation, distance calculations, and dual-wheel averaging in Simulink, this chapter establishes a solid procedure for producing reliable odometry in differential drive robots. The encoder-based model allows for accurate position estimates necessary for more complex motion control architectures which require real-time position feedback. Furthermore, an encoder-based turning model is fundamental for developing robotic control systems having high-accuracy and reliability in both simulated and physical environments.

5.4. Stateflow Logic and Flowcharts

This section will highlight the use of Stateflow in the Simulink environment that can assist with higher-order logic decision making in differential drive robotics systems. Stateflow allows for the development of finite state machines (FSM) using its graphical user interface, which will

allow for the creation of modular, structural control logic to coordinate multiple and various behaviors of the robot.

In the context of differential drive robots, one simple yet effective control strategy is to command the robot to move until it reached a desired travel distance, at which time it enters a stop state. The action of moving until a distance is completed maintains high accuracy, in addition to reliably stopping the robot at a target displacement.

The Stateflow model consists of two distinct operational states: MoveForward and Stop. Each state has its own control outputs, with the MoveForward state driving the robot forward with a non-zero velocity (v = 0.1) and the Stop state stopping the robot's velocity at zero. The two state transition is made based on real-time feedback readings from the distance traveled from the encoder where we considered a transition from MoveForward to Stop when the distance traveled is equal to or greater than one meter (distTrav ≥ 1).

Figure 99 displays the overall Stateflow implementation. Panel (a) shows how the operational logic of the robot is illustrated graphically as a high-level flowchart that conveys the logical sequence from Start to MoveForward and then on to Stop. Panel (b) illustrates the Stateflow chart in Simulink that shows the building blocks of states, transitions, and velocity designations were designed as graphical representations.

The incorporation of Stateflow into Simulink allows for the separation of the control logic and continuous dynamics of the system improving readability and maintenance of the model. Additionally, it also couples real-time sensor data to the control logic decisions, providing assurance that the operational motion of the robot aligns with the intended operational objectives, which only contributes to the strength and dependability of the control framework.

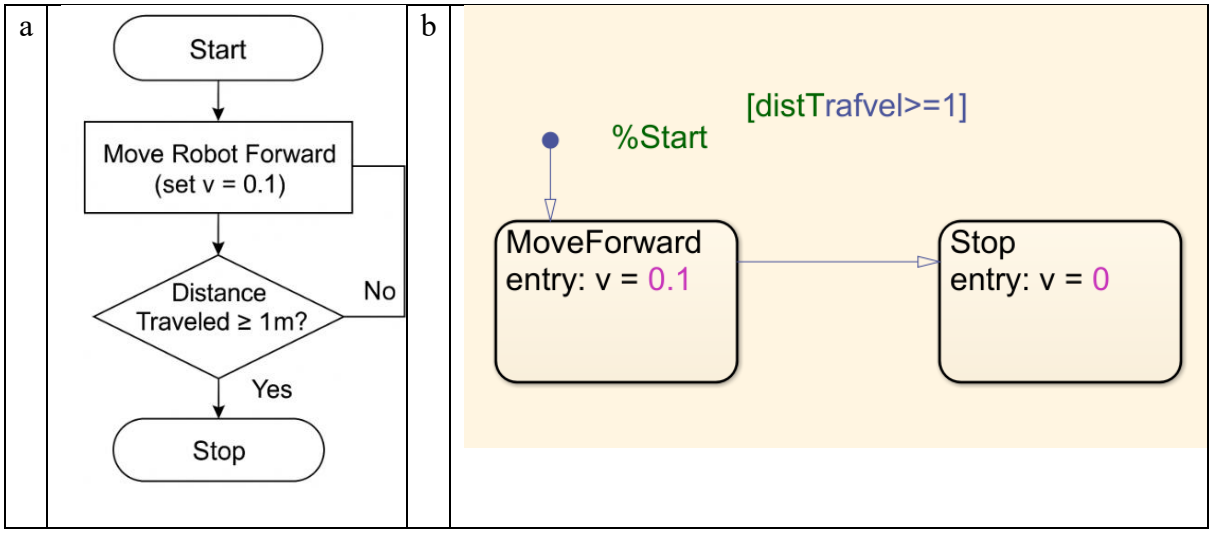


Figure 98: Flowchart representation of the finite state machine controlling robot motion, showing transitions based on traveled distance; (b) Simulink Stateflow chart implementing the same logic, illustrating states, transitions, and assigned control outputs.

5.5. Control Architecture and Performance Analysis

This section provides a full assessment of the distance control systems used for a differential drive mobile robot in a performance evaluation within a simulated environment. In the evaluation, the first distance control system presented is "open-loop" and serves as the baseline configuration. Following the "open-loop" configuration, the baseline proportional (P), proportional-integral (PI), and proportional-integral-derivative controllers (PID), characterize the baseline distance control systems in progressively increasing performance, confidence, accuracy, and robustness.

The simulation is done in a specified infrastructure constructed in Simulink with three major subsystems: the Encoder Processing Subsystem, Dead Reckoning Logic, and Motor Subsystem. Within the Encoder Processing Subsystem there are simulated encoder ticks collected then processed to figure out how far each wheel travels, which drivers average these values to determine the displacement of the robot's center of mass (coam). In the Dead Reckoning Logic the travelled distance is determined, and the reference distance is utilized to compute an error signal which is used for feedback based control architectures. The Motor Subsystem receives a velocity command and applies that to the robot model.

Each control condition is investigated with distance travelled plots which compare the distance travelled to the distance with respect to the reference distance. The study is focused on system characteristics such as rise time, overshoot, settling time, and steady-state error. These performances measures are essential in assessing the robot's ability to accurately and consistently reach the proposed target distance considering the proposed control methods.

The simulation results highlighted the overall positive accumulative effects of feedback control, effectively demonstrating how the feedback control action, whether proportional, integral, or derivative, can produce improved dynamic response and steady-state accuracy. The results emphasized the iterative aspect of the design process of controller architecture by confirming that would-be chosen controller architecture achieves the required performance objectives in real-world applications that will have unmeasured mechanical uncertainties and disturbances.

5.5.1. Open-Loop Control Performance Analysis

In order to have a baseline against which to test future feedback-based control strategies, we created and analyzed an open-loop control strategy in Simulink. The system was designed generally into three main subsystems - Encoder Processing Subsystem, Dead Reckoning Logic, and Motor Subsystem. In this way, the differential drive robot is being operated with a velocity command for both wheels, without any modifications or feedback corrections, at the same time. This mode of operation will enable an initial assessment of the dynamics of the response in an open loop system.

The Encoder Processing Subsystem processes simulated encoder tick signals for the left and right wheels to compute the distance travelled by each wheel. The Calculation applied a conversion factor with a wheel radius of 0.0425 meters and resolution of 1044 ticks/rotation as defined in Parameters.

The Dead Reckoning Logic compares how far it has traveled to a reference distance of 1m to evaluate how well the open-loop controller performed. In this situation, it should move 1 meter exactly, in a straight line. However it is an open-loop system. Therefore it has no feedback capabilities. This means that it cannot alter the velocity input, or correct its input value to the target distance as it approaches it.

The simulation results show key performance characteristics of the open-loop system, presented in the distance travelled plots (Figure100). The Actual Travelled Distance is represented by the yellow curve, while the blue curve illustrates the constant Reference Distance of 1 meter. The reference is exhibited in the rise in distance travelled over time being approximately linear for the initial states, since the robot reached the goal (1 meter travelled distance) at approximately 8.4 seconds. Although the robot reached the goal, because deceleration or stopping control was void in this model, the robot cannot stop and continues to move forward beyond the target goal distance, reaching approximately 1.013 meters distance travelled, which was a distance travelled of approximately an overshoot of 1.3% from the Reference.

This overshoot results from the limitations of open-loop control: there is no feedback loop available to sense and compensate for the error, so the system cannot automatically adjust its velocity and/or braking force. In addition, there are no means to compensate for unmodeled dynamics like mechanical friction, slip, or disturbance due to the environment. The recognition of the nature of the simulation is especially evident in the steady-state response of the

simulation. The Actual Travelled Distance plot levels off substantially greater than the Reference Distance, suggesting that the system cannot eliminate steady-state error.

Further, the simulation plots in (Figure 100) do not present oscillations or temporal transients, reinforcing the feedforward characteristic of the system. In each trial, noting the constant slope of the distance trajectory prior to hitting the target confirms that the open-loop system can provide steady motion, but it does not have the capability to adaptively correct trajectory in real-time.

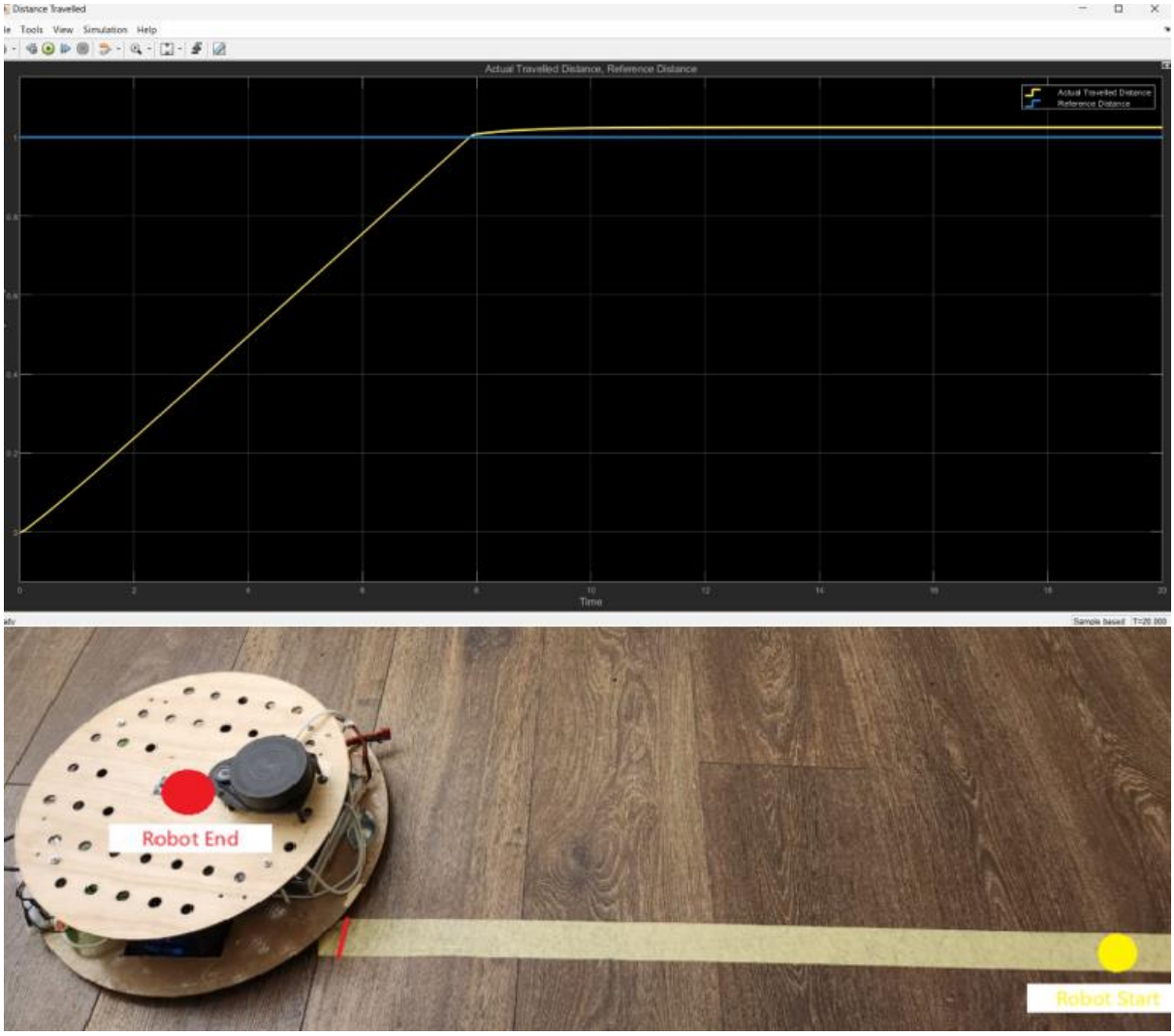


Figure 99: An open-loop control in which the distance signal increases linearly, with overshoot after it hits the target of 1 meter, demonstrated in both MATLAB and Physical Prototype of the Base. Clearly, there is not any dynamic error correction to the distance signal.

In short, the analysis of the open-loop control performance verifies that the system, while able to move a robot to a desired distance, cannot control placement precisely. The open-loop controller traveled a distance about 1.3% greater than the reference distance and was unable to

stop movement at the reference distance. This highlights the benefit of using feedback control in the form of P, PI, or PID controller to make the control action more accurate, robust and reproducible in the real operating environment.

5.5.2. P Controller Performance Analysis

To resolve the limitations seen in the open-loop case, a proportional (P) controller added to the Simulink model, represented a basic configuration of feedback control. The system displayed all the same components: Encoder Processing Subsystem, Dead Reckoning Logic, and Motor Subsystem. Adding a P controller was an effort to allow the robot to dynamically modify its speed before reaching the target distance, thus reducing the steady-state error as well as better tracking the trajectory.

Using the Simulink PID Controller block meant that the P controller was developed in discrete-time with a proportional gain value of 1.1 (hand-drawn in Figure 101). This gain value was obtained from several tuning iterations to achieve trade-off between being aggressive to climb a distance quickly, while being stable to avoid uncontrollable overshoot or prolonged oscillation, adversely affecting overall control effort.

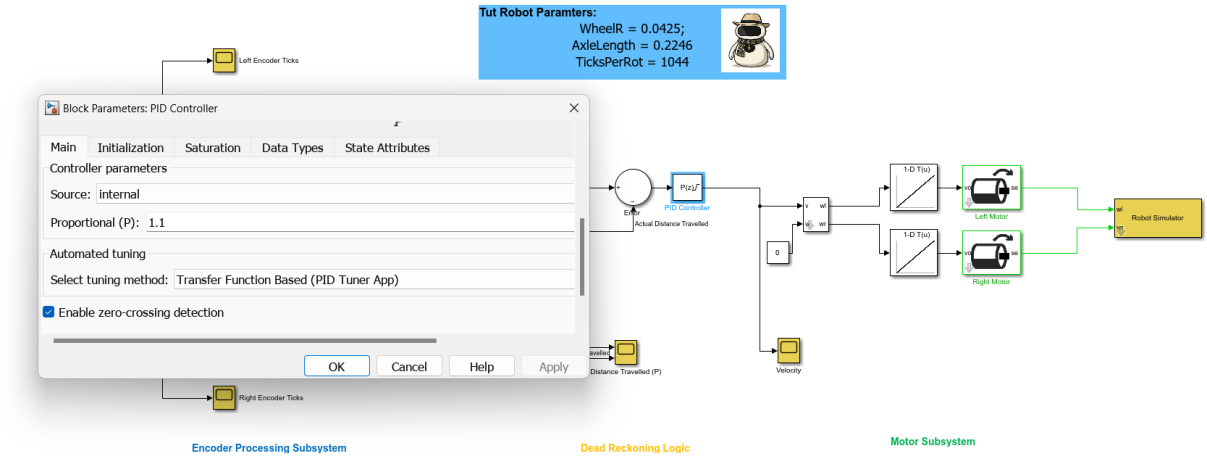


Figure 100: A P Controller in Simulink; involves Proportional Gain P=1.1 and discrete-time mode, which is critical for maintaining the robot's speed as it approaches its reference distance.

In this scenario, the error signal (i.e. the difference between the reference distance of 1 meter and the distance travelled) was the input to the P controller. The controller output, which is a velocity command, then acted on both wheels via the Motor Subsystem to drive the robot that way towards the reference target.

The simulation results, illustrated in the plots of distance travelled (Figure 102), convey some notable insights into P controller performance. The yellow curve, the Actual Travelled Distance, follows a pattern of characteristic exponential rise toward the Reference Distance (blue curve), and relayed how quickly the system converged and became stable. The robot reached 1.0 meters quickly, at right about 4.8 seconds. It is noted that there was a small overshoot (approx. 1%) beyond the target distance, as the Actual Travelled Distance exceeded 1.0 meters by approximately 0.01 meters at that time (not insignificant, of course). That particular overshoot is a common period associated with P-only control and is only informative in nature, as no correction mechanism exists to eliminate or even fully dampen the transient response further at that point, or steady state error for that matter.

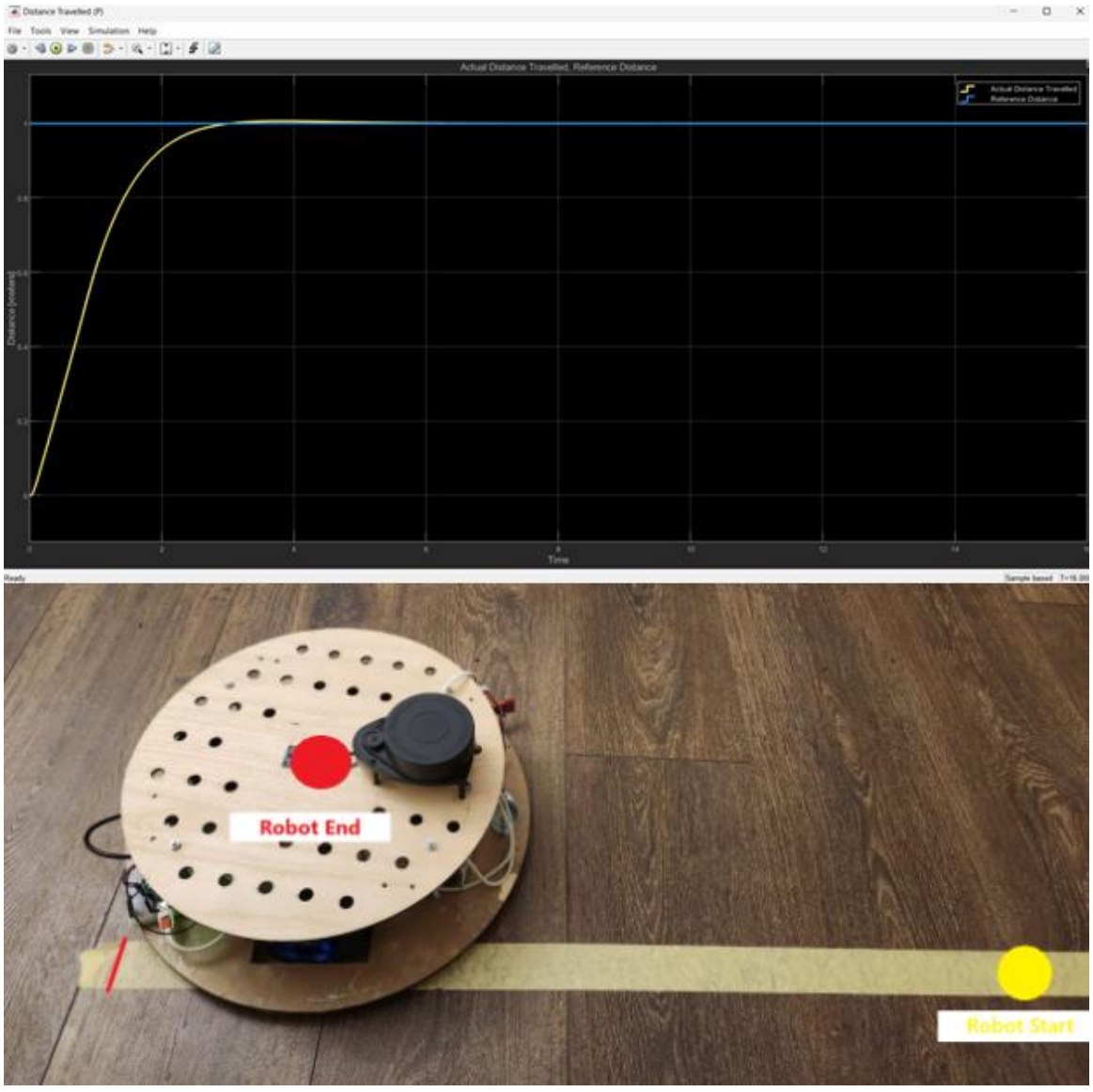


Figure 101: P Controller figures from simulation, depicting the actual travelled distance (yellow) approaching the reference distance (blue) with a small amount of overshoot and steady-state error as in MATLAB and Prototype Base; which is normal with proportional control only.

After the first overshoot, the system settled down slowly on the reference distance. However, the steady-state value was not zero -- that is, there was a small, persistent steady-state offset. This example illustrates the basic limitation of P control: it does a good job of minimizing error when in transient and the steady-state phase, but it can never get rid of offset entirely in steady state. The final steady-state value stabilized just below the reference line. In this case, when the system was commanded to go the reference velocity, it slowed down as it approached the reference distance, but it did not become zero; thus, there was steady state error.

To conclude, the inclusion of a P controller in this case study has yielded a far better performing system than the use of an open-loop system. The controller brings in much-needed dynamic error correcting action that allows the robot to better respond to errors from the target distance. However, the overshoot and steady-state error makes evident the limits of proportional-only control and that it cannot provide the tracking performance at the desired accuracy and stability. This summarizes why additional integral and derivative action will be required in the next controller designs, such as PI or PID controllers, to be able to further improve the behavior of the system and achieve the accuracy and robustness for a real robot in the real world.

5.5.3. PI Controller Performance Analysis

To improve upon the gains achieved with the P controller, a Proportional-Integral (PI) controller was added to the Simulink model to reduce the steady-state error and improve the accuracy of the system tracking the reference distance. In contrast to the proportional action, the PI controller improves the proportional action by adding an integral term which adds the error signal over time and ensures that any persistent steady-state offset is eliminated.

The PI controller was implemented with the help of the Simulink PID Controller block, set to discrete-time mode with proportional gain (P) of 1.1 and integral gain (I) of 0.2 (see Figure 103). Specific gain values were obtained using systematic iterative tuning using the Simulink PID Tuner App that provided tuning of a controller that negatively balances time to respond and steady-state performance. The proportional gain was chosen to get to the target distance quickly, while the integral gain was adjusted to be able to remove the steady-state error without getting too much oscillation.

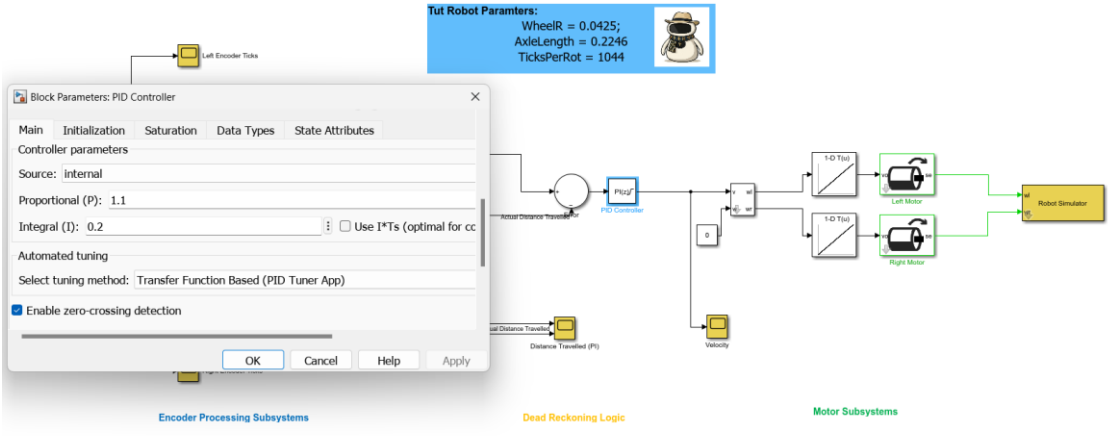


Figure 102: PI Controller block configuration; Proportional Gain P=1.1; and Integral Gain I=0.2, incorporated into the SImulink control topology for the differential drive robot.

The control architecture retained the Encoder Processing Subsystem, Dead Reckoning Logic, and Motor Subsystem as per previous controller implementations. The error signal, which is the error signal for the error, was defined to be the difference between the reference distance of 1 meter and the distance that the robot actually travelled. The PI controller continuously processed this error signal and produced the velocity command that was run on the robot's motors.

The simulation results, represented by the distance travelled plots (Figure XX), provide insight into the dynamic performance of the PI Controller system. The yellow curve representing the Actual Distance Travelled has a rapid rise toward the 1-meter Reference Distance (blue curve) and it is reached at approximately 5.2 seconds. But the system can show a significant overshoot due to the integral term and it overshot the reference target at approximately (0.050 meter) at about 1.04 meters, resulting in an overshoot of approximately 4.0%. That is an expected and natural outcome of an integral action, as it accumulates the error signal over time. The error could even become so significant that it would be considered integral windup.

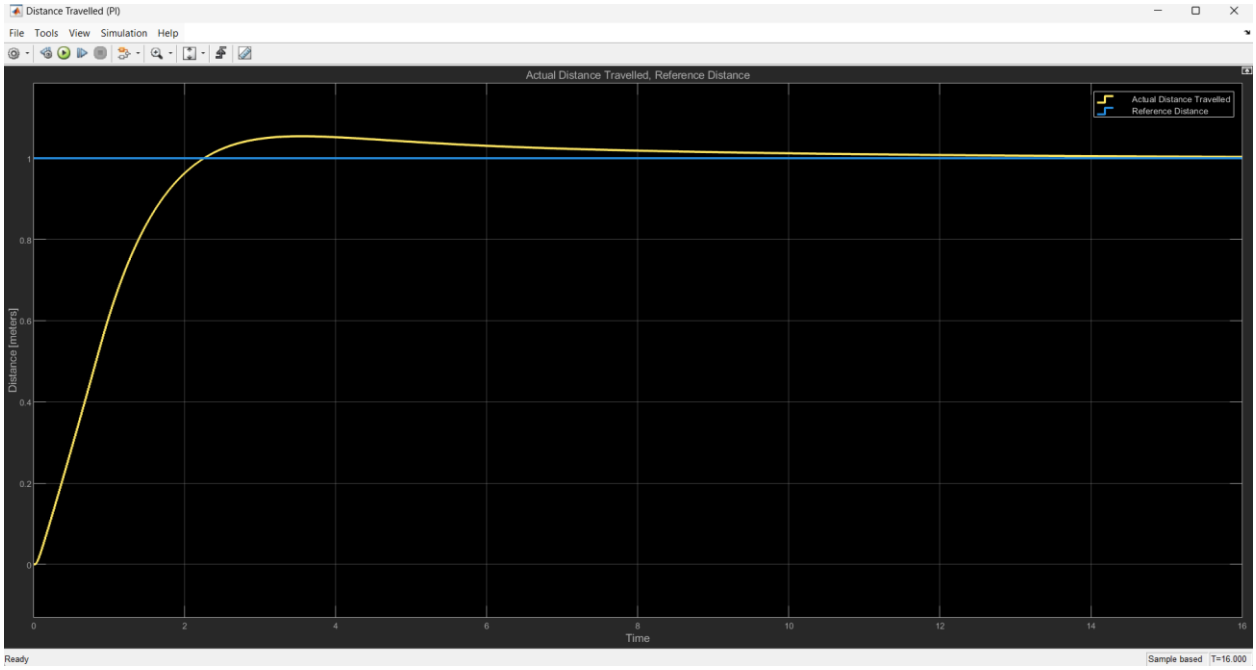


Figure 103: PID controller distance tracking, showing fast rise time, approximately 2% initial overshoot, and then dead on at the target of 1 meter and fast error correction.

After the first overshoot, the system is settling down and the distance trajectory is decreasing, approaching the reference distance. For the steady-state performance, we can see the Actual Distance Travelled, which is stabilizing a fraction of meter above the reference distance of approximately 0.01 meters. This clearly indicates that integral action has reduced the steady-state error relative to a P controller, but we still have a small offset. This may be due in part to the long decay time of the distance trajectory that we observe in the settling period. In particular, this demonstrates the inherent limitation of PI control in its ability to recover from the transient overshoot seen with the disturbance and its response.

Moreover, the trajectory analysis demonstrates that the integral term induces a phase lag in the system contributing to the transients observed. The settling time is longer than when a P controller is used; nevertheless, the P controller achieves more steady-state accuracy but leads to poorer transient performance. This illustrates the importance of derivative action (as in an actual PID controller) or anti-windup measures to reduce the overshoot and have a more damped response.

In summary, we see that using integral control provides a significant gain in steady state performance relative to proportional control because with integral control the steady state error is much less. As noted above, there is a small but negligible overshoot of about 4%, and a much longer settling time, however the constraints of having integral action based on phase lag and the Windup phenomenon can be problematic. The observations shown suggest a further

refinement using derivative action and/or anti-windup would help to maximize the performance gain for precise and consistent trajectory following in the robotic applications described.

5.5.4. PID Controller Performance Analysis

To improve the tracking performance and stability of the system even more, the Simulink model included a Proportional-Integral-Derivative (PID) controller. A PID controller is an extension of the proportional and integral controls used with the integration of a derivative term which operates on the error signal's rate of change. This is useful for the system to anticipate error trends and counteract sudden shifts in trajectory, increasing the transient response and decreasing overshoot.

The PID control-law, in discrete-time format as developed in this study, can be expressed as follows:

$$u(k) = K_p e(k) + K_i \sum_{i=0}^{k} e\,(i) T_s + K_d \frac{e(k)-e(k-1)}{T_s} \qquad (6)$$

where:

- $u(k)$ = control signal given to the motors at the k-th time step,
- $e(k)$ = error calculated by the reference distance and the actual distance at time step k,
- K_p, K_i, K_d = proportional, integral, and derivative gains, respectively
- T_s = simulation time step (sample time)

The PID controller was built using the Simulink PID Controller block in discrete time with parallel architecture for the controller. The final controller gains were identified through a systematic tuning process using the Simulink PID Tuner App (Figure 105) to give an overall controller response that was stable, responsive and robust to modelling errors and disturbances from other sources. The resulting gains (see Figure 106) used in the simulation were:

- **Proportional Gain (P):** 8.8460
- **Integral Gain (I):** 0.4254
- **Derivative Gain (D):** -0.0141
- **Filter Coefficient (N):** 151.7018

These parameters were chosen to allow for a fast approach to the reference distance, small overshoot, and virtually no steady-state error.

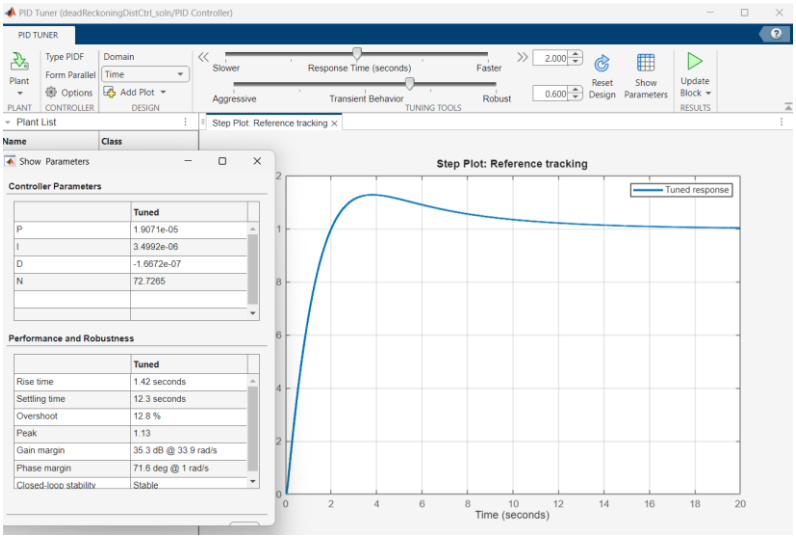


Figure 104: The PID Tuner has a block in Simulink to see the performance of the system in terms of rise time settling time, overshoot, stability margins, and it was useful to iterative tuning and analyzing.

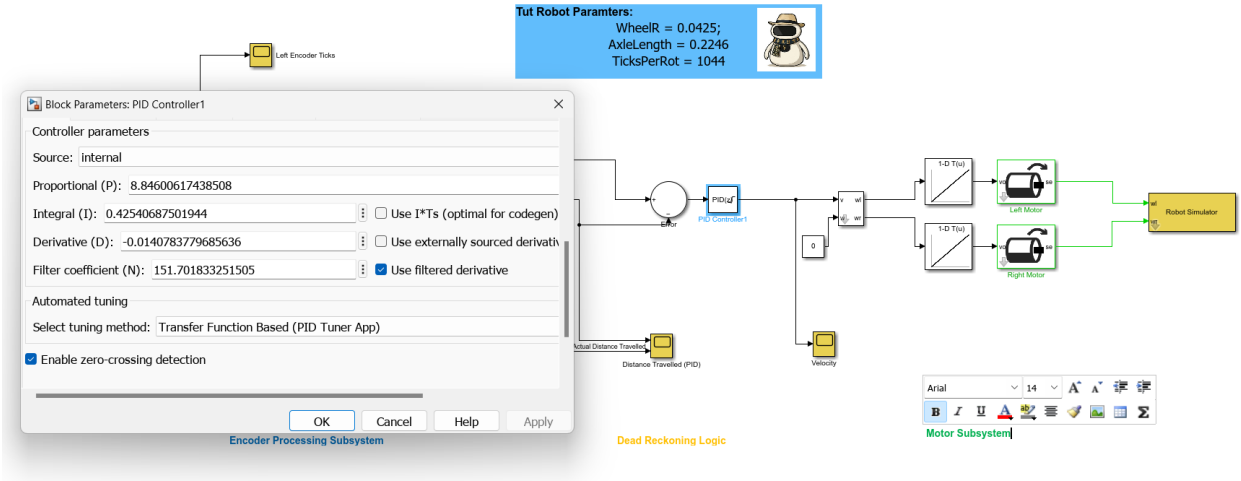


Figure 105: A PID Controller configured for the parameters, showing discrete-time gains (P=8.846, I=0.4254, D=-0.014)and the filter coefficient N=151.7 in the Simulink simulation to achieve dynamic distance specification.

The system architecture has not changed from previous control architectures in terms of the Encoder Processing Subsystem, Dead Reckoning Logic, and Motor Subsystem all unchanged. The PID controller then took the error signal - the difference between the reference distance of 1 meter and distance travelled - to compute a velocity command for the robot's motors.

Simulation results and their representation as distance travelled plots (shown in Figure 107) provide qualitative information on the dynamic behavior of the PID controlled system. The yellow curve shows that the Actual Distance Travelled is quickly rising towards the 1 meter Reference Distance, which is shown by the blue curve, and it achieves the target approximately 4.5 seconds after it started. It provides for an overshoot to approximately 1.015 meters, which is an overshoot of approximately 1.5% relative to the reference target. What is important to note is that the system quickly corrects this overshoot and settles in very close to the reference line with a steady state offset of less than 0.005 meters, which should be considered excellent steady state accuracy.

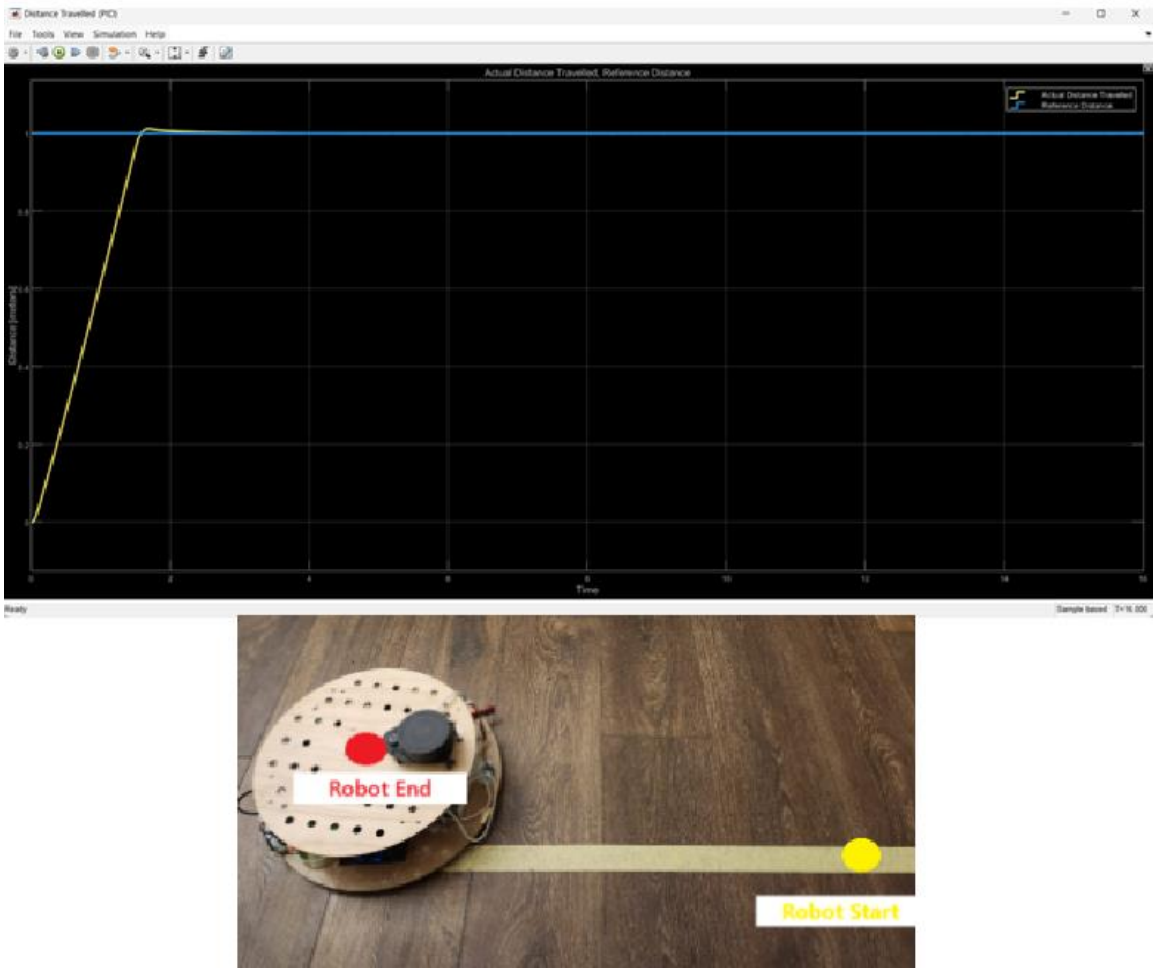


Figure 106: Simulation and Physical Protoype output from a PID controller with a quick rise time and very low steady-state error with a small overshoot which was eliminated quickly, demonstrating good transient results with close distance tracking.

The derivative term addition attenuates oscillations and attenuates the system response, resulting in a smooth trajectory line and little transient difference. The settling time was remarkably better than the PI action with approximately 6 seconds to achieve stable convergence. The derivative term provides additional system robustness by acting to mitigate

the effects of rapidly changing error, which is particularly important for real-world robotic systems that are subjected to disturbances and other modeling uncertainties.

In conclusion, the PID controller method performed the best overall in all the simulations. Using proportional, integral, and derivative actions, the reference distance was followed with low overshoot, fast settling time and low steady-state error. This control method represents a good trade-off between fast response and stability. It is the best control method for differential drive robots for real world applications in uncertain, changing environments.

5.6. Conclusion

The analysis of the distance control architectures established for the differential drive mobile robot shows a gradual improvement in performance with the addition of feedbacks. The open-loop control profile established a baseline, and illustrated limitations of large overshoot (approximately 1.3%) as well as not being able to stop at the distance as there was no feedback corrections in the system. This illustrated the value of feedback control for actual robotic applications in which the inevitability of mechanical uncertainties and external disturbances exist. The P controller provided basic feedback control, allowed the robot to correct for dynamic errors and reduced the time to reach the desired distance (about 4.8 seconds). However, steady state error and overshoot (approximately 1%) indicated the limitations of proportional-only control and needed future development.

The tuning of the PI controller improved steady-state performance because stabilization was accomplished by using the error signal with an integral contribution to eliminate the steady-state error. However, the integral action also produced overshoot during the transient response (~4%) and longer settling time due to error accumulation and phase lag. These findings illustrated the fact that there are trade-offs to integral control in particular and the concept of integral windup which can lead to system instability if not properly addressed.

In conclusion, the PID controller provided the most well-balanced performance and most robust performance among the testing architectures. By incorporating derivative action with the proportional and integral terms, the system obtained fast convergence to the reference distance (about 4.5s) with very little overshoot (~1.5%) and steady-state error (~0.005 meters.). Oscillation was minimized by the derivative term dampening the oscillations in its response and provided stability and robustness against dynamic disturbances.

In conclusion, the analysis indicates the importance of feedback control in achieving accurate, consistent, and stable distance tracking in differential drive robotic systems. In our previous

work, we evolved from an open-loop to a PID controller, demonstrating a stepwise transition of performance measures - rise time, overshoot, settling time, and steady-state error - reinforcing our design process. When applied in practice in environments without models, the PID controller stands out as the most effective choice, with stronger overall dynamic response and steady-state measurements, but tuning must be considered in relation to noise sensitivity to optimize operation.

Chapter 6 :Navigation and Autonomous Mobile Robot (AMR)

Continuing to the autonomous navigation section, which is an advantageous ability to add to a companion robot's features and capabilities. Inserting the autonomous navigation ability to the companion robot allowed the tendency of independent motion and safe transportation through different rooms of its indoor working environment, maintaining a robust and efficient companion robot for children with autism and elderly individuals. Moreover, such functionality allows the robot to respond to the user's command without the need for manual control, which may leave a gregarious impact to the user. In the current section, the navigation system development ought to be explained sequentially, as it was built using the Robot Operating System (ROS) with some key integrations like Odometry, Coordinate Transforms (TF), SLAM, and Path Planning, as illustrated in figure 107 below. Consequently, while following these sequential steps, an effective Autonomous Mobile Robot was developed after the insertion of the robot's URDF, Unified Robot Description Format.

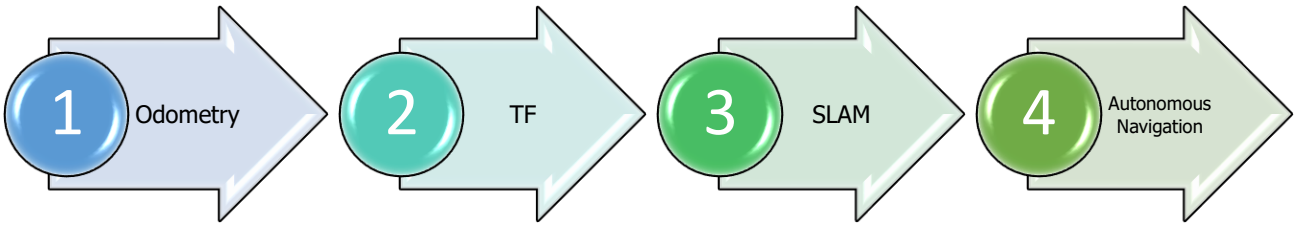


Figure 107 Autonomous Navigation Steps

6.1. Robot URDF

The Unified Robot Description Format (URDF) was employed to define the robot's defined physical and functional specifications in the Robot Operating System (ROS). To illustrate, simple information simulated includes robot dimensions, sensor positions (LiDAR, IMU), wheel configurations, and joint limits. Therefore, the URDF is employed to enable simulation, visualization, and integration into the ROS environment with precise results. As shown in figure 108 below, the 3D model of the URDF of the robot renders and visualizes the joints, wheel alignments, and sensor positions precisely in the simulation as placed in reality.

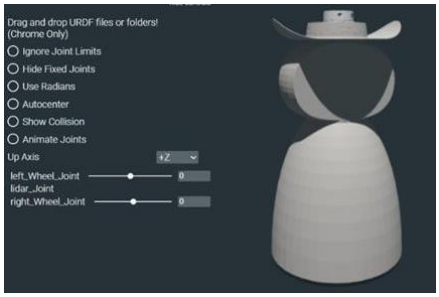


Figure 108 Robot's URDF

6.2. Odometry

A crucial component of the base controller is odometry, as it holds the encoders of the wheels to ease the documentation of the position and the orientation of the robot. To illustrate, odometry is considered the foundation of the pose estimation, position and orientation as mentioned before, as the robot is moving within its working environment. Consequently, to be able to estimate such information, multiple data ought to be collected from the onboard sensors such as the Inertial Measurement Units (IMUs) and the wheel encoders. In such case, the robot was designed to be a differential drive, which are two independently driven wheels which allow the robot to move in various directions such as rotation in place or translation. As illustrated in figure 109 below, the odometry depends on the data extracted from both the IMU and the wheel encoders in order to have a successful localization process later.

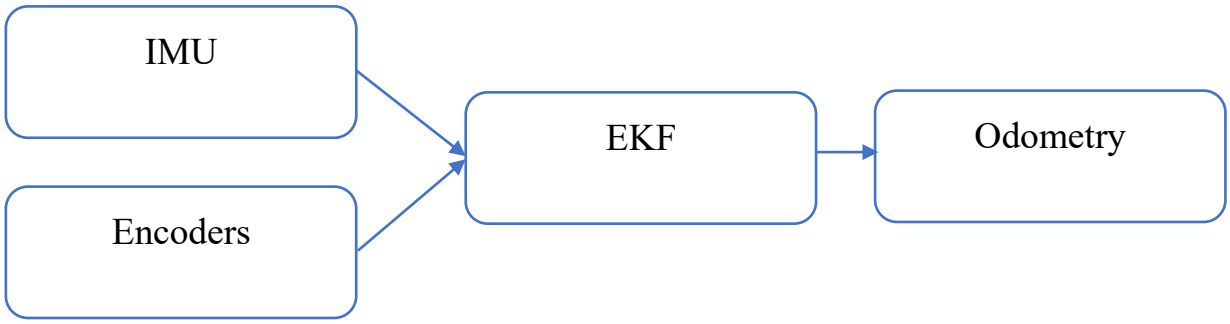


Figure 109 Odometry Process and Data Collection

As shown in figure 109 above, the odometry does not work with the raw data gathered from the onboard sensors; however, the data is filtered by passing through the Extended Kalman Filter (EKF). EKF is applied to fuse the raw data of multiple sensory inputs; moreover, it is usually implemented by applying the ROS package "robot_localization" which has multiple sensor topics to estimate the robot's pose. Therefore, by filtering sensor noise and uncertainties, the optimal state is determined, and an accurate pose estimation is then determined. As a result, EKF significantly reduces the errors and drifts that may occur in the system alongside another controller such as PID, as mentioned before, ensuring a smooth and accurate performance while the robot is autonomously navigating.

6.3. Coordinate Transforms (TF)

Generally, in a mobile robot system, understanding the spatial relationship between multiple coordinates frames is crucial for developing accurate and robust navigation and localization systems. As a result, tf libraries are often utilized to manage the coordinate transforms over time dynamically. Henceforth, for the companion robot developed, the core transformation chain follows the standard hierarchy, such that the odom is the "base_link", as illustrated in figure 110 below, while the map is the fixed global reference like the floor of the robot's environment. Furthermore, the base_link is located at the centre of the robot's base to demonstrate the odometry behaviour. For instance, when the robot is moved, the transform move accordingly between these frames to provide accurate position for other subsystems, such as localization, obstacle avoidance, and visualization.

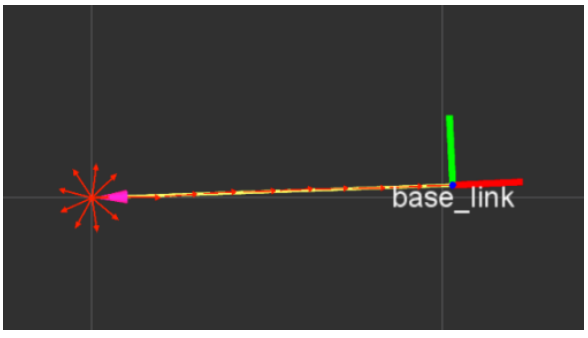


Figure 110 tf in RVIZ simulation [57]

6.4. SLAM and Map Creation

Consequently, to perform mapping, indoor environment was needed in order to have a specified map that the robot ought to follow and navigate in; therefore, an indoor environment was developed using Gazebo, as illustrated in figure 111 below. In addition, the URDF of the robot was inserted in the designed environment to start the mapping and localization process.

Gmapping algorithm was utilized due to its precision and reliability of developing efficient and accurate indoor maps. For instance, Gmapping effectively utilizes the data sensed by the LiDAR to build and update environmental maps regularly; however, such step would not take place correctly unless the robot is localized accurately; therefore, the robot usually localize itself at the same time it is mapping its surrounding environment. As mentioned before, IMU data is used to complement odometry data to correct position drift, making the map more consistent and reliable.

Figure 111 Gazebo indoor environment

Henceforth, for the robot to develop an internal representation of its environment, Simultaneous Localization and Mapping (SLAM) is utilized, which provide the ability to build a map while simultaneously estimating the accurate robot's position within its environment. The SLAM was applied using the Readings of the LiDAR along with the odometry information to generate a 2D occupancy map in order to localize the robot accurately and reliably, to develop an accurate map for the robot to store and user later. Additionally, slam_gmapping node used topics such as laser data and the tf as inputs to continuously construct a consistent map of the robot's environment for manual (teleop) or autonomous navigation for further updates.

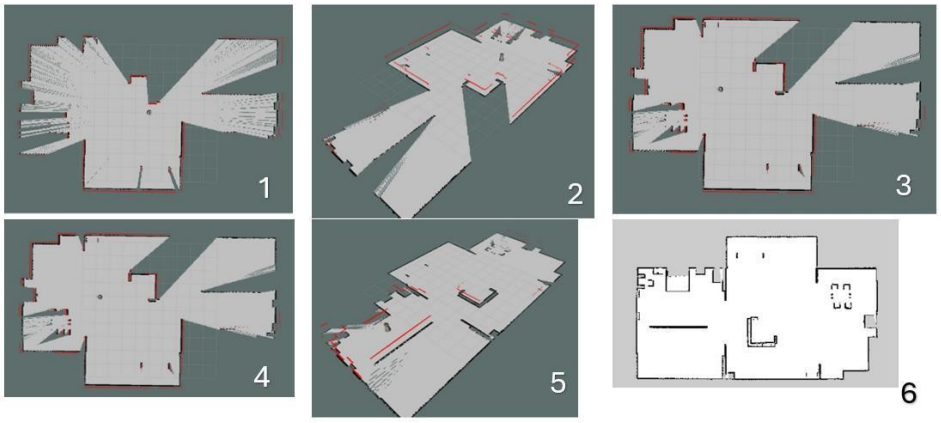


Figure 112 Mapping progress of the robot

As illustrated in figure 112 above, to visualize the mapping process, preferably, Rviz, a visualization tool, is utilized to monitor the real-time system's performance of mapping while using the teleoperation tools such as keyboard or joystick control to guide the robot through the environment to be able to capture the entire environment correctly. Figure 112 above shows the process of Gmapping alongside with the localization of the root, as from image 1 to image 4, the map is incomplete; however, as shown in image 5, the map is completed successfully along with the correct robot's localization; thus, the map is successfully saved, as shown in image 6, using the map_server utility. Subsequently, the saved map is then loaded by the map_server node to enable a robust localization and path planning.

6.5. Autonomous Navigation and Path Planning

By the utilization of the pre-built map, the companion robot ought to achieve a robust autonomous navigation system using the ROS Navigation Stack, which provides reliable capabilities for localization, global and local path planning, and obstacle avoidance. In the current process, the move_base node is the main component in the system, as it is considered the integrator of all navigation modules. Navigation utilized established path-planning algorithms, such as the Dynamic Window Approach (DWA) and global planners such as Dijkstra and A*, to give efficient collision-free navigation trajectories. The global planner is used to construct a path from the robot's current position to the goal set using the saved map, while the local planner ought to handle the real-time trajectory adjustments and obstacle avoidance within 1 meter from the robot in all directions. Therefore, the robot could respond dynamically to changing environments, tracking planned trajectories with great accuracy while also being able to dodge obstacles successfully.

Figure 113 Navigation of The Robot

Furthermore, to keep localization during the navigation process, the Adaptive Monte Carlo Localization (AMCL) algorithm is applied, which basically estimates the robot's pose withing the map by generating a weighed directions to keep or kill while the robot is moving within its environment with respect to the readings of the LiDAR laser beams and the saved map. The AMCL node was utilized to maintain the odom transform to assure the accuracy of the robot's global position throughout its movement. Henceforth, the system applies cost maps, global and local cost maps, to represent the areas where the robot could move in, ensuring the voidance of any dynamic obstacle, safely navigating through narrow areas, and occluded areas as well. As illustrated in figure 113 above, the robot path-planning functionality with ROS navigation stack. The first row illustrates path planning and navigation done in the ROS visualization tool (RViz) graphically showing planned paths (green lines) and robot trajectory. The second row shows corresponding images in the Gazebo simulator visually confirming good tracking of paths, successful collision avoidance, and correct execution of navigation commands in realistic indoor environments.

6.6. Conclusion

Entailing early evaluation and verification in the Gazebo simulation environment, thorough study was enacted under controlled environments. With good localization behaviour, stable path planning, and efficient obstacle negotiation in simulation, the current stage aided in tuning the navigation and localization performance iteratively by simulating the interaction with the real-world environment Gazebo-based simulations. As illustrated in figure 114 below, the system in image 1 has an incorrect localization; however, it was then tuned and updated to be able to localize itself correctly and efficiently.

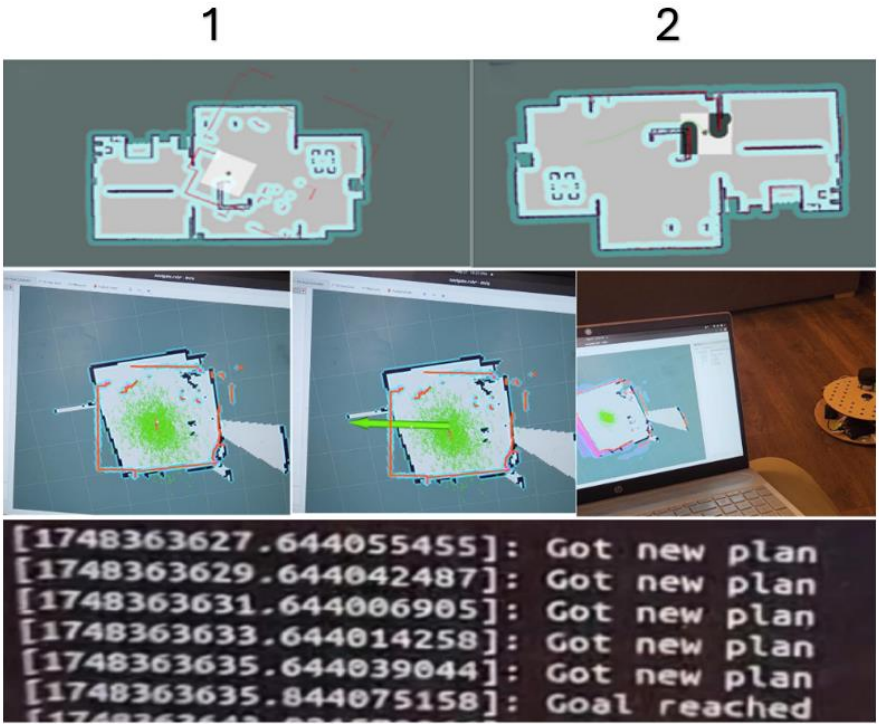


Figure 114 Localization and Navigation Process Performance

The localization process of the mobile robot began with no initial positional awareness, as shown in the AMCL visualization. To initiate localization, an initial pose estimate was manually set in RViz, providing a rough starting point for the AMCL algorithm. As the robot started moving within the mapped environment, the adaptive Monte Carlo localization (AMCL) algorithm began refining its estimate by discarding inaccurate particles and reinforcing the correct ones based on sensor feedback. Gradually, as movement continued and observations matched map features, the particle cloud converged, indicating that the robot successfully localized itself within the environment.

Integration presented sensor fusion accuracy and computation demands for real-time navigation. These were addressed through precise calibration of sensor parameters and

optimization of algorithm parameters. Further development includes integration of more sensors, more sophisticated algorithm calibration, and extensive real-world tests to achieve maximum reliability and autonomy. However, since the robot was designed to move through static environment, Gmapping was an efficient utility to be applies, as it develops accurate indoor static environmental maps, providing optimal performance for a companion robot. As a result, the current performance of the robot's indoor navigation system has proved to have high robustness and efficiency, as the robot could autonomously navigate through the environment automatically effectively, which would be mentioned further. Since the robot was designed to move through static environment, Gmapping was an efficient utility to be applies, as it develops accurate indoor static environmental maps, providing optimal performance for a companion robot.

Chapter 7 : Artificial Emotional Intelligence System

In this chapter various artificial intelligence modules have been articulated in the companion robot, and indicated how speech processing, sentiment analysis, and the chatbot function combine to yield an interactive, emotion-enhanced experience for users. The modules in combination enhance ability of the robot to perceive, respond and engage with humans, who may find the experience a natural one.

7.1. AI Integration Overview

Artificial Intelligence has been employed in this project to make the companion robot more interactive and human-like. Instead of only pre-programmed instructions, the robot was endowed with the ability to understand speech, reply naturally, and convey emotions through face and voice.

- Speech-to-Text: Speech is taken from the user and converted into text.
- Chatbot: response is generated based on the input text from the user.
- Text-to-Speech: The chatbot response is taken and converted into hearable speech.
- Sentiment Analysis: detecting the user sentiment based on the text tone.
- Weather Detection: Weather information is fetched from the openweather api web upon request from users.
- Lip Sync and Facial Expressions: Emotions are expressed through dynamic facial expressions, and lips are synchronized while speaking.

All of these features as illustrated in Figure 115 were combined in a manner such that they functioned harmoniously together so that the robot could operate and respond nicely. This AI helped to make the robot more useful, expressive, and emotionally engaging.

Figure 115 Project Features

In order to have a better view of what goes on within the companion robot, a system flowchart was created to illustrate step by step the procedure witnessed during interaction. The entire process from receiving user input, process it using AI modules, and provide appropriate feedback in the form of speech, expressions, and actions. Each step of the interaction process when the user's voice was picked up to when the robot responded was broken down and depicted. The flowchart in Figure 116 describes how the modules such as Speech-to-Text, Chatbot, Sentiment Analysis, Text-to-Speech, and Expression Control were interconnected and how data was passing from one module to another. With the help of the flowchart it is easy to comprehend the whole behaviour of the robot while conversing.

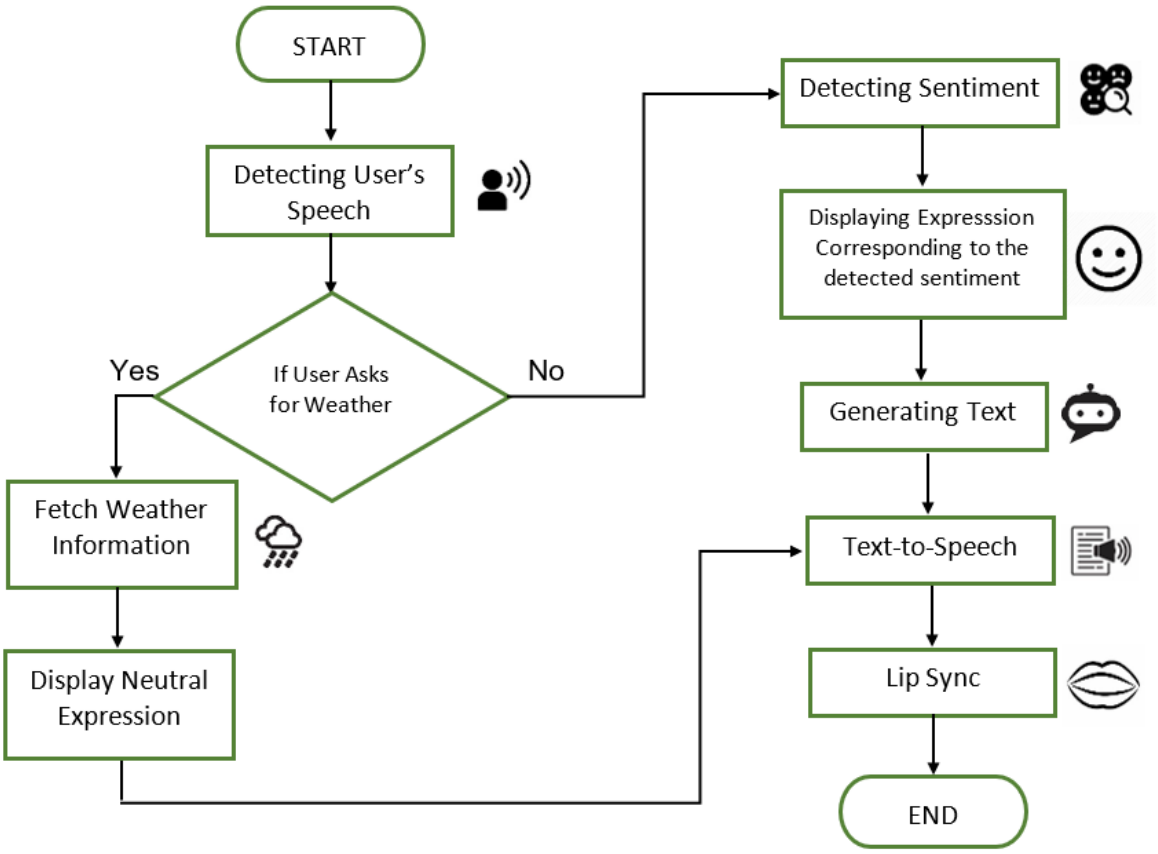


Figure 116 Robot Operation System Flowchart.

7.2. Speech-to-Text Module

In order to enable the robot to understand natural speech a Speech-to-Text system was utilized. Several models were tested such as Whisper AI, Google Speech-to-Text, Vosk, DeepSpeech, and PocketSphinx. All the above models were compared based on the accuracy WER percentage (word error rate). Whisper AI was the most accurate and dependable with 9% WER. It could read out different accents, levels of background noise, and sentence lengths with very few mistakes, Google's service performed well but required reliable internet connection, Offline models such as Vosk, DeepSpeech, and PocketSphinx were also tested but they performed extremely poorly especially when conversational language was input. Therefore, Whisper AI was selected and used to convert voice input into text. User voice is captured through a microphone and converted to text by Whisper and further processed by the chatbot, this helped the robot to be operate better when user asks questions.

7.3. Sentiment Analysis

To enable the robot to respond empathetically and naturally, sentiment analysis was incorporated into the system. A basic model was first used that was capable of detecting only three emotional tones which are positive, negative, and neutral. It was later found to be too limiting during testing as most of the human emotions could not be accurately detected or represented in the actions of the robot. As a result, the model was replaced with a more advanced sentiment classifier known as SamLowe’s 27 model. This model was trained to detect a wider range of human emotions including emotions such as joy, disappointment, approval, anger, gratitude, and many others. With this model more detailed emotional states were recognized from the user’s speech allowing the robot to adjust its tone, expression, and responses accordingly. This one significantly increased the level of quality engagement and made the robot more emotionally intelligent.

7.4. Chatbot

A chatbot was implemented to allow the robot to respond with intelligent, friendly messages. Different models of chatbots were experimented with such as Gemini Pro, ChatGPT-4o, LLaMA. Gemini Pro is free and easy to deploy. However, the responses may not be accurate enough. ChatGPT-4o gave the most accurate responses. It was well acquainted with the user, gave good answers, and even responded emotionally and nicely. It needed to be paid for and had to have access to the internet at all times. LLaMA was also tested because it could work offline, but it demanded a high-performance computer with high-processing powers, something

that was not compatible with the robot's hardware. Finally, ChatGPT-4o was used as the main chatbot because it gave the best experience.

7.5. Text-to-Speech Module

To give the robot the ability to talk in a friendly and supportive way, a Text-to-Speech system was used. The pyttsx3 python library was tested as a first option since it is offline and easy to use. The voice, however, did not sound very clear or friendly, and was quite robotic. Because of this, it was not used in the final model instead of that eleven Labs Text-to-Speech was used. It offered various, natural, expressive, friendly and supportive voices that made the robot's voice more human and friendly. Chatbot replies were converted into speech using eleven labs platform and spoken through the speaker of the robot. This enhanced the overall experience and resulted in more realistic conversations with the robot.

7.6. Weather Information Retrieval Feature

For making the robot more useful, informative and helpful, a feature for getting the weather information was added. When the user asks for the weather in a certain city, an online search is automatically performed in openweather using its online api. Weather report is fetched from the OpenWeather API which include temperature and weather conditions like rain or sunny when the user asks for. The city name is extracted from the user query. Then the OpenWeather API is called and then the weather data is retrieved as a response. Then a brief and simple weather report is prepared and spoken by the robot through the implementation of the TTS system. This allows the robot to answer questions like "What is the weather like in Cairo?" in a supportive and friendly way.

7.7. Facial Expressions Display with Lip Sync

To grant the robot more humanlike and expressive personality, facial expressions and lip sync capabilities were incorporated. Facial expressions as demonstrated in Figure 117 were shown based on the detected emotion from the user's message. Sentiment was assessed by sentiment analysis model and then converted to a specified expression through a mapping established before. If sadness was found, a sad face is displayed. This helped the robot to react visually according to the user's mood. In lip sync (see Figure 118), the chatbot's response was first converted into phonemes (the raw sounds of speech). One-to-one mapping was set up between every phoneme and an equivalent mouth shape image, using a special phoneme-to-mouth mapping. During processing the phonemes the equivalent mouth shapes were shown in

equivalent order and speed, creating the illusion of synchronized speech with the voice. This generated a more natural speech for the robot and made it fun to talk to.

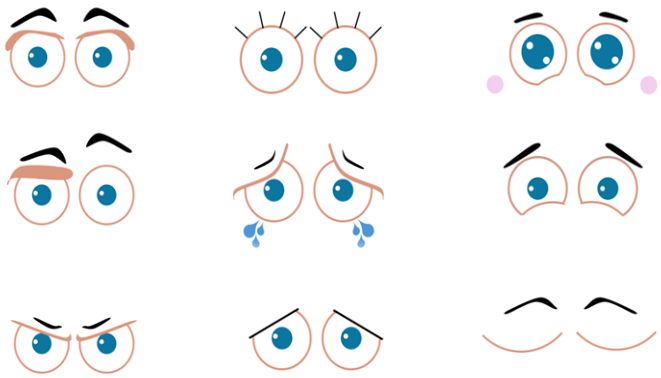


Figure 117 Robot Different Expressions

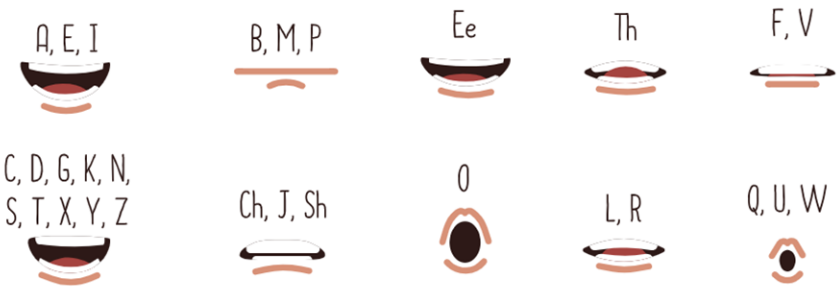


Figure 118 Lip Sync Phonemes Images

Chapter 8 : Graphical User Interface (GUI)

The graphical user interface (GUI) was implemented to ensure a smooth and seamless Human-Robot Interaction (HRI) between the user and the companion robot. The GUI was primarily designed to suit both categories, which were previously mentioned, the individuals with autism and elderly people. Subsequently, the GUI was implemented to illustrate visual communication and interaction which simplifies the multiple and complex functionalities into an engaging and an entertaining experience for the user. As a result, to implement the aim of the GUI, it was designed to priorities emotional responsiveness, minimal cognitive load, and clarity to match the limited technological familiarities of both autistic children and elderly individuals; therefore, users could handle the HRI in a more engaging and comfortable way to be able to take advantage of the robot's multiple features easily.

GUIs were mainly developed to be a control panel for the user; however, in the current state, it was not only designed to control the different features of the robot effectively, but to also act as the bridge at which trust is created between the robot and the user through the cameras and the personalization of the chatbot according to the users preferences. Moreover, the GUI has multiple features, which would be demonstrated later in the current section, such as language selection, playing games, only displaying the chatbot facial expressions, and other features. In addition, the user could switch the robot from a companion robot to a service one, as it may navigate through different rooms, saved in the system after initial mapping, according to the user's command. Therefore, through thoughtful integration between the entire subsystems and features of the robot, the GUI was implemented to contribute the robot's ability to successfully deliver an effective and supportive emotional companionship with the user.

8.1. Design Considerations and Technical Implementation

The initial aim of the GUI design was to be suitable for both elderly people and individuals with autism. The design had to be user-centered; in other words, the design ought to be clear, simple, and friendly in order to uphold the technological knowledge restrictions of the targeted individuals. Furthermore, the design was implemented for minimal cognitive load. Large buttons and simple layout were implemented to be clear to visualize, as illustrated in Figure 119 below. The design maintained a neutral theme to illustrate simplicity with high contrast between colors to easily determined icons.

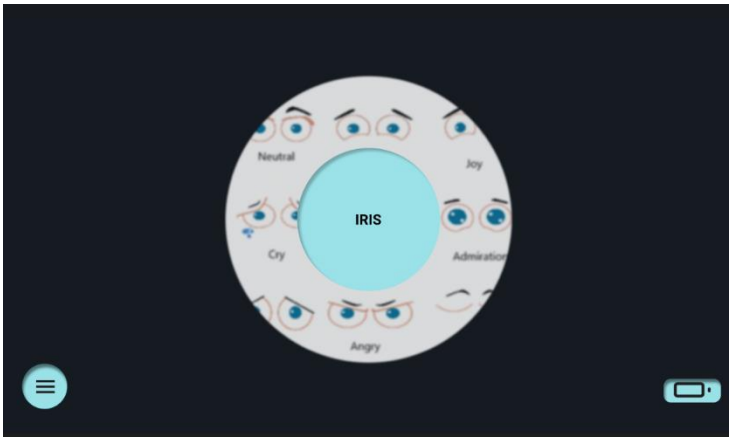


Figure 119 IRIS Home Page

It allows the user to choose among different features, modes, and settings. The GUI was implemented using different python libraries such as Tkinter and Pygame. Both were used to apply certain functions, as pygame was applied to open the emotional features window and control the lip sync transitions, as mentioned before, due to its ability to display animation smoothly. in addition, it was also utilized for the sound effect and the image capture. In contrast, Tkinter was applied to insert the images of the icons designed and applied the functionality of the different images such as the hover effect for each icon, as illustrated in Table 22 below.

Table 21 Python Libraries Implementation

Component	**Library Utilized**
GUI Framework	Tkinter
Sound Effect	Pygame
Facial Expression Display	Pygame
Language Transition	Tkinter
Theme Transition	Tkinter
Game Integration	Tkinter
Autonomous Navigation Command	Tkinter

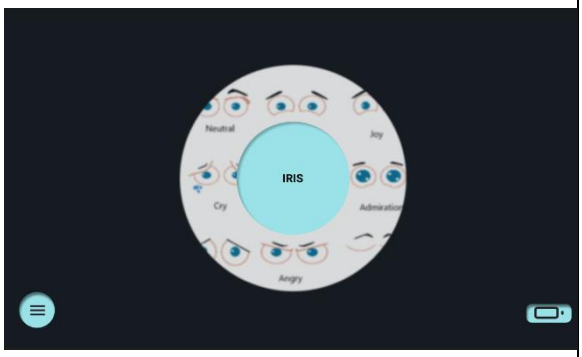
8.2. GUI Features and User Interaction Flow

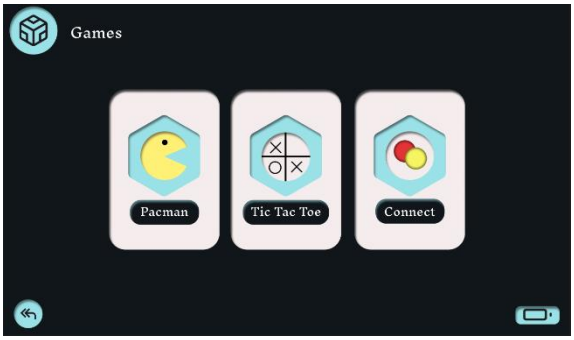
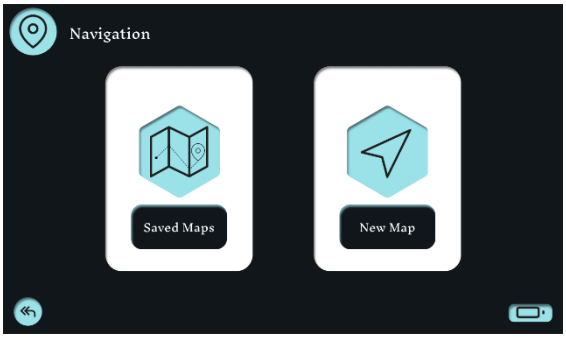
The Graphical User Interface was initially developed to provide multiple features for the user in order to maintain a smooth, entraining, and friendly HRI. There is a specified flow which

initially functions to start the program, as a short welcoming speech, an introductory one to introduce the robot, the insertion of the user's image and personal preferences and hobbies, the language selection (English or Arabic), and other features. One of the principles was to develop a personalized and customizable User Interface to meet the specific needs of elderly people and children with autism.

The features of the UI system include the following, as illustrated in Table 22 below:

Table 22 Features Of the GUI

Page	Functionality	Image from the GUI
Home	The home page displays the facial expressions and the menu button for further selections.	
Settings	The settings page has different features such as: 1) Language: Selection between English or Arabic Languages 2) Theme: Selection between Light or Dark 3) User Addition: 2D Camera is attached to the system which captures the user's image and receives the user's preferences	

Games	An additional feature was inserted for extra entertainment; therefore, games were added to the UI system, such as: 1) Pacman 2) Tic Tac Toe 3) Connect 4	
Navigation	The Navigation page has two main features, such as: 1) Navigate the saved maps in the systems 2) Add a new map	

Finally, the GUI forms a connection bridge between the user and the robot. Multiple functions were maintained to be presented in the UI system to ease the communication and the experience of the user to assure personalized, entertaining, and smooth interaction. Various features and functionalities may be added for more customization and more efficient experience.

Chapter 9 :IRIS IoT Monitoring System

This chapter introduces the IRIS IoT System designed to enhance the companion robot's environmental awareness through real-time monitoring and intelligent response. It outlines the system's data acquisition, processing, cloud integration, and alarm management, highlighting how these features empower the robot to operate safely and autonomously in indoor environments.

9.1. Introduction and System Overview

Continuous environmental monitoring is a vital context for socially-assistive robotics that increases user safety and comfort by allowing the robot to sense and act upon indoor conditions. The IRIS IoT System - for Indoor Monitoring provides the function to monitor in real-time through data acquisition; cloud communication; and data visualization as in Figure 120, provided opportunities for integration into the architecture of the IRIS robot.

The IRIS IoT system solution includes the essential work to connect the hardware and software of the robot with cloud-based tools in an easily provided modular design. This means that the environmental factors of temperature, humidity, air quality, and mandatory fire detection and on-going analysis can be maintained continually. Similarly, this data-driven approach facilitates user intervention before challenges arise, for the benefit of older adults, and for individuals with Autism Spectrum Disorder.

In conclusion, the IRIS IoT System provides a unique solution for real-time environmental monitoring by including a wide array of sensing capabilities, reliable communication, and very suitable data-viewing capabilities to provide better function of the robot and greater user experience.

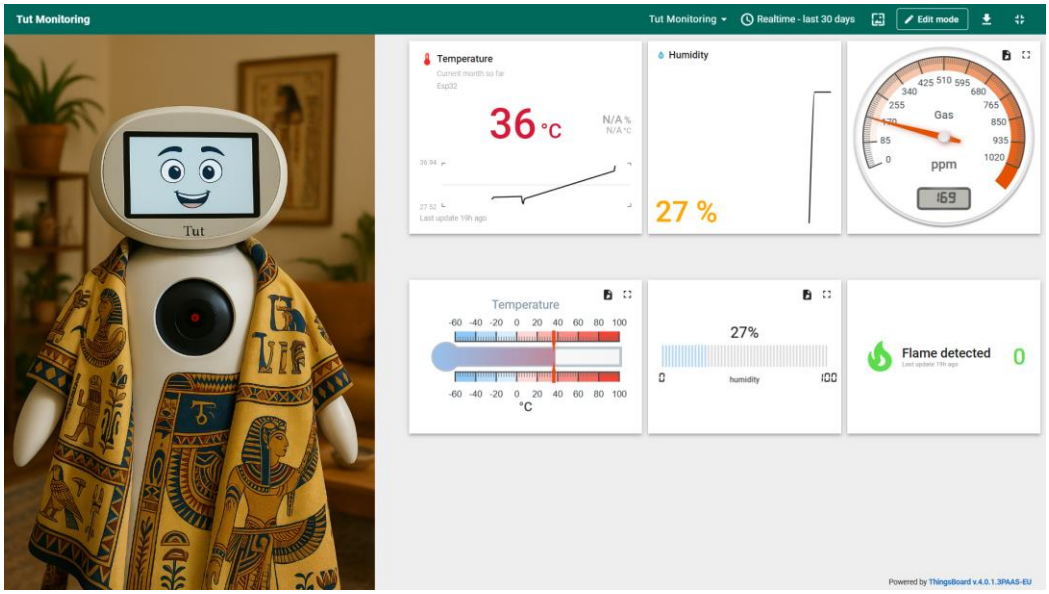


Figure 120: IRIS IoT Monitoring Data Visualization

9.2. Data Acquisition, Processing, and Cloud Integration

The IRIS IoT System leverages the ESP32 microcontroller in the data (Figure 121) acquisition subsystem to acquire the sensor data from the following sensors:

- **DHT11**: measures temperature (C^o) and humidity (%) .
- **Analog Gas Sensor**: measures air quality in ppm
- **Digital Flame Sensor**: Detects fire hazards.

On-board processing will filter and validate the sensor measurements to remove excessive noise and accidental triggering. Also, the data will be secured and transmitted via MQTT over TLS access token authenticated transport layer security ensuring secured transport between the device and cloud.

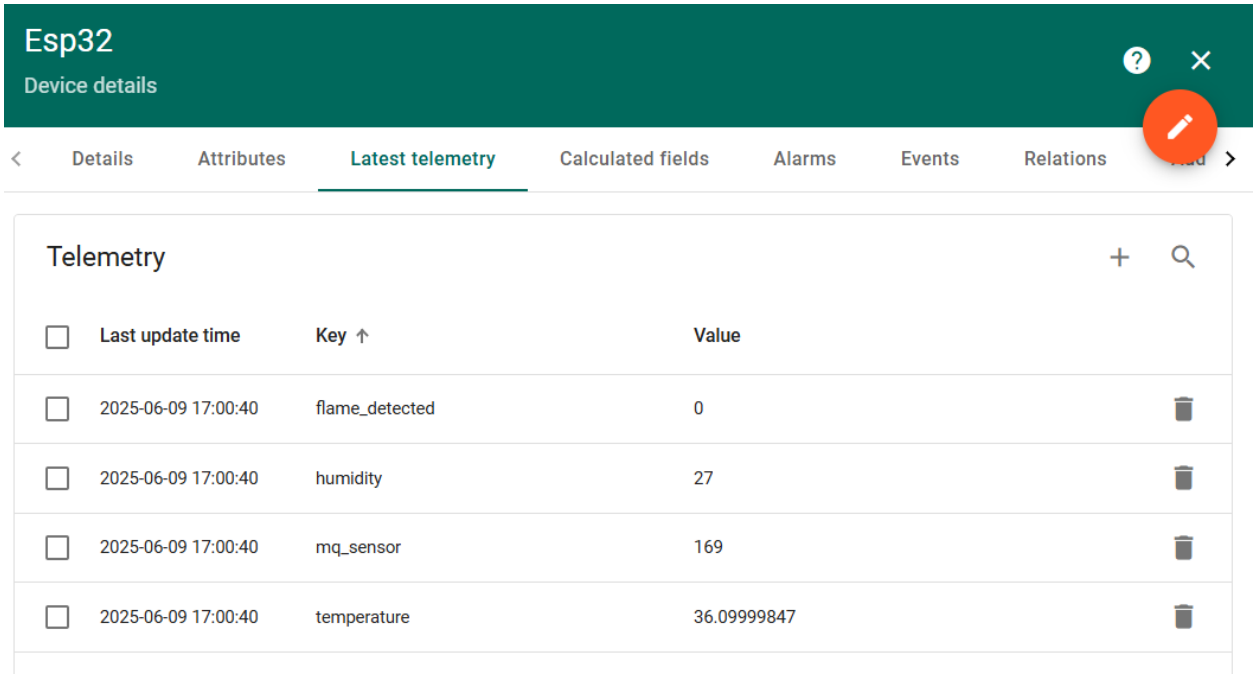


Figure 121: Data telemetry from sensors

The ThingsBoard Cloud Platform supports the management of devices through profiles for real time monitoring and credential management. The IRIS Monitoring Dashboard displays sensor data and supports historical data viewing of time windows between 10 minutes and 30 days as in Figure 122. This provides not only a capability to respond to environmental events as they happen, but also allows us to identify historic trends in environmental data. This architecture integrates all of these factors so that the IRIS robot has a reliable, scalable, and secure view of the environment, enhancing its ability to operate autonomously and safely in indoor environments.

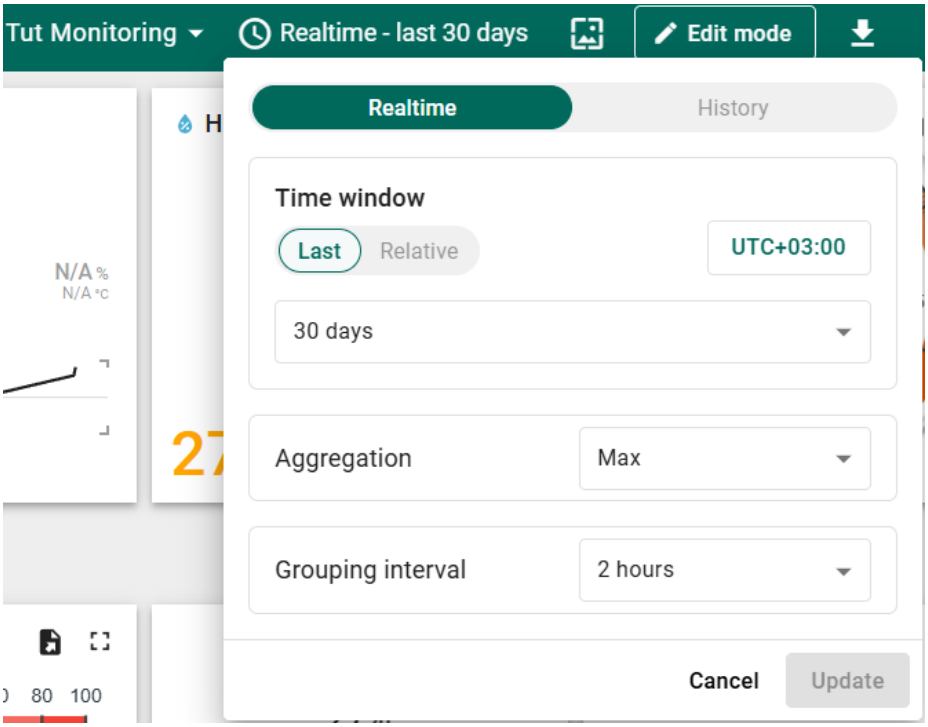


Figure 122: Time window customization of period

9.3. Real-Time Monitoring and Alarm Management

The IRIS IoT System's IRIS Monitoring Dashboard allows the user to visualize real-time temperature, humidity, gas concentration levels, and flame detection from the ESP32, with time windows ranging from 10 minutes to 30 days. This means the person can view their immediate conditions and then make long-term changes.

Alarms created for the IRIS Monitoring Dashboard are configured with varying severities for Critical, Warning, and Informational. A Critical alarm can be triggered upon rapid changes to the environmental data, such as exceeding exceeding humidity limits at over 33%. The active alarm panel also displays metadata including timestamp, device ID, type, severity, and status of the alert demonstrated in Figure 123, which assists the user in pinpointing which action to take when an alarm is generated during their long-term monitoring activity.

Figure 123: IRIS triggering alarms

ThingsBoard alarm management utilizes rule chains to select rules dynamically based on the unpacking of a sensor data stream (i.e., humidity or other thresholds). Therefore, a rule chain allows for a continuous monitoring process with respect to alarm generation and if there is a defined alarm threshold, the alarm can be triggered and cleared automatically with a rule chain. As such, the operational efficiency of the user will largely be predicated upon the Rule Chain and alert states previously defined when an alarm is created, which makes the user and IoT system ideally operationally responsive, producing positive safety outcomes.

9.4. Conclusion

The IRIS IoT System is established using secure Wi-Fi communication, TLS encryption, and access token authentication to ensure that communication and data are safe and the data integrity is maintained. When using the IRIS IoT system, environmental data is transmitted securely and maintain reliability for the system.

Benefits include real-time monitoring, cloud scalability, an easy to use IRIS Monitoring Dashboard, modularity for future sensor options, etc. Users can customize dashboards and investigate trends using varying time windows to provide flexibility.

Limitations include the system being based on strong internet connectivity, threshold tuning of alarms can require user adjustments to limit false acceptances, unassigned alarms currently need enhanced automation to facilitate immediate attention.

The IRIS IoT System, with security, reliable monitoring, and adaptability characteristics, are important to be considered as part of the IRIS robot's environmental awareness response.

Chapter 10 : Experimental Work

The chapter presents the entire procedure of the experiment that was carried out to test the performance, reliability and feasibility of the TUT the companion robot in the real world. The work is organized in two main parts, i. e. simulation-based analysis and physical prototyping with live testing. Simulation phase comprises of control modelling in MATLAB/Simulink, stress testing with Finite Element Analysis (FEA) and autonomous navigation simulations in Robot Operating System (ROS) and Gazebo framework. Such simulations played an important role in the tuning of control algorithms, confirmation of mechanical robustness, and navigation behaviour testing in changing conditions. After simulation, a full-scale prototype was produced through additive manufacturing and put to test in real worlds to determine its mobility, affective communication and reaction to voice and sentiment commands. This two-way validation of virtual and physical makes sure that the robot does not just work properly on theory but also works well in the realistic conditions that the users will be exposed to.

6.1 Analysis and Simulation

As part of ensuring that TUT the companion is robust and fault-tolerant before the real implementation, a number of simulation and analysis methods were taken into use. These included mechanical strength, the accuracy of motion control, and autonomous navigation within the indoor environment. The simulation process performed early in the design process was able to verify the design and optimize parameters to be used in the real world with very little physical prototyping errors.

6.1.1 Mechanical Analysis Finite Element Analysis (FEA)

The major structural parts of the robot were performed by Finite Element Analysis to ascertain mechanical integrity under operational loads. Simulations were all done on the assumption of static forces applicable on normal handling, transport, and interaction stresses in an indoor environment.

- Torso Component: Bristled with a vertical peak weight of 20 N to represent inner bracing and manipulation pressure. The peak stress was 135.6 kPa, which is much lower than the yield strength of PLA (=51 MPa), ensuring the structural safety.

- Head Assembly: A load of 30 N, which corresponds to LCD mounting and interaction forces, caused the maximum stress of 154.2 kPa, which is also far below the material safety limit.
- Base Frame: Made out of contour wood and loaded with conservative force of 100 N. The structure was subjected to stresses well within the material capability of 20 MPa, and thus it is confirmed that it can sustain the entire system mass (~8 kg).

These tests and analyses reaffirmed that the mechanical design of the robot, including the choice of PLA to 3D-printed components, was safe, stable, and lightweight construction, applicable to indoor use.

6.1.2 Control and Motion Simulation PID Tuning

The movement of the robot is controlled by the differential drive mechanism where a PID controller is used to control the linear and the angular velocities. This control loop was modelled and simulated in MATLAB/Simulink in order to achieve accurate navigation.

Parameters of Simulation:

- Initial pose: X = 0m, Y = 0m, theta = 0 degree
- Target location: X = 2□m, Y = 0.7m
- Solver: ODE3 (BogackiShampine) to real-time simulation

PID Performance:

Positional and angular errors were reduced to a minimum with the controller. The robot was able to traverse in a smooth curved way showing real time adaptability and tracking precision towards the target. This verified the tuning of the PID parameters and also ensured that the motor commands can surely guarantee stability of control despite small perturbations.

6.1.3 Autonomous Navigation ROS and Gazebo Simulation

The assessment of autonomous mobility was performed in the virtual indoor environment based on the Robot Operating System (ROS) with Gazebo simulation.

- URDF Model: The physical geometry of the robot, joint limits, wheel spacing, and sensor positions were replicated with a complete URDF (Unified Robot Description Format). This offered life-like movement and simulation of sensors.
- AMCL localization: Adaptive Monte Carlo Localization (AMCL) was used to localize the robot using LiDAR and IMU sensor measurements.

- After carefree movement, scattered particles initially encountered one another in the proximity of the real site of the robot, and this portrayed a great level of localization accuracy.
- SLAM based mapping (G-mapping): The G-mapping package enabled simultaneous localization and mapping.
- A sparse-to-dense map generation sequence was visibly possible, and this confirmed that the robot was able to gradually improve its knowledge of the environment by using LiDAR measurements and odometry updates.
- Path Planning and Obstacle Avoidance: ROS navigation stack with Dynamic Window Approach (DWA) local planner and Dijkstra/A* algorithms global planner enabled the robot to mix and match collision-free paths in dynamic environments.
- The simulations in RViz and Gazebo demonstrated the robot moving dynamically to evade obstacles, which implies that its motion planning is dependable.

6.1.4 Simulation Insights

These simulations taught a couple of things:

- The mechanical design is capable of sustaining the normal operation loads without collapsing.
- Under simple kinematic limits, PID-based control enables exact path following
- Autonomous navigation stack allows efficient self-localization, mapping of the surroundings, and making decisions in real time.

6.2 Prototyping and Testing

After the successful simulations, a physical prototype of TUT the companion was produced and experimented with in order to confirm its functionality in the real-world setting. The fabrication procedures, assembly procedure, experimental arrangement, test procedures and discussion of the results with regard to the theoretical expectations are explained in this section.

6.2.1 Prototype Fabrication

The robot chassis was designed and produced via additive manufacturing (3D printing) using PLA filament, as it was selected because of the user-friendly features, the uniformity of the construction, and the adaptation to the indoor environment. The other components such as the base and inner plates were of contour wood that is light weight.

- Joining Technique: It was constructed as modular stacking plate. Heat inserts were molded into the 3D-printed pieces to allow the easy removal and installation using threaded screws.
- Bottom Plate: In this plate, the primary electronics box is mounted, which contains the Jetson nano, arduino mega, and motor drivers.
- Side Plates: They are bolted to the bottom plate with spacers and the vertical structure is placed on them.
- Motor Mounts: The motor mounts are screwed onto the bottom side plates in order to hold the differential drive motors.
- Display and LiDAR: they are attached to the head unit to communicate with the user and perceive the surroundings.
- Upper Cover: attached all the internal modules and improved the cosmetic appearance.

6.2.2 Experimental Setup

The test environment was a constrained indoor environment. The factors, which came into consideration, were as follows:

- Mobility Tests: Navigating through pre-designated courses comprising of turn, stop and obstacle-avoidance incidents.
- Navigation Validation: LiDAR and IMU based SLAM based mapping and localization in an area.
- Voice Interaction Tests: OpenAI Whisper and ChatGPT-4 were applied in real-time to carry out speech to text translation and deliver chatbot responses.
- Emotional Feedback: Skinned facial expression and emotional state caused by the detected emotion.
- Power Consumption: Measured battery output and component behaviour.

Chapter 11 : Analysis and Discussion

The purpose of the project was to develop an emotionally intelligent companion robot which could assist elderly people as well as children with autism by interactive communication and autonomous navigation for further assistance. Therefore, the primary hypothesis was that user involvement, psychological well-being, and independence may be enhanced in residential settings by integrating a reliable and reliable navigation system with a sentiment-aware AI chatbot.

As mentioned in chapter 10 above, according to the evaluation results, the robot was able to successfully travel between rooms using the indoor map detected and developed using SLAM alongside with its autonomous navigation system while avoiding obstacles and reaching its goal; furthermore, accurate localization and secure navigation were provided by the algorithms Gmapping and AMCL, while the Extended Kalman Filter increased odometry precision within a small range to prevent accidents. Moreover, the chatbot employed dynamic facial expressions and synchronized lip movements display for relevant respond, as a reflection to the user's sentiment and interaction which aided in increasing empathy; and finally, the GUI permitted accessible interaction and aided in maintaining entertainment and smooth interaction. When everything was considered, the system's navigation and interaction modules were found to function robustly and reliably.

11.1. Findings and Results

According to the data shown above, the implementation of an artificial intelligence powered chatbot along with the dynamic facial expression which were displayed according to the user's sentiment analysis and the lip synchronization for the generated answer have developed a genuine feeling from the user towards the companion robot, as illustrated in figure 126 below. That connection was built due to the robot's ability to reflect the actual mood of the user, which encouraged the user to grow empathy towards the child book while interacting. Accordingly, the users have concluded that they felt comfortable while interacting with the robot, especially people with cognitive and developmental challenges.

Figure 124 The Robot's Facial Expressions.

Technically, the robot proved that the system was designed and implemented successfully. In addition to the AI powered chatbot, the navigation system behavior was also successful in indoor static environments. The autonomous navigation system had effectively performed in real life along with the rust components such as Gmapping, AMCL, and coordinate Transforms; Addition the comma the EKF fusion of the IMU and the encoders data ought to reduce the localization drift with the implementation of another controller such as the PID, as mentioned above. Thus, the robot has the ability to reach destination and navigation goals within a small range, as primarily designed. As illustrated in figure 127 below, the magenta arrow in image 1 is the desert goal and in image 2 the green line is the successful path planning done by the robot to reach the goal, which was reached in the simulation.

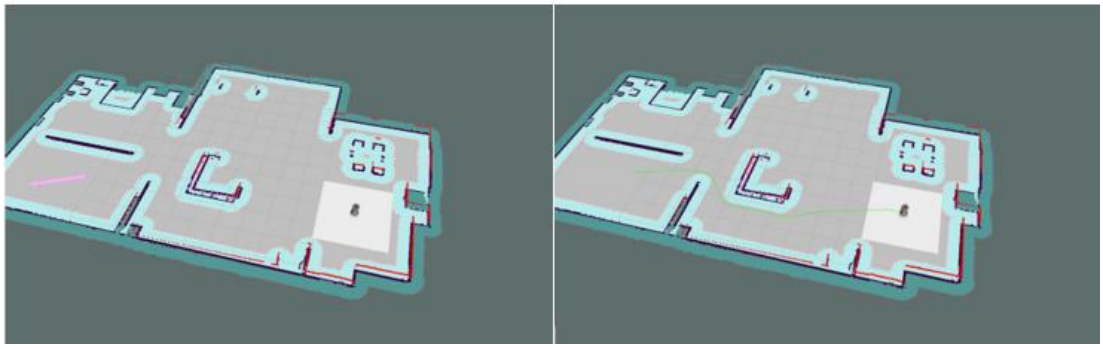


Figure 125 Successful Path Planning

Finally, as the visualization in Rviz is a tool to estimate and evaluate the robot's performance in real-world, it could be concluded that the robot had successfully employed the path planning algorithm developed in the navigation system. Therefore, the navigation system ought to be considered a robust one for the indoor environment, in the current case. As a result, the findings of the experimental work and evaluation had clearly illustrated the success of the subsystem implementation and integration. As the AI chatbot system along with the autonomous navigation system to complement each other simultaneously to develop a successful emotional and social assistive companion robot, it was determined that the prior aim of development a companion robot had successfully been implemented and developed.

11.2. Discussion

To proceed, the findings of this study demonstrated the current potential and need of current autonomous robotics and the development of an affective computing in real assistive applications, as the current case. The system minimizes users' reliance and enhances their daily activities through providing companionship and assistive presence whether in navigating indoor environments or being emotional and social support for the user. Furthermore, the GUI's

primary aim was to develop an interface which focuses on entertainment, emotionally responsive visuals, and predictable interaction flow providing a comforting and engaging experience that encourages cognitive stimulation for youngsters with autism and may satisfy social support for elderly people as well.

Finally, the current research sets the path for future integration with personal assistants, home automation, and healthcare monitoring systems, permitting additional support for certain individuals which need assistance and support. However, the current companion robot development ought to be built for educational purposes, this framework could be utilized in multiple domains such as care facilities, schools, or therapy centres where spatial autonomy and emotional involvement are crucial. Unfortunately, due to the AI HRI involvement of a companion robot, multiple aspects ought to be considered; in particular, ethical concerns ought to be taken into consideration, especially for companion robots that involve confidentiality of data, emotional manipulation, and long-term psychological effects.

11.3. Limitations of Study

During the stages of development and testing, few boundaries were identified. For instance, the program's versatility was restricted by the sentiment detection component's dependency on user-provided personality characteristics, as for further development, the camera ought to be employed for the real-time recognize the individual’s emotion along with the sentiment detection continuously to be able to develop a robust system with higher accuracy. Henceforth, the autonomous navigation system might additionally require modification or improved in order to handle dynamic barriers, moving people, or multi-floor configurations, as it was designed for static indoor environments.

In addition, the core of the experimental evaluation process was restricted by time and physical space, along with the limitation of the targeted individuals’ availability and presence. Therefore, the generalization of the results ought to be restricted within complex environments or severe cases, for example. The robot was tested with human interaction, mainly children, which had significantly proven a successful experience and created emotional connection, as they showed empathy and relativeness towards the companion robot. As a result, a successful HRI system took place; however, limited amount of elderly people has interacted with the robot; therefore, the results might not be able to be generalized. To illustrate, legal permission and accessibility issues limited read-world testing with many elderly or autistic people. However, for further development, surveys, Health Center integration, and AI system extension

for further emotional analysis and contextual comprehension might become an aspect for further investigation.

Chapter 12 : Conclusions

Within the context of this multidisciplinary engineering project, the development of an IRIS companion robot stands as an embodiment of mechanical, electrical, and intelligent systems. This chapter shall summarize the entire project, assessing the overall success of the design and implementation process, and describing the three most significant outcomes and obstacles to overcome for future progress and commercialization of the IRIS Platform.

14.1. Summary of Achievements

This project successfully implemented the development and integration of a complete companion robot system, IRIS, as demonstrated in the Figure 128 below intended to assist elderly individuals and users on the autism spectrum. Using a multidisciplinary and multi-faceted approach by using mechanical engineering, electronics, control systems, artificial intelligence, and human-computer interaction, a fully functional prototype was designed, fabricated, and tested.

The mechanical structure and analysis of the robot were completed using Finite Element Analysis (FEA) and concluded that the structure was capable of tolerating an extraordinary amount of load with excellent material properties of PLA. The head, body, arms, and base all remained well within the stress and strain tolerances of PLA when subjected to the loading conditions proposed during testing. The control system was capable of accurately coordinating movement, while the artificial intelligence module created voice, vision, and behavior interactions. The introduction of internet of things (IoT) technologies greatly contributed to monitoring and communications, establishing IRIS as a viable and interactive platform in a companion capacity.

Figure 126 Integrated System of IRIS

14.2. Evaluation of Project Objectives

The goal of developing a safe, modular, and interactive robotic companion was accomplished in full. The mechanical structure was designed and built using PLA with a natural fiber reinforcement which enabled lightweight yet adequately performing mechanical characteristics (verified through comprehensive FEA simulations). The embedded control system could actuate motors and servos, while sensing the environment full-time with sensor fusion.

Artificial intelligence modules, like facial recognition and base emotion recognition, were able to be deployed and tested in real-world applications through computer vision modules.The navigation capabilities, implemented in the Robot Operating System (ROS), enabled basic navigation, obstacle avoidance and localization (using LiDAR, odometry and IMU, with an average accurate positional estimate of approximately 5-7 cm in a structured environment) adequately. The user interface was designed to allow switching between modes, manual control and optional feedback in a simple GUI framework. The IoT system provided notifications for alerts, allowing ongoing environmental updates and maintained real-time awareness and safety

14.3. Future Improvements

There are some potential improvements that can be made to move this work forward. Mechanically with filleting high stress junctions and adding structure ribs may improve the strength without affecting the weight too much. To improve on motion precision and consistency in the arm assembly please consider using some higher torque fully closed-loop

servo. To use more advanced AI models the recommended replacement for the single board computer should be something like the Jetson Orin Nano.

In navigation, using LiDAR and visual SLAM together would provide improved robustness in localization. Battery flexibility and hot-swappable packs would improve operational time. Finally, additional sensors with physiological measurement of an elder person's heart rate, temperature, etc. would increase the robot's capability in eldercare contexts.

In general, it seems that the IRIS companion robot sites achieved an acceptable milestone in that they established the grounds for developing low-cost social assistive dynamic robots, in socially assisted circumstances. They were able to establish a working prototype, by successfully integrating mechanical, electrical, software, and AI subsystems into a multi-functional device. It is an essential project in that it demonstrates that these designs could assist in improving the quality of life for their users.

Furthermore, both the knowledge and experience gained in this project advances the multidisciplinary knowledge needed for the next-generation understanding of HRI. With some improvements to its feasibility and involving users in refining the iteration process, IRIS can become a deployable product for providing companionship, assistance, and monitoring services for an elder person in both a home and clinical environment. Furthermore, both the knowledge and experience gained in this project advances the multidisciplinary knowledge needed for the next-generation understanding of HRI.

References

[1] K. Dautenhahn, S. Woods, C. Kaouri, M. Walters, K. Koay, and I. Werry, *What is a robot companion - Friend, assistant or butler?* 2005, p. 1197. doi: 10.1109/IROS.2005.1545189.

[2] J. A. Rincon, A. Costa, P. Novais, V. Julián, and C. Carrascosa, "A new emotional robot assistant that facilitates human interaction and persuasion," *Knowl. Inf. Syst.*, vol. 60, Jul. 2019, doi: 10.1007/s10115-018-1231-9.

[3] I. Ciuffreda *et al.*, "Design and Development of a Technological Platform Based on a Sensorized Social Robot for Supporting Older Adults and Caregivers: GUARDIAN Ecosystem," *Int. J. Soc. Robot.*, pp. 1–20, Sep. 2023, doi: 10.1007/s12369-023-01038-5.

[4] J. Weizenbaum, "ELIZA—a computer program for the study of natural language communication between man and machine," *Commun. ACM*, vol. 9, no. 1, pp. 36–45, Jan. 1966, doi: 10.1145/365153.365168.

[5] B. Kuipers, E. A. Feigenbaum, P. E. Hart, and N. J. Nilsson, "Shakey: From Conception to History," *AI Mag.*, vol. 38, no. 1, pp. 88–103, Mar. 2017, doi: 10.1609/aimag.v38i1.2716.

[6] S. Sugano and I. Kato, "WABOT-2: Autonomous robot with dexterous finger-arm--Finger-arm coordination control in keyboard performance," in *Proceedings. 1987 IEEE International Conference on Robotics and Automation*, Raleigh, NC, USA: Institute of Electrical and Electronics Engineers, 1987, pp. 90–97. doi: 10.1109/ROBOT.1987.1088025.

[7] M. C. Rose, "The Future Is Furby: Cute-Creepy Encounters with a Zoomorphic Robot," *MC J.*, vol. 26, no. 2, Apr. 2023, doi: 10.5204/mcj.2955.

[8] M. Fujita, "AIBO: Toward the Era of Digital Creatures," *Int. J. Robot. Res.*, vol. 20, no. 10, pp. 781–794, Oct. 2001, doi: 10.1177/02783640122068092.

[9] C. Breazeal, "Emotion and sociable humanoid robots," *Int. J. Hum.-Comput. Stud.*, vol. 59, no. 1–2, pp. 119–155, Jul. 2003, doi: 10.1016/S1071-5819(03)00018-1.

[10] A. V. Libin and E. V. Libin, "Person-robot interactions from the robopsychologists' point of view: the robotic psychology and robotherapy approach," *Proc. IEEE*, vol. 92, no. 11, pp. 1789–1803, Nov. 2004, doi: 10.1109/JPROC.2004.835366.

[11] J. Mizuno *et al.*, "Effect of the Information Support Robot on the Daily Activity of Older People Living Alone in Actual Living Environment," *Int. J. Environ. Res. Public. Health*, vol. 18, no. 5, p. 2498, Mar. 2021, doi: 10.3390/ijerph18052498.

[12] T. Ishida, "Development of a small biped entertainment robot QRIO," in *Micro-Nanomechatronics and Human Science, 2004 and The Fourth Symposium Micro-Nanomechatronics for Information-Based Society, 2004.*, Nagoya, Japan: IEEE, 2004, pp. 1–6. doi: 10.1109/MHS.2004.1421265.

[13] A. Van Breemen, X. Yan, and B. Meerbeek, "iCat: an animated user-interface robot with personality," in *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, The Netherlands: ACM, Jul. 2005, pp. 143–144. doi: 10.1145/1082473.1082823.

[14] A. Tuna and G. Tuna, "The Use of Humanoid Robots with Multilingual Interaction Skills in Teaching a Foreign Language: Opportunities, Research Challenges and Future Research Directions," *Cent. Educ. Policy Stud. J.*, vol. 9, no. 3, pp. 95–115, Sep. 2019, doi: 10.26529/cepsj.679.

[15] Z. A. Barakeh, S. Alkork, A. S. Karar, S. Said, and T. Beyrouthy, "Pepper Humanoid Robot as a Service Robot: a Customer Approach," in *2019 3rd International Conference*

on Bio-engineering for Smart Technologies (BioSMART), Paris, France: IEEE, Apr. 2019, pp. 1–4. doi: 10.1109/BIOSMART.2019.8734250.
[16] B. Goertzel, R. Lian, I. Arel, H. De Garis, and S. Chen, "A world survey of artificial brain projects, Part II: Biologically inspired cognitive architectures," *Neurocomputing*, vol. 74, no. 1–3, pp. 30–49, Dec. 2010, doi: 10.1016/j.neucom.2010.08.012.
[17] K. Wada, T. Shibata, T. Saito, and K. Tanie, "Effects of robot-assisted activity for elderly people and nurses at a day service center," *Proc. IEEE*, vol. 92, no. 11, pp. 1780–1788, Nov. 2004, doi: 10.1109/JPROC.2004.835378.
[18] H. W. Park *et al.*, "Jibo Community Social Robot Research Platform @Scale," in *Companion of the 2024 ACM/IEEE International Conference on Human-Robot Interaction*, Boulder CO USA: ACM, Mar. 2024, pp. 1346–1348. doi: 10.1145/3610978.3638171.
[19] L. Hung, H. Ito, and J. Wong, "LOVOT ROBOT AS COMPANIONS FOR OLDER ADULTS IN LONG-TERM CARE," *Innov. Aging*, vol. 7, no. Supplement_1, pp. 1077–1077, Dec. 2023, doi: 10.1093/geroni/igad104.3460.
[20] E. Broadbent, M. Billinghurst, S. G. Boardman, and P. M. Doraiswamy, "Enhancing social connectedness with companion robots using AI," *Sci. Robot.*, vol. 8, no. 80, p. eadi6347, Jul. 2023, doi: 10.1126/scirobotics.adi6347.
[21] N. L. Robinson, T. V. Cottier, and D. J. Kavanagh, "Psychosocial Health Interventions by Social Robots: Systematic Review of Randomized Controlled Trials," *J. Med. Internet Res.*, vol. 21, no. 5, p. e13203, May 2019, doi: 10.2196/13203.
[22] B. Scassellati, Henny Admoni, and M. Matarić, "Robots for Use in Autism Research," *Annu. Rev. Biomed. Eng.*, vol. 14, no. 1, pp. 275–294, Aug. 2012, doi: 10.1146/annurev-bioeng-071811-150036.
[23] D. J. Ricks and M. B. Colton, "Trends and considerations in robot-assisted autism therapy," in *2010 IEEE International Conference on Robotics and Automation*, Anchorage, AK: IEEE, May 2010, pp. 4354–4359. doi: 10.1109/ROBOT.2010.5509327.
[24] M. R. Banks, L. M. Willoughby, and W. A. Banks, "Animal-Assisted Therapy and Loneliness in Nursing Homes: Use of Robotic versus Living Dogs," *J. Am. Med. Dir. Assoc.*, vol. 9, no. 3, pp. 173–177, Mar. 2008, doi: 10.1016/j.jamda.2007.11.007.
[25] C. Moro, S. Lin, G. Nejat, and A. Mihailidis, "Social Robots and Seniors: A Comparative Study on the Influence of Dynamic Social Features on Human–Robot Interaction," *Int. J. Soc. Robot.*, vol. 11, no. 1, pp. 5–24, Jan. 2019, doi: 10.1007/s12369-018-0488-1.
[26] M. Moore, H. Venkateswara, and S. Panchanathan, "Whistle-blowing ASRs: Evaluating the Need for More Inclusive Speech Recognition Systems," in *Interspeech 2018*, ISCA, Sep. 2018, pp. 466–470. doi: 10.21437/Interspeech.2018-2391.
[27] A. Pande, B. Shrestha, A. Rani, and D. Mishra, "A Comparative Analysis of Real Time Open-Source Speech Recognition Tools for Social Robots," in *Design, User Experience, and Usability*, vol. 14033, A. Marcus, E. Rosenzweig, and M. M. Soares, Eds., in Lecture Notes in Computer Science, vol. 14033. , Cham: Springer Nature Switzerland, 2023, pp. 355–365. doi: 10.1007/978-3-031-35708-4_26.
[28] C. Landesvatter, J. Behnert, and P. C. Bauer, "Comparing Speech-to-Text Algorithms for Transcribing Voice Data from Surveys," Oct. 10, 2023, *SocArXiv*. doi: 10.31235/osf.io/vk6wj.
[29] X. Chen, K. Luo, T. Gee, and M. Nejati, "Does ChatGPT and Whisper Make Humanoid Robots More Relatable?," Feb. 11, 2024, *arXiv*: arXiv:2402.07095. doi: 10.48550/arXiv.2402.07095.
[30] A. M. Fuad, S. J. Ahmed, N. J. Anannya, M. F. Mridha, and K. Nur, "An Open-Source Voice Command-Based Human-Computer Interaction System Using Speech Recognition Platforms," in *Proceedings of the 2nd International Conference on Big Data, IoT and*

Machine Learning, vol. 867, M. S. Arefin, M. S. Kaiser, T. Bhuiyan, N. Dey, and M. Mahmud, Eds., in Lecture Notes in Networks and Systems, vol. 867. , Singapore: Springer Nature Singapore, 2024, pp. 527–545. doi: 10.1007/978-981-99-8937-9_36.

[31] S. N. Akter *et al.*, "An In-depth Look at Gemini's Language Abilities," Dec. 24, 2023, *arXiv*: arXiv:2312.11444. doi: 10.48550/arXiv.2312.11444.

[32] G. Rossettini *et al.*, "Comparative accuracy of ChatGPT-4, Microsoft Copilot and Google Gemini in the Italian entrance test for healthcare sciences degrees: a cross-sectional study," *BMC Med. Educ.*, vol. 24, no. 1, p. 694, Jun. 2024, doi: 10.1186/s12909-024-05630-9.

[33] F. H. Fattah *et al.*, "Comparative analysis of ChatGPT and Gemini (Bard) in medical inquiry: a scoping review," *Front. Digit. Health*, vol. 7, p. 1482712, Feb. 2025, doi: 10.3389/fdgth.2025.1482712.

[34] C. Kennington, D. Moro, L. Marchand, J. Carns, and D. McNeill, "rrSDS: Towards a Robot-ready Spoken Dialogue System," in *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, O. Pietquin, S. Muresan, V. Chen, C. Kennington, D. Vandyke, N. Dethlefs, K. Inoue, E. Ekstedt, and S. Ultes, Eds., 1st virtual meeting: Association for Computational Linguistics, Jul. 2020, pp. 132–135. doi: 10.18653/v1/2020.sigdial-1.17.

[35] S. Srinivasan, "Misty-A Development Platform for Socially Assistive Robots [Student's Corner]," *IEEE Robot. Autom. Mag.*, vol. 26, no. 2, pp. 103–105, Jun. 2019, doi: 10.1109/MRA.2019.2910420.

[36] F. O. Zhao, N. White, B. Cagiltay, P. Niedenthal, J. Michaelis, and B. Mutlu, *Designing Emotional Expressions for a Reading Companion Robot*. 2023. doi: 10.31234/osf.io/7p2ns.

[37] Z. Liu *et al.*, "A facial expression emotion recognition based human-robot interaction system," *IEEECAA J. Autom. Sin.*, vol. 4, no. 4, pp. 668–676, 2017, doi: 10.1109/JAS.2017.7510622.

[38] N. Churamani, M. Kerzel, E. Strahl, P. Barros, and S. Wermter, *Teaching Emotion Expressions to a Human Companion Robot using Deep Neural Architectures*. 2017. doi: 10.1109/IJCNN.2017.7965911.

[39] N. T. White, B. Cagiltay, J. E. Michaelis, and B. Mutlu, "Designing Emotionally Expressive Social Commentary to Facilitate Child-Robot Interaction," in *Proceedings of the 20th Annual ACM Interaction Design and Children Conference*, in IDC '21. New York, NY, USA: Association for Computing Machinery, Jun. 2021, pp. 314–325. doi: 10.1145/3459990.3460714.

[40] A. K. Pandey and R. Gelin, "A Mass-Produced Sociable Humanoid Robot: Pepper: The First Machine of Its Kind," *IEEE Robot. Autom. Mag.*, vol. PP, pp. 1–1, Jul. 2018, doi: 10.1109/MRA.2018.2833157.

[41] S. Chew, W. Tay, D. Smit, and C. Bartneck, "Do Social Robots Walk or Roll?," in *Social Robotics*, vol. 6414, S. S. Ge, H. Li, J.-J. Cabibihan, and Y. K. Tan, Eds., in Lecture Notes in Computer Science, vol. 6414. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 355–361. doi: 10.1007/978-3-642-17248-9_37.

[42] J. R. Silva, M. Simao, N. Mendes, and P. Neto, "Navigation and obstacle avoidance: a case study using Pepper robot," in *IECON 2019 - 45th Annual Conference of the IEEE Industrial Electronics Society*, Lisbon, Portugal: IEEE, Oct. 2019, pp. 5263–5268. doi: 10.1109/IECON.2019.8927009.

[43] A. Vega, R. Cintas, L. J. Manso, P. Bustos, and P. Núñez, "Socially-Accepted Path Planning for Robot Navigation Based on Social Interaction Spaces," in *Robot 2019: Fourth Iberian Robotics Conference*, vol. 1093, M. F. Silva, J. Luís Lima, L. P. Reis, A. Sanfeliu, and D. Tardioli, Eds., in Advances in Intelligent Systems and Computing, vol.

1093. , Cham: Springer International Publishing, 2020, pp. 644–655. doi: 10.1007/978-3-030-36150-1_53.
[44] S.-H. Chan, P.-T. Wu, and L.-C. Fu, "Robust 2D Indoor Localization Through Laser SLAM and Visual SLAM Fusion," in *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Miyazaki, Japan: IEEE, Oct. 2018, pp. 1263–1268. doi: 10.1109/SMC.2018.00221.
[45] W. Yuan, Z. Li, and C.-Y. Su, "RGB-D sensor-based visual SLAM for localization and navigation of indoor mobile robot," in *2016 International Conference on Advanced Robotics and Mechatronics (ICARM)*, Macau, China: IEEE, Aug. 2016, pp. 82–87. doi: 10.1109/ICARM.2016.7606899.
[46] J. Guo, X. Xiao, P. Pan, and X. Luo, "A design of multi-vision localization and navigation service robot system," in *2017 12th International Conference on Computer Science and Education (ICCSE)*, Houston, TX, USA: IEEE, Aug. 2017, pp. 787–790. doi: 10.1109/ICCSE.2017.8085601.
[47] H.-M. Gross *et al.*, "Robot companion for domestic health assistance: Implementation, test and case study under everyday conditions in private apartments," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Hamburg, Germany: IEEE, Sep. 2015, pp. 5992–5999. doi: 10.1109/IROS.2015.7354230.
[48] P.-T. Wu, C.-A. Yu, S.-H. Chan, M.-L. Chiang, and L.-C. Fu, "Multi-Layer Environmental Affordance Map for Robust Indoor Localization, Event Detection and Social Friendly Navigation," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Macau, China: IEEE, Nov. 2019, pp. 2945–2950. doi: 10.1109/IROS40897.2019.8968455.
[49] M. Luber, L. Spinello, J. Silva, and K. O. Arras, "Socially-aware robot navigation: A learning approach," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vilamoura-Algarve, Portugal: IEEE, Oct. 2012, pp. 902–907. doi: 10.1109/IROS.2012.6385716.
[50] H. Kretzschmar, M. Spies, C. Sprunk, and W. Burgard, "Socially compliant mobile robot navigation via inverse reinforcement learning," *Int. J. Robot. Res.*, vol. 35, no. 11, pp. 1289–1307, Sep. 2016, doi: 10.1177/0278364915619772.
[51] M. Luber, L. Spinello, J. Silva, and K. O. Arras, "Socially-aware robot navigation: A learning approach," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vilamoura-Algarve, Portugal: IEEE, Oct. 2012, pp. 902–907. doi: 10.1109/IROS.2012.6385716.
[52] D. Helbing and P. Molnár, "Social force model for pedestrian dynamics," *Phys. Rev. E*, vol. 51, no. 5, pp. 4282–4286, May 1995, doi: 10.1103/PhysRevE.51.4282.
[53] G. Ferrer, A. Garrell, and A. Sanfeliu, "Robot companion: A social-force based approach with human awareness-navigation in crowded environments," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Tokyo: IEEE, Nov. 2013, pp. 1688–1694. doi: 10.1109/IROS.2013.6696576.
[54] A. Vega, L. J. Manso, D. G. Macharet, P. Bustos, and P. Núñez, "Socially aware robot navigation system in human-populated and interactive environments based on an adaptive spatial density function and space affordances," *Pattern Recognit. Lett.*, vol. 118, pp. 72–84, Feb. 2019, doi: 10.1016/j.patrec.2018.07.015.
[55] A. Sharkey and N. Sharkey, "Granny and the robots: ethical issues in robot care for the elderly," *Ethics Inf. Technol.*, vol. 14, no. 1, pp. 27–40, Mar. 2012, doi: 10.1007/s10676-010-9234-6.
[56] H. Drukarch, C. Calleja, and E. Fosch-Villaronga, "An iterative regulatory process for robot governance," *Data Policy*, vol. 5, p. e8, 2023, doi: 10.1017/dap.2023.3.

[57] “3. Odometry · linorobot/linorobot Wiki.” Accessed: Jun. 10, 2025. [Online]. Available: https://github.com/linorobot/linorobot/wiki/3.-Odometry

Appendix A - GUI Application

This Python code snippet displays the initialization and layout part of the GUI build using Tkinter, pygame, and subprocess, which is the front-end interface for accessing the robot's functions such as settings, games, navigation, and saved maps.

```
“

from pathlib import Path

from tkinter import Tk, Canvas, PhotoImage, Button, Frame

import pygame  # To play the sound

import subprocess


# Initialize pygame mixer for button sound feedback

pygame.mixer.init()


# Asset paths

OUTPUT_PATH = Path(__file__).parent

ASSETS_FRAME0 = OUTPUT_PATH / Path(r"C:\Users\Asus\Desktop\GUI\buildTemp\assets\frame0")


def relative_to_assets(path: str, base_path: Path) -> Path:

    return base_path / Path(path)


class App(Tk):

    def __init__(self):

        super().__init__()
```

```
        self.geometry("1024x600")
        self.configure(bg="#151A21")
        self.resizable(False, False)

        self.current_theme = "dark"
        subprocess.Popen(["python", "chatbot.py"], shell=True)

        # Initialize GUI frames
        self.frame0 = Frame0(self)
        self.frame1 = Frame1(self)
        self.frame0.place(x=0, y=0, relwidth=1, relheight=1)

class Frame0(Frame):
    def __init__(self, master):
        super().__init__(master, bg="#151A21")
        self.canvas = Canvas(self, bg="#151A21", height=600, width=1024, bd=0, highlightthickness=0)
        self.canvas.place(x=0, y=0)

        # Load and configure menu button
        self.button_image = PhotoImage(file=relative_to_assets("button_1.png", ASSETS_FRAME0))
        self.button_hover = PhotoImage(file=relative_to_assets("button_1_hover.png", ASSETS_FRAME0))
        self.menu_button = Button(
```

```
            self,
            image=self.button_image,
            borderwidth=0,
            highlightthickness=0,
            bg="#151A21",
            activebackground="#151A21",
            relief="flat",
            command=self.master.show_frame1
        )
        self.menu_button.place(x=20, y=490, width=90, height=90)
        self.menu_button.bind("<Enter>", lambda e: self.menu_button.config(image=self.button_hover))
        self.menu_button.bind("<Leave>", lambda e: self.menu_button.config(image=self.button_image))
        self.menu_button.bind("<Button-1>", self.play_pop_sound)

    def play_pop_sound(self, event):
        pygame.mixer.music.load("C:/Users/Asus/Downloads/pop_sound.wav")
        pygame.mixer.music.play()

“
```

Appendix B - IoT Sensor and ThingsBoard Integration Code

The Arduino sketch below is designed to read temperature, humidity, flame detection and gas sensor data from the IRIS robot's sensor package on and transmit that information securely via MQTT to the ThingsBoard IoT cloud platform via WiFi with SSL/TLS.

“

```
#include <ESP8266WiFi.h>
#include <DHTesp.h>
#include <ThingsBoard.h>
#include <Arduino_MQTT_Client.h>
#include <WiFiClientSecure.h>

// Pin for DHT sensor
#define pinDht 4  // GPIO4 (D2 on NodeMCU)

// Pin for flame sensor (digital output)
#define pinFlame 0  // GPIO14 (D3 on NodeMCU)

// DHT Sensor object
DHTesp dhtSensor;

// WiFi credentials
#define WIFI_AP "Pizzanetwork"
#define WIFI_PASS "Digital10"

// ThingsBoard server and device token
#define TB_SERVER "eu.thingsboard.cloud"
#define TOKEN "vsCGpL7nZRA55581e2RN"

// MQTT configuration
constexpr uint16_t MAX_MESSAGE_SIZE = 128U;

// Clients
WiFiClientSecure espClient;  // Secure connection
Arduino_MQTT_Client mqttClient(espClient);
ThingsBoard tb(mqttClient, MAX_MESSAGE_SIZE);

// Function to connect to WiFi
void connectToWiFi() {
  if (WiFi.status() != WL_CONNECTED) {
```

```
    Serial.print("Connecting to WiFi...");
    WiFi.begin(WIFI_AP, WIFI_PASS);
    int attempts = 0;
    while (WiFi.status() != WL_CONNECTED && attempts < 20) {
      delay(500);
      Serial.print(".");
      attempts++;
    }
    if (WiFi.status() == WL_CONNECTED) {
      Serial.println("\nConnected to WiFi");
    } else {
      Serial.println("\nFailed to connect to WiFi");
    }
  }
}

// Function to connect to ThingsBoard
void connectToThingsBoard() {
  if (!tb.connected()) {
    Serial.println("Connecting to ThingsBoard server...");

    // For testing, skip certificate validation (⚠️ use with caution!)
    espClient.setInsecure();

    if (tb.connect(TB_SERVER, TOKEN, 8883)) {
      Serial.println("Connected to ThingsBoard");
    } else {
      Serial.println("Failed to connect to ThingsBoard");
    }
  }
}

// Function to send telemetry data (temperature, humidity, MQ sensor, flame
sensor)
void sendDataToThingsBoard(float temp, int hum, int mqValue, int
flameDetected) {
  bool success = true;

  if (!tb.sendTelemetryData("temperature", temp)) {
    Serial.println("Failed to send temperature");
    success = false;
  }

  if (!tb.sendTelemetryData("humidity", hum)) {
    Serial.println("Failed to send humidity");
    success = false;
  }
```

```
  if (!tb.sendTelemetryData("mq_sensor", mqValue)) {
    Serial.println("Failed to send MQ sensor value");
    success = false;
  }

  if (!tb.sendTelemetryData("flame_detected", flameDetected)) {
    Serial.println("Failed to send flame sensor value");
    success = false;
  }

  if (success) {
    Serial.println("Data sent successfully");
  }
}

void setup() {
  Serial.begin(9600);
  dhtSensor.setup(pinDht, DHTesp::DHT11);
  pinMode(pinFlame, INPUT);  // Setup flame sensor pin as input
  connectToWiFi();
  connectToThingsBoard();
}

void loop() {
  connectToWiFi();

  // Read DHT sensor
  TempAndHumidity data = dhtSensor.getTempAndHumidity();
  float temp = data.temperature;
  int hum = data.humidity;

  // Read MQ sensor (connected to A0 through voltage divider)
  int mqValue = analogRead(A0);

  // Read flame sensor (digital output)
  int flameDetected = digitalRead(pinFlame) == LOW ? 1 : 0;


  // Print readings
  Serial.print("Temperature: ");
  Serial.println(temp);
  Serial.print("Humidity: ");
  Serial.println(hum);
  Serial.print("MQ Sensor Value: ");
  Serial.println(mqValue);
  Serial.print("Flame Detected: ");
  Serial.println(flameDetected);

  if (!tb.connected()) {
```

```
    connectToThingsBoard();
  }

  // Send all data
  sendDataToThingsBoard(temp, hum, mqValue, flameDetected);

  tb.loop();
  delay(3000);
}
```

“