People‘s Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
University of Mohamed Khider - Biskra
Faculty of Exact Sciences and Sciences of Nature and Life
Computer Science Department

N° d’ordre : IA_Start Up_02/M2/2024

Thesis

Submitted in fulfillment of the requirements for the Master’s degree in

Computer Science

Speciality: Artificial Intelligence

Integrating Conversational AI into a Physical Autonomy for Medical Applications

---

Presented By:

Djellab Mohamed Ayhem

Meriem Bekhoucha

Samar Hibat Errahmane Guessoum

Graduating on June 23, 2024, before the jury composed of:

Chighoub Rabia	MCB	Examiner
Youkana Imene	MCA	President
Zouai Meftah	MCA	Supervisor
Aloui Ahmed	MCB	Co-Supervisor

Academic Year 2023/2024

Acknowledment

First and foremost, we are immensely grateful to Allah, the Most Gracious, the Most Merciful, for His blessings, guidance, and unwavering support throughout this journey.

We are profoundly grateful to ***Dr. Meftah Zouai*** and ***Dr. Aloui Ahmed*** for their exceptional mentorship, invaluable insights, and unwavering support. Their expertise, patience, and dedication to our success have played a crucial role in shaping this thesis.

We are profoundly grateful to our family and friends for their unwavering support, prayers, and motivation. Their belief in our abilities and constant encouragement have been a driving force behind our academic achievements.

Dedication

This work is dedicated to:

Our dear family,

Our esteemed teachers in the Computer Science Department.

Our friends and classmates.

Members of the Biskra Computer Science Club.

And to everyone who has supported us throughout Our lives.

Special dedication to our precious parents and siblings.

ملخص

تشهد الرعاية الصحية ثورة روبوتية حيث تقدم الروبوتات المساعدة دعمًا متعدد الأوجه. بالنسبة للمرضى، فهي توفر الراحة العاطفية، والتذكير بالأدوية، والمساعدة في المهام اليومية، خاصة لأولئك المعزولين أو الذين يحتاجون إلى يد المساعدة. تعمل الروبوتات الجراحية على توسيع مهارات الجراح، مما يتيح إجراء عمليات جراحية طفيفة التوغل مع أوقات تعافي أسرع. وإلى جانب المرضى، تتعامل الروبوتات مع الأعباء اللوجستية، وإدارة الإمدادات، وتطهير الأسطح، وتحرير الموظفين للرعاية المعقدة. تعمل هذه القفزة التكنولوجية على تحسين نتائج المرضى وتبسيط عمليات المستشفى .

يهدف مشروعنا إلى إحداث ثورة في رعاية المرضى من خلال توفير الرفقة والمساعدة العملية. ومن خلال أتمتة المهام الروتينية مثل توزيع الأدوية وجلب الإمدادات، سيوفر الروبوت الممرضات والأطباء لتفاعلات أكثر أهمية. بالإضافة إلى ذلك، يمكن لميزاته تعزيز وعي المريض من خلال الإجابة على الأسئلة الطبية. يتمتع هذا الروبوت بالقدرة على تحسين الكفاءة ورعاية المرضى بشكل.

الكلمات المفتاحية : الذكاء الاصطناعي للمحادثة، الذكاء الاصطناعي، المساعدة الطبية، الروبوتات، إنترنت الأشياء، معالجة اللغات الطبيعية، الرعاية الصحية، الشابوت، التعلم المعزز.

Résume

La santé connaît une révolution robotique avec des robots assistants offrant un accompagnement multiforme. Pour les patients, ils apportent un réconfort émotionnel, des rappels de médicaments et une aide aux tâches quotidiennes, notamment pour les personnes isolées ou ayant besoin d'un coup de main. Les robots chirurgicaux étendent les compétences des chirurgiens, permettant des procédures peu invasives avec des temps de récupération plus rapides. Au-delà des patients, les robots s'attaquent aux tâches logistiques, gèrent les fournitures, désinfectent les surfaces et libèrent le personnel pour des soins plus complexes. Cette avancée technologique améliore les résultats pour les patients et rationalise les opérations hospitalières.

Notre projet vise à révolutionner les soins aux patients en fournissant un accompagnement et une assistance pratiques. En automatisant les tâches de routine telles que la distribution de médicaments et la récupération de fournitures, le robot libérera les infirmières et les médecins pour des interactions plus importantes. De plus, ses fonctionnalités peuvent sensibiliser les patients en répondant à des questions médicales. Ce robot a le potentiel d'améliorer considérablement l'efficacité et les soins aux patients au sein des établissements de santé.

Mote clé : Intelligence artificielle conversationnelle, Intelligence artificielle, Assistance médicale, Robotique, Internet des objets, Traitement du langage naturel,

Santé, Chabot, Apprentissage par renforcement.

Abstract

Healthcare is experiencing a robotic revolution, with assistant robots offering multi-faceted support. For patients, they provide emotional comfort, medication reminders, and daily task assistance, especially for those who are isolated or need a helping hand. Surgical robots extend surgeon skills, enabling minimally invasive procedures with faster recovery times. Beyond patient care, robots tackle logistical burdens, managing supplies, disinfecting surfaces, and freeing up staff for more complex tasks. This technological leap is improving patient outcomes and streamlining hospital operations.

Our project aims to revolutionize patient care by providing companionship and practical assistance. By automating routine tasks such as dispensing medications and fetching supplies, these robots will free up nurses and doctors for more important interactions. In addition, its features can enhance patient awareness by answering medical questions. This robot has the potential to significantly improve efficiency and patient care within healthcare facilities.

keywords : Conversational Artificial Intelligence, Artificial Intelligence, Medical Assistance, Robotics, Internet of things, Natural language processing, Healthcare, Chabot, Reinforcement learning.

Contents

List of Figures

General Introduction

Context

In recent years, the field of medical assistance has witnessed remarkable developments greatly influenced by the emergence of the IoT. This technological revolution has enabled the integration of a vast network of interconnected devices and systems into healthcare, ranging from telemedicine platforms and wearable health monitoring devices to AI-based diagnostic tools and robotic surgical systems. By integrating the IoT into healthcare practices, we can create intelligent, efficient, and interconnected systems that dramatically improve patient care, streamline hospital operations, and improve health outcomes.

One notable development is the integration of conversational AI into bodily autonomy for medical applications. This innovative approach combines advanced AI algorithms and autonomous robotic systems to provide smart, efficient, and interconnected healthcare solutions.

Problematic and Motivation

Throughout medical history, providing effective and timely medical assistance has been complex and demanding. Healthcare professionals have faced challenges in delivering personalized care, early diagnosis, and precise treatments due to limitations in existing technologies and practices. As the field of medicine continues to advance and the global population increases, there is a growing need for innovative solutions that can enhance the efficiency and effectiveness of medical assistance.

Advancements in technologies such as the IoT, AI, Robotics, and Biomedical Engineering are driving forces behind the development of these solutions, offering the potential to revolutionize the healthcare industry and improve the quality of life for patients worldwide.

Objectives

The primary goal of this project is to develop and implement an intelligent robotic system that integrates conversational artificial intelligence with physical autonomy to provide medical assistance in various environments, including interacting with patients, navigating independently within medical facilities by mapping the surrounding areas and locating itself on the map, and creating its automated system. Additionally using artificial neural network model, equipped to make decisions independently. The system will contain natural language processing for patient interaction, autonomous navigation for navigation, and real-time data analysis to support medical staff. By the end of the project, the automated system is expected to improve the efficiency of patient care, reduce the workload of healthcare professionals, and enhance overall healthcare delivery.

Following the introductory section, the thesis will be structured into four chapters as outlined below:

Chapitre 01: Fundamental Concepts

This chapter begins the discussion by exploring the IoT, presenting its key components, architectural principles, and protocols, and discussing its advantages and challenges. Furthermore, it delves into the diverse application areas of IoT.

Chapitre 02: Overview of Robotics.

This chapter provides a comprehensive overview of robotics, elucidating its foundational concepts, navigation methodologies, and the classification of robots, with a particular focus on autonomous mobile robots akin to our robot.

Chapitre 03: System Design

This chapter entails a detailed description of the system under development, elucidating the functionality of each component.

Chapitre 04: System Implementation.

This chapter details the tools and methodologies employed for system implementation, providing code specifics, and discussing the anticipated outcomes of the project.

Chapter I

Fundamental Concepts

I.1 Introduction

The Artificial Intelligence of Things (AIOT) in healthcare combines Artificial Intelligence (AI with IoT devices to transform patient care -by collecting and analyzing real-time data- from wearable sensors and smart medical equipment to computers or centralized systems. This integration enables remote monitoring, early disease detection, and optimized treatments. Despite challenges like data security, interoperability, and regulatory compliance, AIOT promises to enhance personalized medicine, telehealth, and overall patient care with continued advancements.

In this chapter, we will address some basic terms in general and the Convergence of AI and IoT in Healthcare in particular.

I.2 Artificial Intelligence

Artificial Intelligence (AI) is transforming industries and everyday life, bringing with it both opportunities and challenges. This section provides an overview of AI, starting with its definition and reviewing its basic concepts and others [45].

I.2.1 Definition

AI can be defined as an area of study in the field of computer science concerning the development of computers able to engage in human lines through processes such as learning, reasoning, and self-correction. Moreover, AI is the study of techniques to use computers more effectively by improved programming techniques[5].

I.2.2 Historical Background

Understanding the history of AI helps us appreciate its current state and future potential. The development of AI can be traced back to many key milestones, including:

1940s-1950s: Foundations

- **Alan Turing :** Proposed the Turing Test for machine intelligence[12].

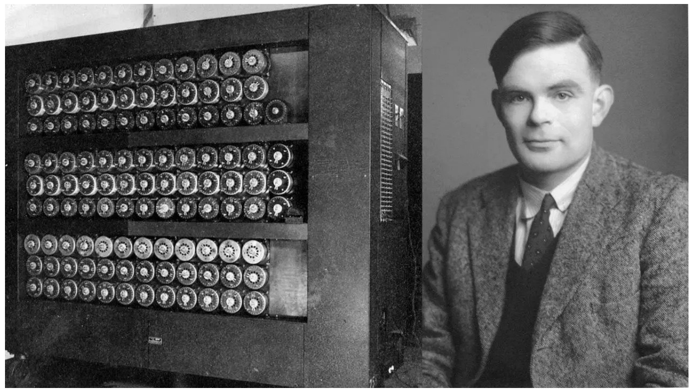

Figure I.1: Alan Turing & The Turing Machine.

- **Dartmouth Conference (1956) :** Birth of AI as a field, the term "Artificial Intelligence" was coined. This conference brought together prominent mathematicians and scientists to explore the possibility of creating machines that could simulate human intelligence[12].

- **Logic Theorist (1955):** This thinking machine was a program called the Logic Theorist (Newell and Simon). Its thinking consisted of creating proofs for theorems in propositional logic[22].

Figure I.2: The 1956 Dartmouth Conference

1960s-1970s: Early Developments

- **ELIZA (1966):** Early chatbot mimicking a psychotherapist, developed by Joseph Weizenbaum, It is arguably among the most influential computer programs ever written [6].

- **Backpropagation (1974):** Paul Werbos introduced the concept of backpropagation for training neural networks, which later became crucial in the development of deep learning [72].

Figure I.3: AI pioneers Allen Newell (right) and Herbert Simon.

1980s: AI Winter

In 1986, **expert systems** were a breakthrough in the AI field. By mimicking the decision-making process made by humans and using a knowledge base of rules, researchers were hopeful that this invention was about to change the world and solve many issues in different fields. However, after some time it failed to fulfill every task it was given, many AI companies went bankrupt, and research funding decreased because of all the criticism AI was suffering from and its actual value that was under question [11].

1990s:Machine Learning (ML) and Achievements

The world was able to get over this hurdle after more innovations and advancements appeared in this field, especially when it comes to ML and Neural Networks. One of the most notable successes was in mid-1990 when IBM developed a chess-playing machine called Deep Blue. It played with Garry Kasparov in 1996 and lost four games to two, then upgraded in 1997 and defeated Kasparov.[8]

2000s-Present: AI Renaissance

- **Big Data and Deep Learning:** Major advancements in AI capabilities. The spread of social media around the world and the advent of the Internet of Things in the 2000s marked a significant shift. Organizations began collecting massive amounts of data, alongside the development of deep learning algorithms, AI systems were able to learn complex patterns and improve their performance over time

- **AlphaGo (2016):** One of the landmark achievements during the AI Renaissance was DeepMind's AlphaGo system defeating the world-champion Go player, Lee Sedol, in a five-game match.[63]

  Over time AI has been integrated into devices that we use daily like voice assistants and autonomous vehicles, this integration has made AI an essential part of our everyday lives.

I.2.3 Key Technologies

AI includes a variety of technologies that contribute to enhancing its capabilities. These technologies are considered essential for the development and implementation of AI systems, including:

I.2.3.1 Expert Systems

An expert system is a computer program that uses artificial intelligence (AI) techniques to simulate the decision-making ability and problem-solving skills of a human expert in a specific domain. There are two types of an expert system:

- **Rule-Based Systems:** Use predefined rules to make decisions. Examples include medical diagnosis systems like MYCIN.
- **Knowledge-Based Systems:** Incorporate human expertise to solve specific problems.

I.2.3.2 Computer Vision

Computer vision is a field that enables computers to interpret data (photos, videos) into valuable information using algorithms and techniques.[42]

- **Image Recognition:** Identifying objects, people, or scenes in images. Examples include facial recognition and object detection.
- **Video Analysis:** Understanding and interpreting video content. Examples include activity recognition and video summarization.

I.2.3.3 Machine Learning (ML)

Machine Learning is the backbone of AI, focusing on developing algorithms that allow computers to learn from, and make predictions based on data. It includes:

- **Supervised Learning:** Algorithms that learn from labeled data to make predictions. Examples include regression and classification.
- **Unsupervised Learning:** Algorithms find patterns in unlabeled data. Examples include clustering and dimensionality reduction.

  In Figure I.4, **a schematic** represents an unsupervised learning model, where the unlabelled data is grouped into different clusters depending on specific features. **b Schematic** represents a supervised learning model, which uses labeled data for training and classifies any new data into different classes.

- **Reinforcement Learning (RL):** Algorithms learn by interacting with an environment and receiving rewards or penalties. Examples include Q-learning and Proximal Policy Optimization (PPO).

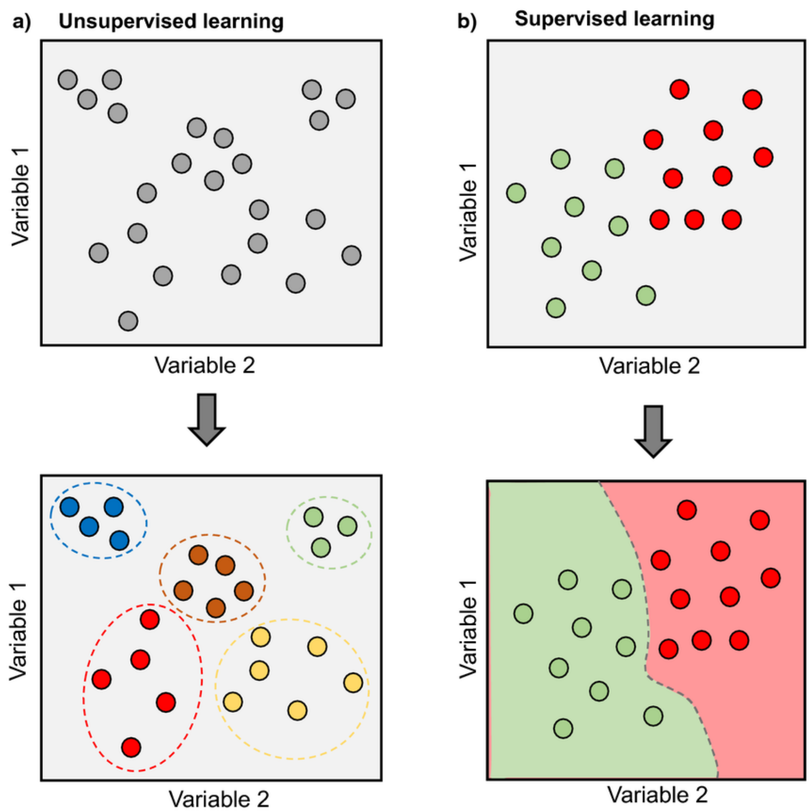

Figure I.4: Supervised and unsupervised machine learning

I.2.3.4 Deep Learning

Deep learning is a specialized subset of machine learning methods that utilizes artificial neural networks with multiple hidden layers to learn and make inferences from complex data [58].

- **Neural Networks:** Composed of layers of interconnected nodes (neurons) that process data in complex ways.
- **Convolutional Neural Networks (CNNs):** Specialized for image and video processing.
- **Recurrent Neural Networks (RNNs):** Specialized for sequential data, like time series or natural language.

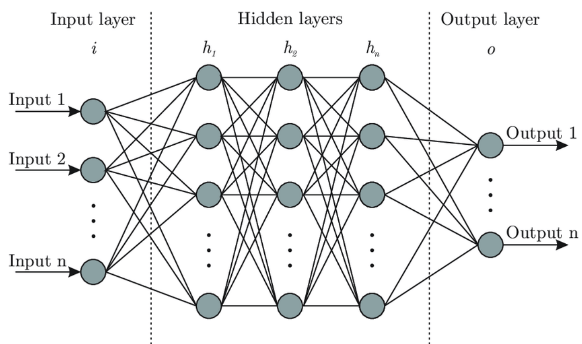

Figure I.5: Artificial neural network architecture

I.2.3.5 Natural Language Processing (NLP)

NLP is a branch of AI that enables computers to understand, interpret, and generate human language. Some of its applications:

- **Text Analysis:** Techniques for understanding and processing human language.
- **Machine Translation:** Automatic translation of text between languages. Examples include Google Translate.
- **Chatbots and Virtual Assistants:** Systems like Siri, Alexa, and GPT-3 can interact with users in natural language.

I.2.3.6 Robotics

Robotics is the field that designs and builds robots, combining electronics, mechanics, and computer science. Robots perform tasks in industries from manufacturing to exploration, including:

- **Autonomous Vehicles:** Self-driving cars that navigate using AI.
- **Industrial Robots:** Robots used in manufacturing for tasks like assembly and quality control.
- **Service Robots:** Robots that provide services, such as cleaning or delivery.

I.2.4 Applications in General

The advancement of AI that keeps pushing the boundaries of what machines can do has led to a wide range of applications in different domains. Here are some examples [69]:

Finance

Fraud Detection: AI systems analyze transaction patterns to identify and prevent fraudulent activities.

Customer Service: AI-powered chatbots handle customer inquiries, providing instant responses and support.

Retail

Recommendation Systems: AI recommends products to customers based on browsing and purchasing history, enhancing the shopping experience.

Customer Insights: AI analyzes customer data to identify trends and preferences, helping businesses tailor their marketing strategies.

Manufacturing

Predictive Maintenance: AI monitors equipment performance and predicts maintenance needs, reducing downtime and repair costs.

Quality Control: AI systems inspect products for defects, ensuring high quality and consistency.

Supply Chain Optimization: AI optimizes logistics and supply chain processes, improving efficiency and reducing costs.

Transportation

Autonomous Vehicles: AI powers self-driving cars and trucks, enhancing road safety and reducing traffic congestion.

Traffic Management: AI analyzes traffic patterns and optimizes traffic flow, reducing travel time and emissions.

Predictive Maintenance: AI predicts maintenance needs for vehicles and infrastructure, preventing breakdowns and improving safety.

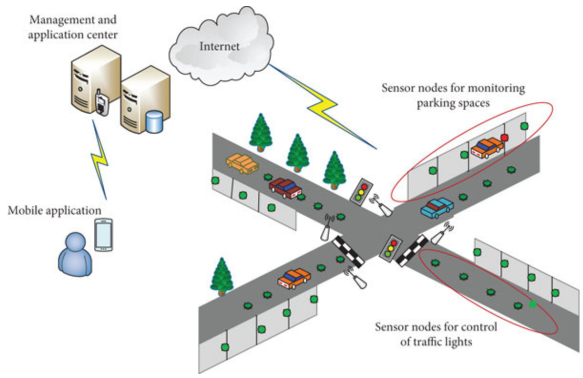

Figure I.6: Intelligent traffic control system

Education

Personalized Learning: AI provides customized learning experiences based on individual student needs and progress.

Administrative Tasks: AI automates administrative tasks such as grading and scheduling, freeing up time for educators.

Learning Analytics: AI analyzes student data to identify at-risk students and provide targeted interventions.

Agriculture

Precision Farming: AI optimizes crop management by analyzing soil health, weather patterns, and crop performance.

Automated Machinery: AI-powered robots and drones assist with planting, harvesting, and monitoring crops.

Disease Detection: AI detects plant diseases and pest infestations early, enabling timely interventions.

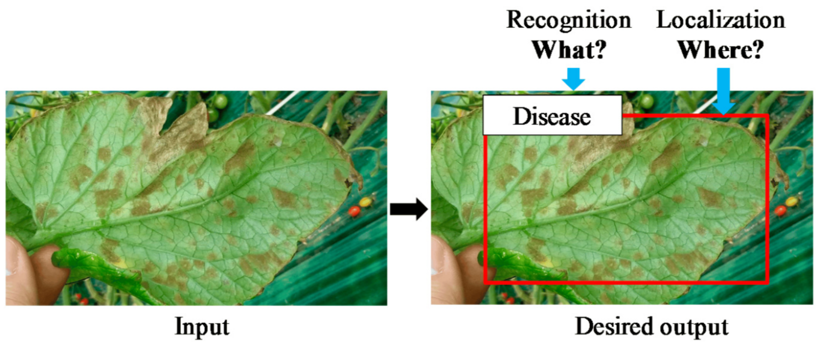

Figure I.7: Recognition and localization of plant diseases

Healthcare

Healthcare is an area where AI is making significant strides, with applications that are transforming the way medical professionals diagnose and treat patients.

Medical Imaging: AI algorithms can analyze medical images to detect abnormalities, such as tumors, more accurately and quickly than human radiologists.

Predictive Analytics: AI models predict patient outcomes, disease outbreaks, and treatment success, aiding in proactive healthcare management.

Personalized Medicine: AI helps tailor treatments to individual patients based on their genetic makeup and health history.

I.2.5 Challenges and Opportunities in AI

As AI continues to evolve, it brings with it both significant challenges and exciting opportunities. Addressing these challenges is crucial for the responsible and beneficial deployment of AI technologies.

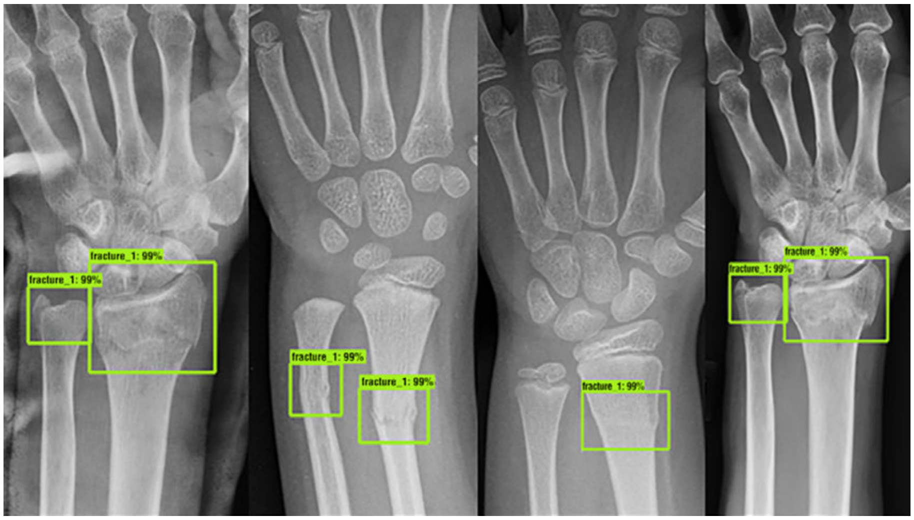

Figure I.8: Fracture detection using AI

I.2.5.1 Challenges in AI

Data Privacy and Security

Ensuring data privacy and security is a significant challenge, especially with the increasing amount of data collected and processed. Breaches can lead to severe consequences, including identity theft and loss of trust[14].

Bias and Fairness

AI systems can inherit biases from the data they are trained on, leading to unfair and discriminatory outcomes. Ensuring fairness and reducing bias is crucial but challenging due to the complexity of human biases and diverse data sources[17].

Interpretability and Transparency

Many AI models, especially deep learning models, are often seen as "black boxes" due to their complexity. Understanding how these models make decisions is crucial for trust and accountability but remains a significant challenge[41].

Ethical Considerations

The ethical implications of AI applications, such as surveillance, autonomous weapons, and job displacement, require careful consideration and regulation to prevent misuse and ensure societal benefit.

Financial Services

AI can enhance fraud detection, automated trading, and provide personalized financial advice. AI-powered algorithms can analyze vast amounts of data to identify patterns and anomalies, enabling more effective fraud detection and prevention. This can lead to more secure, efficient, and customer-centric financial services.

Manufacturing and Industry

AI-driven automation and predictive maintenance can improve quality control in manufacturing. And by automating repetitive tasks and optimizing production workflows, AI can increase productivity and throughput in industrial settings [36]. This can lead to more efficient and cost-effective industrial processes.

Agriculture

AI can optimize farming practices through precision agriculture, which uses data and AI to manage crops and livestock more efficiently. This can lead to higher yields, reduced waste, and more sustainable farming practices. AI can help farmers reduce their environmental impact by optimizing irrigation, minimizing pesticide use, and improving waste management[25].

Urban Planning and Smart Cities

AI can help design and manage smart cities by optimizing resource usage, improving public services, and enhancing the quality of life for residents. This includes intelligent waste management, energy distribution, Green Urban Planning, and Smart Mobility[62].

I.2.5.2 Public Health Systems

Public health includes a range of initiatives and programs aimed at promoting and protecting the health and well-being of populations. Key components of public health systems include:

- **Universal Health Coverage (UHC):** Many countries are working towards UHC to ensure all individuals have access to essential healthcare services without financial hardship.
- **Preventive Care:** Increased emphasis on preventive measures, vaccinations, and public health campaigns to reduce the incidence of diseases.

I.2.6 Technological Innovations

AI is driving technological innovations across various industries, transforming how businesses operate and how people interact with technology.

I.2.6.1 Telemedicine

Telemedicine combines the fields of communications technologies, information technology, biomedical engineering, and medicine[35]. Telemedicine has seen growth in telehealth services, providing remote consultations and monitoring vitals during the

COVID-19 pandemic. There has also been increased access to healthcare for rural and underserved populations.

I.2.6.2 Artificial Intelligence (AI) and Machine Learning (ML)

- AI-driven diagnostics and predictive analytics improving accuracy and efficiency.
- Personalized medicine through AI algorithms tailoring treatments to individual patient data.

I.2.6.3 Electronic Health Records (EHRs)

Electronic Health Records are electronic versions of patients' healthcare records. An electronic health record gathers, creates, and stores the health record electronically. The widespread adoption of EHRs for better patient data management, care coordination, and reduction of medical errors.

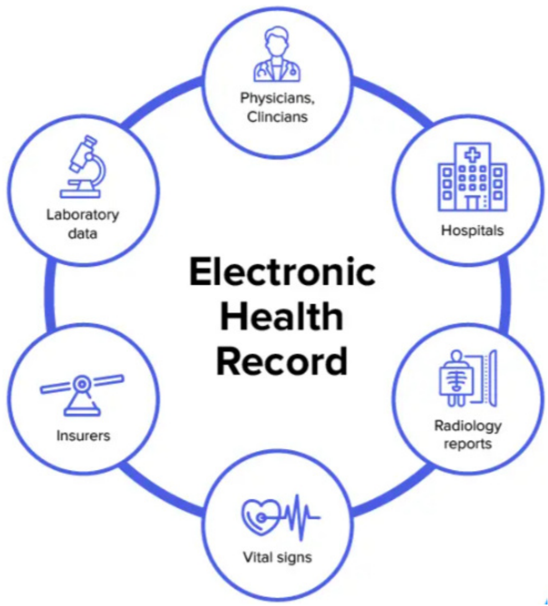

Figure I.9: information related to patients' health

I.2.6.4 Wearable Technology

Use of wearable devices for continuous health monitoring, promoting preventive care, and enhancing patient engagement.

I.2.6.5 Robotics and Automation

Implementation of robotic systems in surgeries, rehabilitation, and routine tasks, increasing precision and reducing human error.

I.2.7 AI in Healthcare

AI is transforming healthcare, with advancements in different practices[49].

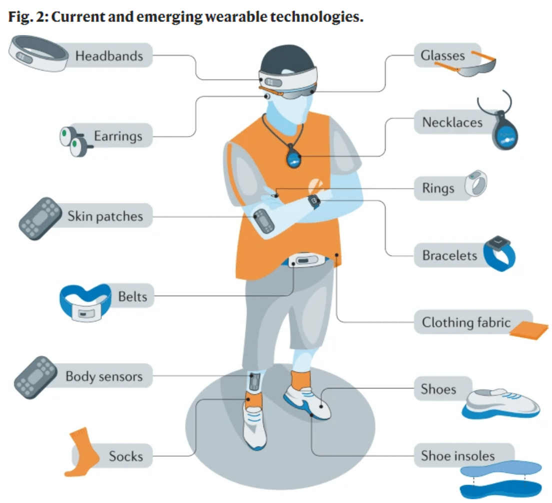

Figure I.10: Wearable Technology

I.2.7.1 Diagnostics

Using advanced imaging technologies (MRI, CT scans) and molecular diagnostics improved early detection of diseases. It is marked as a point-of-care diagnostic tool that enables rapid and accurate disease identification.

I.2.7.2 Treatment

- The development of targeted therapies and personalized medicine based on genetic profiles resulted possibility of early treatments and the creation of customized drugs.
- Minimally invasive surgical techniques and advanced prosthetics improving patient outcomes.

I.2.7.3 Patient Management

- Integrated care models provide coordinated and comprehensive services across different levels of healthcare.
- AI enhanced patient monitoring and management through telehealth and wearable devices.

I.2.7.4 Healthcare Delivery

- Innovative care delivery models such as home-based care and mobile health units increased accessibility.
- Digital health platforms streamline appointment scheduling, medication management, and patient education.

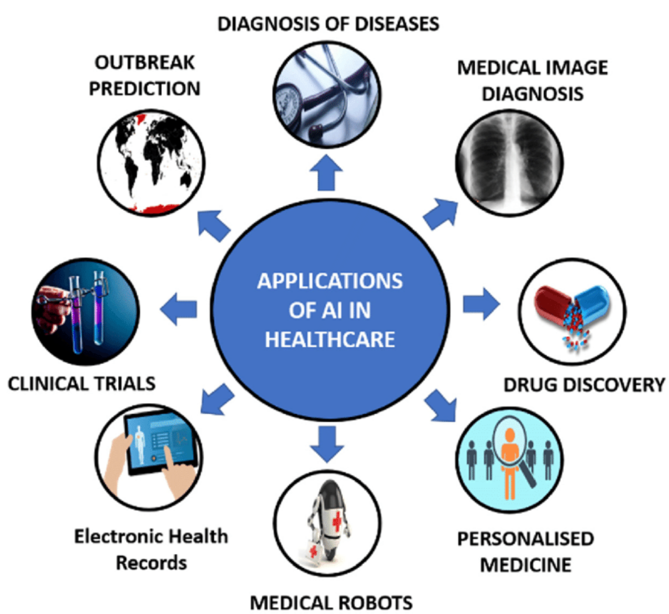

Figure I.11: Applications of AI in healthcare

I.2.8 Challenges in Healthcare

Healthcare systems worldwide face a multitude of challenges that impact the quality, accessibility, and efficiency of care. Here are some of the primary challenges:

- **Cost and Affordability :** The high cost of healthcare services and medications leads to financial strain on patients and disparities in access. And the rise in healthcare expenditure puts pressure on national budgets and insurance systems.

- **Access to Care :** There is a geographic disparities in healthcare access, especially in rural and underserved areas. In addition to the shortage of healthcare professionals which impacts the availability and quality of care.

- **Quality and Efficiency :** There is variability in the quality of care across different regions and healthcare providers. Healthcare delivery needs improvements to reduce waiting times and enhance patient experiences.

- **Aging Population :** The increase in the elderly population leads to a higher demand for healthcare services, chronic disease management, and long-term care. This puts a strain on healthcare resources and infrastructure to cater to the needs of the aging demographic.

- **Public Health and Disease Management :** The ongoing management of infectious diseases such as COVID-19, tuberculosis, and HIV/AIDS. And the rising prevalence of chronic diseases like diabetes, cardiovascular diseases, and cancer requires long-term care strategies.

- **Mental Health :** There has to be more awareness and need for mental health services addressing issues such as depression, anxiety, and substance abuse.

The stigma and lack of resources impact the quality and accessibility of mental health care.

I.3 Internet of Things (IoT)

IoT is revolutionizing industries and everyday life, presenting a spectrum of possibilities and challenges. This section offers an introduction to IoT, beginning with its definition and exploring fundamental concepts, architecture and applications.

I.3.1 Definition

The IoT refers to the network of interconnected devices that communicate and exchange data with each other through the Internet. These devices, often embedded with sensors, software, and other technologies, can collect and share data without human intervention. The primary goal of IoT is to create smarter environments by integrating physical objects with the digital world, enabling improved efficiency, convenience, and decision-making.

I.3.2 Basic Concepts

- **Connectivity:** Devices are connected to the internet, allowing them to communicate with each other.
- **Sensors:** Devices collect data from their environment through various sensors.
- **Data Processing:** Collected data is processed, often in real-time, to derive meaningful information.

- **Automation:** IoT systems can automate tasks based on the data they collect and process.
- **Interoperability:** Different devices and systems need to work together seamlessly.

I.3.3 Historical Development

The concept of IoT has evolved over several decades:

- **the 1960s-1980s:** Early foundations of IoT began with the development of the internet and Advanced Research Projects Agency Network (ARPANET), focusing on connecting computers.
- **1990s:** The term "Internet of Things" was coined by Kevin Ashton in 1999. This period saw the advent of Radio Frequency Identification (RFID) technology, which allowed objects to be identified and tracked.
- **2000s:** The Rapid growth of wireless technologies, such as Wi-Fi and Bluetooth, facilitated the growth of IoT. Cloud computing also emerged, providing the necessary infrastructure for storing and processing large amounts of data.
- **2010s:** IoT gained significant traction with the rise of smart devices and applications. Key technological advancements included the development of IPv6, which expanded the number of possible IP addresses, and improvements in sensor technology.
- **2020s and beyond:** IoT continues to expand rapidly, with increasing integration into various industries and daily life.

I.3.4 Key Components

- **Sensors:** These devices gather data from the environment, such as temperature, humidity, motion, light, and more. They are the primary means by which IoT devices interact with the physical world.

- **Data Processing:** Once data is collected, it needs to be processed to derive actionable insights. This can occur locally on the device (edge computing) or be sent to centralized servers (cloud computing) for more extensive analysis.

- **H2M:** (Human-to-Machine) communication in IoT allows users to interact with connected devices. It enables management, data monitoring, and personalization. In healthcare, it facilitates real-time interaction with smart medical devices for enhanced monitoring, data collection, and personalized care.

- **M2M:** M2M (Machine-to-Machine) technology enables automated transmission of data between distinct devices. Examples include smart meters, vehicle telematics, asset tracking, wearable tech, and automated supply chain management. M2M operates without human intervention but allows for real-time interventions when needed.

- **Connected object:** A connected object (CO) has internet connectivity or an algorithm for enhanced functionality. Examples include smart home devices, industrial machinery, vehicles, and wearables. These objects enable remote monitoring, control, and automation, increasing efficiency and innovation.

I.3.5 Architecture of IoT

The architecture of IoT has been proposed as a 6-tier architecture based on a hierarchical network structure, it is typically separated into six layers. The following describes each of the six IoT levels [54]:

At first, there is **Coding layer** where each object is given a distinct ID, making it simple to identify between things. Next, we have **Perceptual layer** that gives each object a physical meaning. It is made up of several data sensors that may measure an object's temperature, humidity, velocity, position, and others. This layer gathers important data from the sensing devices about an object and transforms it into a digital signal. Following that **Network layer** receives the digital signal to take further action. Then, **Middleware layer** is where data is sent by the sensor device is handled. It includes innovations like cloud computing and ubiquitous computing that let you store all the data you require and instantly access the database. **Application layer** which is based on processed data, this layer, enables IoT applications for various industries. As applications support IoT development, this layer is advantageous for the broad expansion of IoT networks. Smart homes, intelligent transportation, and smart planets are examples of IoT-related applications. And finally, **Business layer** is the one that oversees IoT applications and services.

I.3.6 Characteristics of IoT

The fundamental characteristics of the IoT are as follows [50]:

- **Inter-connectivity:** With regard to the IoT, anything can be interconnected with global information and communication infrastructure.
- **Things-related services:** The IoT is capable of providing thing-related ser-

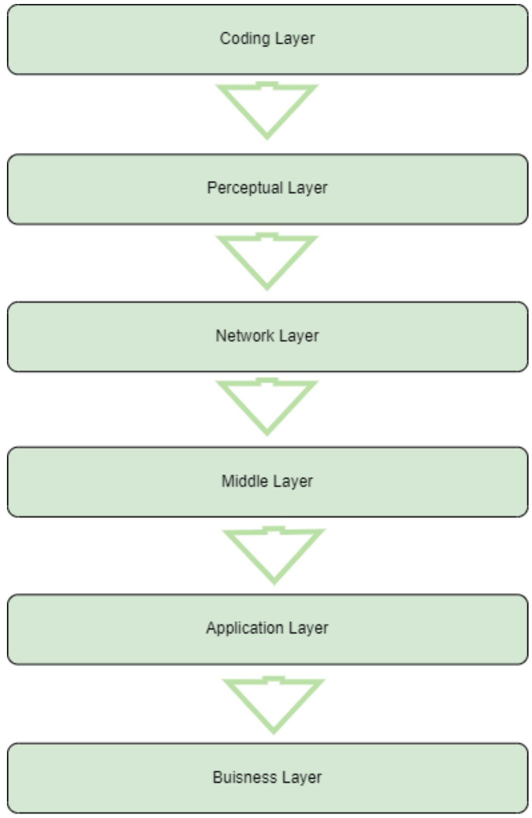

Figure I.12: Architecture of IoT

vices within the constraints of things, such as privacy protection and semantic consistency between physical things and their associated virtual things. In order to provide thing-related services within the constraints of things, both the technologies in the physical world and the information world will change.

- **Heterogeneity:** The devices in the IoT are heterogeneous based on different hardware platforms and networks. They can interact with other devices or service platforms through different networks.

- **Dynamic changes:** The state of devices changes dynamically, e.g., sleeping and waking up, connected and/or disconnected as well as the context of devices including location and speed. Moreover, the number of devices can change dynamically.

- **Enormous scale:** The number of devices that need to be managed and that communicate with each other will be at least an order of magnitude larger than the devices connected to the current Internet.

  Even more critical will be the management of the data generated and their interpretation for application purposes. This relates to the semantics of data, as well as efficient data handling.

- **Safety:** As we gain benefits from the IoT, we must not forget about safety. As both the creators and recipients of the IoT, we must design for safety. This includes the safety of our personal data and the safety of our physical well-being. Securing the endpoints, the networks, and the data moving across all of it means creating a security paradigm that will scale.

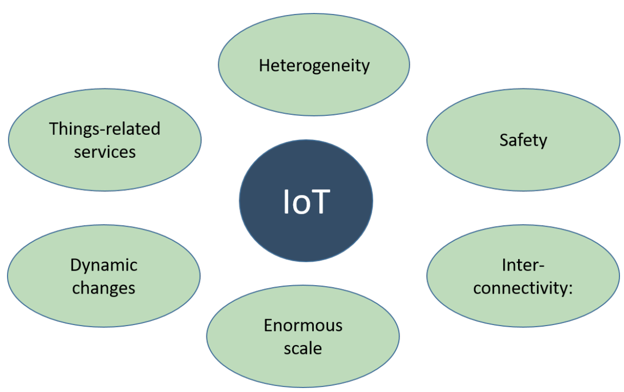

Figure I.13: Characteristics of IoT

I.3.7 Applications in General

IoT has applications across numerous sectors, enhancing efficiency and providing innovative solutions[38]. Examples include:

- **Smart Homes:** Devices like smart thermostats, lighting, security systems, and home assistants (e.g., Amazon Echo, Google Home) provide enhanced control and automation of home environments.
- **Healthcare:**Wearable devices monitor vital signs and fitness levels, while connected medical devices improve patient care and streamline hospital operations.
- **Agriculture:** IoT enables precision farming with sensors monitoring soil conditions, weather, and crop health, leading to optimized irrigation and fertilization.
- **Manufacturing:** Industrial IoT (IIoT) improves operational efficiency through predictive maintenance, real-time monitoring, and automation of production processes.
- **Transportation:** Connected vehicles and smart traffic systems enhance safety, efficiency, and user experience in transportation networks.
- **Retail:** IoT applications include inventory management, personalized shopping experiences, and automated checkout systems.

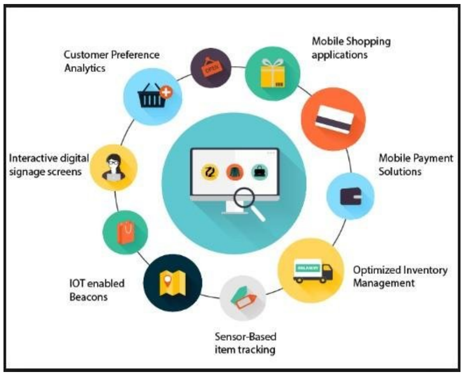

Figure I.14: Smart Retail

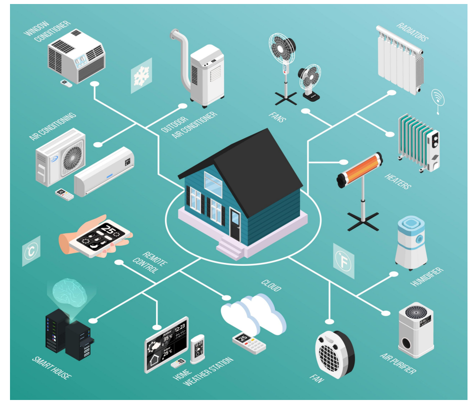

Figure I.15: Smart house

I.3.8 Components of IoT system

The four fundamental components of an IoT system (function and mechanism)[4]:

- **IoT devices and sensors:** A Sensor is one of IoT devices that can detect, measure, and collect data from the physical environment such as light, motion, heat, pressure, or similar entities.
- **IoT gateways:** The IoT gateway is a bridge between sensor networks and cloud services. The role of the gateway is to process the collected data from sensors, and then send it to cloud computing.
- **Cloud function:** Cloud function facilitates the advanced analytics and the monitoring of IoT devices to shorten the execution time, reducing costs and reducing energy consumption.
- **User interfaces:** User interfaces are the visible and tangible part of the IoT system. They enable users to contact and monitor their activities in services that they have already subscribed to using an IoT system.

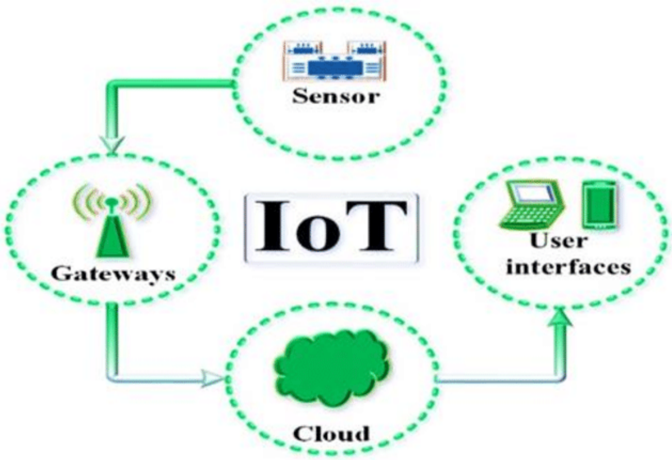

Figure I.16: Components of IoT system

I.3.9 Challenges and Opportunities

The IoT presents both challenges and opportunities in various aspects[57, 9]:

I.3.9.1 Challenges

- **Security and Privacy:** The vast number of connected devices increases the risk of cyberattacks and data breaches. Ensuring robust security measures and protecting user privacy are paramount.
- **Interoperability:** Diverse devices and standards can lead to compatibility issues. Developing universal standards and protocols is necessary for seamless integration.
- **Data Management:** The enormous volume of data generated by IoT devices requires efficient storage, processing, and analysis solutions.

- **Scalability:** IoT systems must be scalable to accommodate growing numbers of devices and users without compromising performance.
- **Energy Consumption:** Many IoT devices rely on batteries, necessitating energy-efficient designs to prolong operational life.

I.3.9.2 Opportunities

- **Enhanced Efficiency:** IoT can streamline operations across various industries, reducing costs and improving productivity.
- **Improved Quality of Life:** Smart homes, healthcare, and other IoT applications can significantly enhance convenience, safety, and well-being.
- **Data-Driven Insights:** IoT provides valuable data for analytics, enabling informed decision-making and innovation.
- **Sustainability:** IoT can contribute to environmental sustainability through optimized resource usage, reduced waste, and improved energy management.
- **New Business Models:** IoT opens up opportunities for new products, services, and business models, driving economic growth and innovation.

I.4 Convergence of AI and IoT in Healthcare (AIOT Healthcare)

I.4.1 Definition and Concept

AIOT, or the convergence of AI and the IoT, refers to the integration of AI technologies with IoT devices to enhance their capabilities and provide more intelligent, efficient, and effective solutions. In healthcare, AIOT aims to improve patient care, streamline operations, and facilitate personalized medicine by leveraging real-time data and advanced analytics[3].

I.4.2 Synergy of AI and IoT

AI enhances IoT capabilities by:

- **Advanced Data Analysis:** AI algorithms can analyze large volumes of data collected by IoT devices, extracting meaningful insights and identifying patterns that traditional methods might miss[68].
- **Predictive Analytics:** AI can predict potential health issues or equipment failures, allowing for proactive interventions[68].
- **Automation:** AI enables automated decision-making processes based on IoT data, improving efficiency and response times[68].

IoT enhances AI capabilities by:

- **Real-time Data Collection:** IoT devices provide continuous, real-time data streams that AI systems can analyze.
- **Enhanced Data Volume:** The extensive network of IoT devices increases the volume of data available for AI to process, improving the accuracy and robustness of AI models.
- **Contextual Data:** IoT devices collect diverse types of data from various sources, providing AI with a richer context for analysis.

I.4.3 Key Components of AIOT Healthcare

- **AI Algorithms:** These are crucial for processing and analyzing the data collected by IoT devices. They enable tasks such as predictive analytics, anomaly detection, and personalized recommendations.

- **Enhanced Data Volume:** The extensive network of IoT devices increases the volume of data available for AI to process, improving the accuracy and robustness of AI models.

- **IoT Devices:** These devices, such as wearables, smart sensors, and connected medical equipment, collect real-time health data from patients.

- **Data Analytics:** Advanced analytics techniques process the data to generate actionable insights. This involves real-time data processing, big data analytics, and ML models.

- **Connectivity:** Reliable and secure connectivity solutions (e.g., 5G, Wi-Fi, Bluetooth) ensure seamless data transmission and communication between devices and systems.

I.4.4 Benefits of AIOT in Healthcare

AIOT in healthcare offers several benefits that can significantly improve patient care, operational efficiency, and overall outcomes. Here are some key benefits:

- **Improved Patient Outcomes:** Continuous monitoring and early detection of health issues lead to timely interventions and better health outcomes.

- **Cost Efficiency:** AIOT solutions can reduce healthcare costs by optimizing resource utilization, reducing hospital readmissions, and minimizing unnecessary tests and procedures.
- **Personalized Care:** AIOT enables personalized treatment plans based on individual patient data, leading to more effective care.
- **Operational Efficiency:** Automation and real-time data analytics streamline healthcare operations, reducing administrative burdens and enhancing workflow efficiency.

I.4.5 Applications of AIOT in Healthcare

AIOT (Artificial Intelligence of Things) in healthcare is being applied across various areas to enhance patient care, improve operational efficiency, and advance medical research. Here are some specific application[51]s:

- **Remote Patient Monitoring:** IoT devices monitor patients' vital signs continuously, while AI algorithms analyze the data to provide alerts and recommendations.
- **Smart Diagnostics:** AI-powered diagnostic tools analyze data from medical tests and imaging, providing accurate and timely diagnoses.
- **Telehealth Services:** AIOT facilitates virtual consultations, remote monitoring, and telemedicine, expanding access to healthcare services.
- **Hospital Management:** AIOT improves asset tracking, inventory management, and workflow optimization within hospitals.

- **Patient Engagement:** AIOT systems provide personalized health recommendations, reminders, and educational content to engage patients in their care.

I.4.6 Case Studies and Examples

- **Example 1:** Wearable devices equipped with sensors monitor chronic conditions like diabetes, with AI algorithms analyzing the data to provide real-time alerts and treatment adjustments.

- **Example 2:** Smart hospital beds equipped with IoT sensors track patient movements and vital signs, alerting staff to potential issues such as bedsores or falls.

- **Example 3:** AI-powered telehealth platforms enable remote consultations and continuous patient monitoring, reducing the need for in-person visits and improving access to care.

I.4.7 Challenges and Considerations

Implementing AIOT in healthcare involves navigating several challenges and considerations:

- **Data Privacy:** Ensuring the privacy and security of patient data is paramount, requiring robust encryption, access controls, and compliance with regulations such as GDPR and HIPAA.

- **Security:** Protecting IoT devices and data from cyber threats is critical to maintaining the integrity and trustworthiness of AIOT healthcare systems.

- **Interoperability:** Achieving seamless integration and communication between diverse devices and systems is essential for the effectiveness of AIOT solutions.
- **Regulatory Compliance:** Navigating the complex regulatory landscape in healthcare requires adherence to standards and guidelines to ensure patient safety and data protection.

I.4.8 Future Trends

Looking ahead, several future trends are emerging in the integration of AIOT in healthcare[26]:

- **Edge Computing:** Processing data at the edge, closer to where it is generated, will enhance real-time analytics and reduce latency.
- **5G Connectivity:** The rollout of 5G networks will provide faster, more reliable connections, supporting the growth of AIOT applications.
- **Advanced AI Models:** Continued advancements in AI, including deep learning and NLP, will further enhance the capabilities of AIOT systems.
- **Increased Adoption of Wearables:** The proliferation of wearable health devices will provide more comprehensive and continuous health monitoring.
- **Integration with Electronic Health Records (EHRs):** Seamless integration of AIOT data with EHRs will improve clinical decision-making and patient care coordination

I.5 Conclusion

In conclusion, AIOT in healthcare represents a transformative approach to improving patient care and operational efficiencies through the integration of AI and IoT technologies. By harnessing the power of real-time data analytics, predictive algorithms, and remote monitoring capabilities, AIOT enables healthcare providers to deliver more personalized, efficient, and effective treatments. While challenges such as data security and interoperability persist, ongoing advancements and regulatory compliance efforts are paving the way for a future where AIOT plays a central role in shaping the healthcare landscape, offering enhanced patient outcomes and cost-effective healthcare solutions.

Chapter II

Overview of Robotics

II.1 Introduction

This chapter describes robotics and its historical evolution as an innovative field. Later, it discusses different types of robots found in real-world applications, followed by an exploration of the construction process, and the components and parts that combine to produce a robot.

II.2 Robotics

Robotics is an engineering field that combines electronics engineering, mechanical engineering, computer science, mathematics, and more. It aims to embody a functioning machine with understanding capabilities using advanced technologies such as machinery, electronics, control, computers, sensors, and AI. Robots are designed to work in different environments to execute any task, some of these tasks can be dangerous such as bomb detection and disaster zones, or safe but sensitive like microsurgery.

Researchers used nature concepts and designs as an inspiration to make these robots move and react with simple or complex strategies to give impressive results in any task given.

HubSpot's market analyst Mimi An describes it as the "technology able to do things as a human being can — regardless of conversation, vision, and learning, or social contact and inference", just as AI applications in iPhone-based Siri and Google Assistant [76].

II.2.1 History of robotics

The history of robots started around 1804, when weaving just like many other jobs required a bigger number of assistants to raise and lower threads to produce patterns. Many inventors tried to automate the process when Joseph-Marie Jacquard created "Jacquard Loom"(Figure II.1). This machine used patterns from punch cards and translated them into commands that determined the lifting and lowering of threads. The New York World's Fair 1911 in the USA displayed the home robot "Elektro"(Figure II.2) made by Westinghouse Electric Corporation. "Elektro" was controlled by a cable and could walk, speak 77 words, and even have a cigarette, making people's dream of the home robot more specific [76].

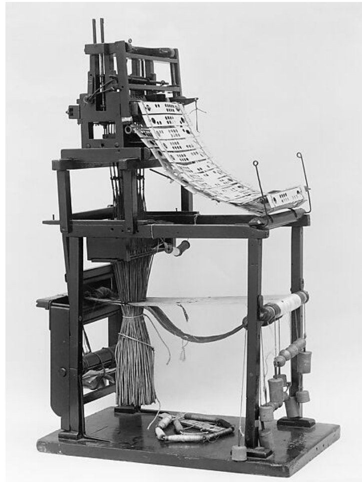

Figure II.1: Model of a Jacquard loom.

In 1949, The concept of sensor technology was introduced, when an American-born British neurophysiologist and inventor named William Grey Walter introduced a pair of battery-powered, tortoise-shaped robots that could maneuver around objects in a room(Fig II.3), guide themselves toward a source of light and find their way back to a charging station using the same components.

Figure II.2: World's First Mobile Intelligent Robot.

After that, The idea of using robotic arms in factories appeared between 1959 and 1969. The first one was only able to move up and down on the X and Y axis and possessed a rotatable, pincer-like gripper. Then the second one had more flexibility and was quicker than the first one.

Figure II.3: The tortoise-shaped robot.

AI was added to this field for the first time in 1972 when a robot named "Shakey" was first created. "Shakey" was its name because of the stuttering way it moved around, but the most interesting feature of this robot was that it was able to navigate its way across a room or push a box by observing its environment and creating a plan and executing.

Following this, numerous innovative concepts were brought to reality, and integrating human-like senses and reactions was done more and more. And since 2012, advancements in deep learning have introduced groundbreaking techniques to the world such as image recognition and voice recognition.

II.2.2 Types of robots

Robots are classified into various types based on their functionality, design, and application. Here are some of the main types based on applications:

II.2.2.1 Industrial robots

They serve as automatic workers and the most suitable alternative to manual labor since it ensures the consistency of the output with the ability to control the performance, capability, and cost. Since the introduction of the first industrial robot ever in 1961 at General Motors(Fig II.4), factories and production facilities have taken a lot of interest in this technology, which inspired them to improve these machines to satisfy additional needs.

Figure II.4: First industrial robot ever in 1961 at General Motors.

An industrial robot has been defined by ISO 8373 (published by the International Federation of Robotics, 2013) as An automatically controlled, reprogrammable, multipurpose manipulator programmable in three or more axes, which may be either fixed in place or mobile for use in industrial automation applications. Within this definition, further clarification of these terms is as follows:

- Reprogrammable – motions or auxiliary functions may be changed without physical alterations.
- Multipurpose – capable of adaptation to a different application possibly with physical alterations.
- Axis – an individual motion of one element of the robot structure, which could be either rotary or linear [75].

II.2.2.2 Service robots

Figure II.5: Robot Dogs for Search and Rescue.

Figure II.6: Unmanned combat aerial vehicle.

The definition of a service robot can be summarized as a robot that can assist humans in different tasks. This type of robot is being developed for a wide range of applications, including unmanned aircraft for the military, machines for milking cows, search and rescue robots, robot vacuum cleaners, and educational and toy robots [74]. The use of these robots is spreading day by day to ensure task efficiency and ease, but Human-robot interaction is one of the most important challenges since the service robots work in unconstructed environments. Although the keyboard, mouse, joystick, and touch screen could be utilized to control the robot, it is not an easy task for nonprofessional users to program the robots to meet users' various requests and needs. Therefore, natural behavior understanding is required for these service robots [39].

II.2.2.3 Medical robots

Figure II.7: The da-Vinci Surgical System.

According to the Journal of Patient Safety, up to 440,000 deaths each year can be attributed to medical errors, injuries, accidents, and infections. As a way to reduce this number medical robots were presented. In the late 1980s, the first commercial surgical robots were built and the first commercial myoelectric prostheses were being used in rehabilitation centers around the world [7]. And since this type of robot has to be very close to patients, this caused so much concern in the medical community and just a small number of doctors at the time accepted this intervention. Slowly and after many experiments and affirmations, including a fundamental knowledge of biological systems, experts were able to convince more doctors to accept them by offering advantages, including reduction of access to trauma, faster recovery, scar limitation, cost reduction, and so on.

One of the most known medical robotic systems is the da Vinci Surgical System. gives surgeons more precise control for a range of procedures. Using magnified 3D high-definition vision and controls that strap to a surgeon's wrists and hands, the da Vinci System makes tiny, exact incisions that human hands might not otherwise be able to make.

II.2.2.4 Field robots

This type of robot is widely used in agriculture mainly. They are performing various tasks to empower farmers to manage their fields efficiently, even in situations where labor is limited—assisting with tasks like planting, harvesting, and monitoring crops, while their aerial counterparts, drones, take to the skies for tasks like photography, delivery, and search and rescue.

Figure II.8: Farming Robot.

Field robots are also used in other areas like construction and mining. The key difference about this type is the ability to work outdoors and in conditions like unpredictable terrain, weather conditions, and variable lighting.

II.2.3 Robotic system components

Robotic systems consist of both hardware and software components, each playing a crucial role in their operation. Here's an overview of the hardware and software components typically found in robotic systems:

II.2.3.1 Hardware

A robotic system typically consists of several key components that work together to perform tasks. These components can vary depending on the type and complexity of the robot, but generally include:

Sensors

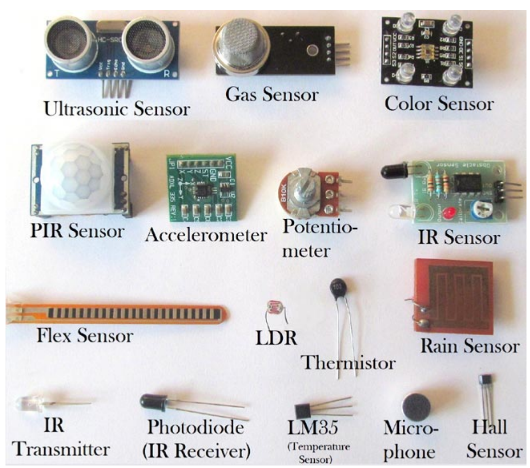

Figure II.9: Basic Sensors.

devices, machines, or subsystems that detect physical phenomena, provide feedback on the environment and translate it as an output signal to other electronics. They can be used in everyday objects to detect temperature, acidity level, finger touch, and more. With advancements in micromachinery and user-friendly microcontroller platforms, the applications of sensors have expanded beyond traditional uses, such as the integration of MARG sensors.

Actuators

Figure II.10: An Actuator.

Just like muscles in the human body, actuators produce, by translating signals as motions. It's a device that creates mechanical movement in the form of a rotating shaft or a rod that extends or retracts. Actuators in different fields from robotics and industrial automation to transportation and aerospace.

Motors

As a robot needs to move around and move its limbs, a motor is a necessary entity to enable robotic parts in different angles. There are different types of motors with different functionalities, advantages, and disadvantages each one of them is preferred for a specific task. Here are some common types of motors used in robotics:

- **AC Motor :**

  it's an electrical motor that converts alternating current into mechanical rotation. It easily operates since it only needs to be connected to an AC power sup-

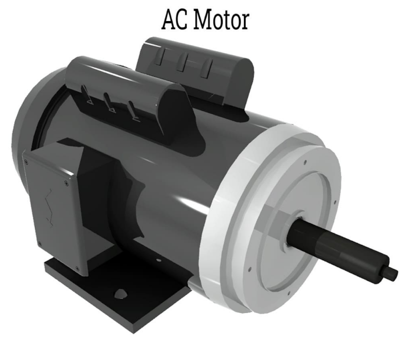

Figure II.11: AC Motor.

ply. They are easy to handle, low in cost, and require maintenance compared to other types of robots, which makes them widely used in various devices. Its disadvantage is that the speed of rotation can be controlled only by the frequency of the power source.

- **DC Motor :** just like an AC motor, a DC motor uses direct current and

Figure II.12: DC Motor

converts it into mechanical rotation. It's used in many applications including electric wheelchairs and coffee machines. DC Motors is famous because it offers many control solutions for speed and it can quickly start/stop. It also needs less maintenance and has a longer life span.

- **Servo motor :**

Figure II.13: Servo Motor.

This type of motor is a rotary actuator usually used as a sensor for position feedback since it allows for precise control of angular position or linear position (able to control the movement of an object along a straight line). Choosing this type of motor depends on the need for accurate motion control, fast response time, and continuous feedback which is much needed in the closed-loop control system. However, it can be more expensive and requires complex controlling algorithms.

- **Stepper motor :**

Figure II.14: Stepper motor.

A stepping motor is a type of DC motor that rotates in a series of small angular steps (fixed number of degrees). It comes in different sizes and styles and it's a little similar to a servo motor, only it may experience errors and can have slower response times. The advantage though is that it's very cost-effective and much simpler to use in a system. It's mostly used in open-loop systems where it can function without feedback.

Controller

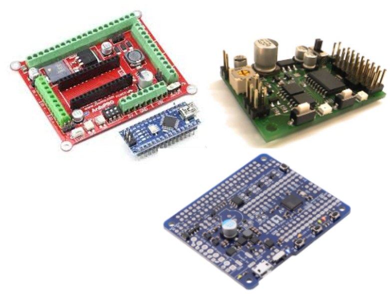

Figure II.15: Diffrent types of Controller.

It is the main component of a robot, it serves as the brain for humans. It's the computer system that receives sensory data, processes it, makes decisions, and sends signals to the actuators to control the robot's movements and actions. There are different types of controllers each with its own strengths, weaknesses, and suitable applications some are traditional controllers such as the Proportional-Integral-Derivative (PID) controllers, and more advanced types like fuzzy logic controllers, neural network controllers, adaptive controllers, and robust controllers.

Power Supply

As with any machine, robots need the power to provide the voltage signals that make the motors turn, processing process data, operate sensors, and move actuators. There are different ways to supply robots with energy :

- **Batteries:**

Figure II.16: Car Batteries.

Figure II.17: Lithium Batteries.

  using batteries is the simplest and most known way to get energy, small robots like the rover robot use small batteries (six AA batteries), and big mobile robots need batteries that are used in cars like Autonomous Electric Tractor developed by John Deere company. The problem with batteries is that they are easily affected by the weight of the robot itself and they run down over time, although Rechargeable batteries can help with this.

- **Energy harvesting :**

  It's the process of using a natural source from the environment like solar power, thermal energy, or wind energy . . . and converting it into usable electrical power, then stored for use. This method is very good for the environment and

Figure II.18: Wind Farm.

it can last for a long time but its disadvantage is that it only provides a very small amount of power, which makes it used more in small wireless autonomous devices. An example of power harvesting exists in pacemakers that gain their energy through the oxidation of blood sugar.

- **Hardwired power source :** it means providing energy by directly integrating it into the electrical system. For example autonomous floor-cleaning, the connection to outlets or stations to replenish its power supply when needed

End-Effectors

This component is an important part of some robots, it's possible not to find it in others and that depends on the robot's functionality and what it does. Effectors are mechanical or electromechanical and they can be of any type :

- **Grippers :** allow to pick place places and manipulate objects, usually found

Figure II.19: Arm Gripper.

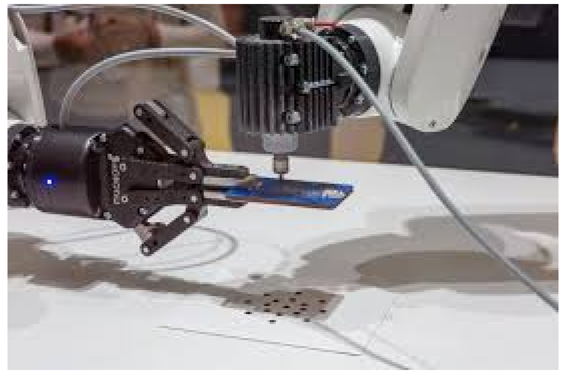

Figure II.20: Process Tool.

in robots used in factories. Some of them are designed in the shape of human hands and have fingers to allow them to do their tasks efficiently.

- **Process Tools :** this type of end effector is capable of changing and making work-pieces and not just manipulating them like grippers like 3d printing tools or CNC milling heads.

- **Welding and Cutting Tools :** as the name says these are used for cutting and welding elements like wood or metal most industrial robots have them. examples of this end effector are welding torches and laser cutting heads How to Use an end effector With Your Robot

Choosing what type of effector depends on the task assigned to the robot and what purpose it serves and using them depends on communication protocols and programming interfaces.

II.2.3.2 Software and Algorithms

Software and algorithms enable robots to perform complex tasks and make autonomous decisions. They process sensor data to interpret the environment, control movements through advanced control systems, plan paths, avoid obstacles, and make high-level decisions using AI and ML. This integration allows robots to operate efficiently and independently in dynamic and unpredictable environments, developing their functionality and adaptability across various applications.

Software Frameworks

Software frameworks in robotics are collections of pre-written code, libraries, tools, and conventions that provide an environment for developing robotic applications. These frameworks simplify the complex tasks associated with programming robots by offering reusable modules and components, thereby accelerating development, ensuring Interconnection, and promoting best practices. These are some of the most known software frameworks used in robotics:

- **Robot Operating System (ROS):** is an open-source framework that helps researchers and developers build and reuse code between robotics applications. It is a collection of programs that allow a user to easily control the mobile operations of a robot. The operating system also provides a user-friendly environment for the execution of the application programs, with the ultimate goal of producing useful work. A user-friendly environment is one in which the low-level details of the bare hardware machine are separated and hidden so as to provide the user with an interface that is clean, uncluttered, and easy to navigate[32].

- **Yet another robot platform (YARP) :** is written by and for researchers in humanoid robotics, who find themselves with a complicated pile of hardware to control and with an equally complicated pile of software. Achieving visual, auditory, and tactile perception while performing elaborate motor control in real time requires a lot of processor cycles[40].

Programming Languages

A programming language is a formal language comprising a set of instructions that can be used to produce various kinds of output, including software applications, scripts, and control systems for hardware. In robotics, programming languages are crucial for developing control algorithms, sensor processing, robot behavior modeling, and interfacing with hardware components. Some of the most common programming languages used in robotics include:

- **Python:** Renowned for its simplicity and readability, Python is widely utilized in robotics for prototyping, scripting, and developing ML algorithms.
- **C++:** Highly valued for its performance and efficiency, C++ is commonly employed in real-time control, hardware interfacing, and performance-critical applications.
- **Java:** Favored in certain educational and research robotics projects because of its portability and extensive libraries.

Algorithms

There are several algorithms, including:

Mapping algorithm Robotic mapping addresses the problem of acquiring spatial models of physical environments through mobile robots [67]. These algorithms enable robots to understand their surroundings, which is necessary for navigation and task execution. One of the most commonly used techniques is Simultaneous Localization and Mapping (SLAM), where the robot builds a map while keeping track of its location within it. There are several commonly used map representations in robotic systems, including Point Cloud maps, Geometric maps, and Occupancy Grid maps. Each map type serves a specific purpose in capturing and representing the environment [16]. The choice of mapping algorithm depends on the application and the environment's complexity landmarks.

Localization algorithm: These can be defined as the process of tracking and controlling the movement of objects from one location to another[60]. There are many techniques such as dead reckoning, which involves calculating the current position based on previous positions and motion data, but this method can accumulate errors over time. Some other complicated approaches like Kalman filters and particle filters offer probabilistic methods for better accuracy. These algorithms integrate data from various sensors, such as GPS, IMUs, and cameras, to provide a reliable estimate of the robot's position, allowing it to navigate while avoiding obstacles.

Navigation algorithm Navigation and obstacle avoidance are some of the fundamental problems in mobile robotics, which have been solved by various researchers in the past two decades. The aim of navigation is to search an optimal or suboptimal path from the start point to the goal point with obstacle avoidance competence[48]. Path planning algorithms, such as Dijkstra's, A*, and Rapidly-exploring Random Tree (RRT), calculate the optimal path for a robot to follow, considering factors like distance, energy consumption, and time.

II.3 Combining Robotics with AI

Integrating AI with robotics significantly enhances the capabilities of robots by enabling them to learn from data, make decisions, and adapt to new situations. This synergy leads to improved accuracy, efficiency, and patient outcomes in medical applications. In this section, we explore the various forms of AI utilized in the field of robotics.

II.3.1 Reinforcement Learning (RL)

RL is a type of machine learning paradigm where an agent learns to make sequences of decisions in an environment to maximize cumulative rewards. Unlike supervised learning, where the algorithm is trained on labeled input-output pairs, RL learns through trial and error, receiving feedback in the form of rewards or penalties [73].

II.3.1.1 Key Concepts

To provide a foundational understanding, here are some key concepts essential for comprehending the integration of AI in robotics [73].

- **Agent:**Agents are entities that perceive the environment, select actions, and receive rewards. They can be simple algorithms or complex systems capable of learning and decision-making.
- **Environment:**The environment is the external system with which the agent interacts. It includes everything the agent cannot control, such as the rules of a game, the physical world, or the behavior of other agents.
- **Actions:** Actions are the decisions or choices available to the agent at any

given state. These actions lead to transitions from one state to another, ultimately determining the agent's interaction with the environment.

- **Rewards:** Rewards are feedback from the environment that the agent uses to learn. They indicate the immediate benefit or detriment of taking a particular action in a specific state. The agent's goal is to maximize the cumulative reward over time.

- **Policy:** The policy is the strategy or rule that the agent follows to determine its actions based on the current state. It defines the agent's behavior and can be deterministic or stochastic.

- **States:**States represent the current situation or configuration of the environment. They capture all relevant information that the agent needs to make decisions, including past actions and rewards.

II.3.1.2 Significance in AI and Robotics:

RL enables robots to learn optimal behaviors through trial and error, making it essential for tasks requiring adaptability and real-time decision-making in dynamic environments.

II.3.1.3 Proximal Policy Optimization (PPO):

II.3.1.3.1 Explanation of PPO and Its Advantages: Proximal Policy Optimization (PPO) is an advanced RL algorithm that improves the stability and performance of policy gradient methods. PPO uses a surrogate objective function to ensure stable and reliable policy updates, which is critical in training complex robotic systems.[59].

II.3.1.3.2 Usage in Training AI Models:

PPO is widely used in training AI models for decision-making and control tasks in robotics. It balances exploration and exploitation, allowing robots to learn effective strategies while maintaining stability during the learning process.

II.3.1.3.3 Mathematical Relations :

- **Policy Update :** PPO updates the policy by optimizing a clipped surrogate objective function:

$$L^{CLIP}(\theta) = \mathbb{E}_t \left[\min \left(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t \right) \right] \tag{II.1}$$

[59]. where $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$ is the probability ratio, $\hat{A}_t$ is the advantage estimate at time step t, and ϵ is a hyperparameter that controls the clipping range.

- **Advantage Estimation:**The advantage function $\hat{A}_t$ measures how much better an action a_t taken in the state s_t is compared to the expected value, and it can be estimated using:

$$\hat{A}_t = \delta_t + (\gamma\lambda)\delta_{t+1} + (\gamma\lambda)^2\delta_{t+2} + \dots \tag{II.2}$$

where $\delta_t = r_t + \gamma V(s_{t+1}) - V(s_t)$ is the temporal difference error, γ is the discount factor, and λ is the Generalized Advantage Estimation (GAE) parameter. [59].

II.3.1.4 PPO in Sports

- **Applications in Sports Analytics and Training:** PPO has been utilized in sports to optimize strategies, enhance player performance, and simulate various game scenarios. For example, AI agents trained with PPO can learn tactics in soccer, basketball, and other sports by simulating matches and adjusting strategies based on outcomes.

- **Examples and Benefits:** In soccer, PPO can help AI agents learn to cooperate, pass the ball efficiently, and make strategic plays. These AI agents can provide insights into optimal game strategies, player positioning, and decision-making under pressure.

- **Insights Applied to Medical Robotics :** The adaptability and strategic learning capabilities of PPO in sports can be translated to medical robotics. For instance, a medical robot can learn to navigate complex hospital environments, optimize its path to deliver medication, and interact with patients in a personalized manner, all while continuously improving through feedback.

II.4 Chatbot Technologies and Natural Language Processing (NLP)

AI chatbots, or AI Conversational Agents (CA), are software applications that communicate using natural language. They are interactive systems in which humans interact with computers [46].

In recent years, the use of CA has become more widespread due to more recent advances in AI and ML, as well as the launch of frameworks for CA integration by

Microsoft and Meta.

II.4.1 Architecture

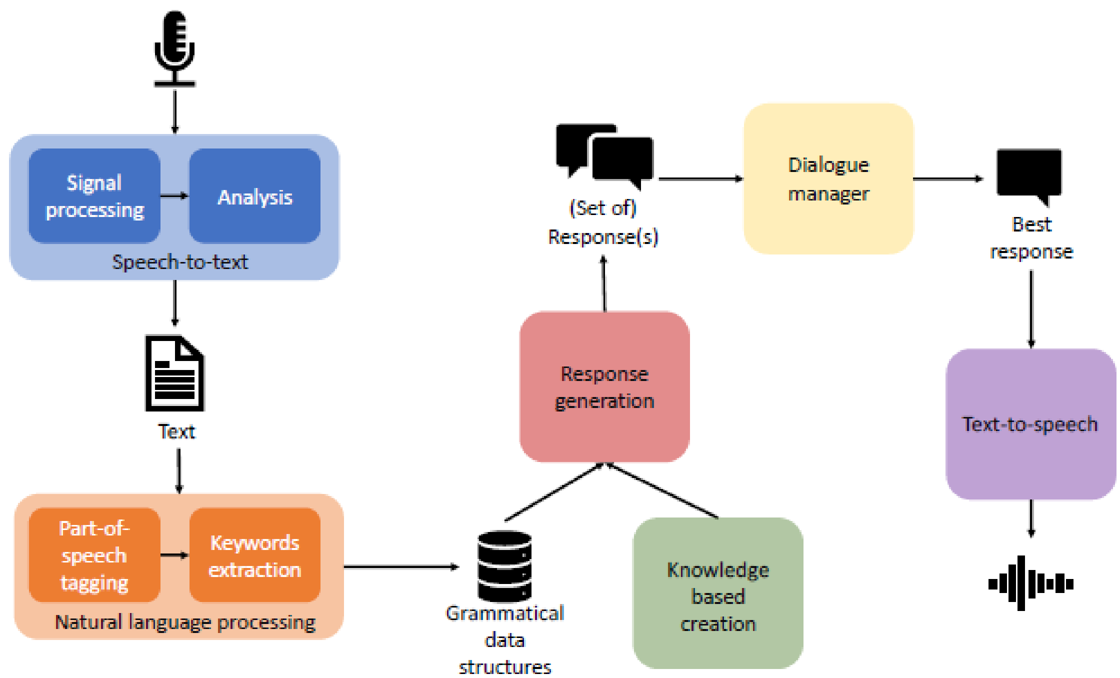

Figure II.21: Architecture of a basic spoken chatbot.

Although there are many chatbots, they all have the same architecture. this architecture is generally composed of five principal components. in this section, we will present the most common version.

II.4.1.1 Automatic Speech Recognition (ASR)

The ASR aspect represents one of the AI technologies for information herbal language. Its task is to take an acoustic speech signal, analyze it, and transcribe it into a sequence of words. It is likewise referred to as voice popularity or speech-to-text[47].

II.4.1.2 Natural language understanding (NLU)

The NLU, issue is in the middle of any Conversational AI (CA) system. It's the part that figures out what the user truly means by what they say. There are foremost approaches to building an NLU element:

- Using Rules: This approach is predicated on pre-described policies to recognize personal input.
- Machine Learning: This more advanced technique trains a version on a massive quantity of facts to apprehend two key matters:
  - intent: The universal intention of the consumer.
  - Named Entities: Important information inside the message.

By expertise in these factors, the NLU element helps the CA system interpret personal requests and reply for that reason.

II.4.1.3 Dialogue management

The DM, or Dialogue Manager, is the brain behind a Conversational AI (CA) gadget. It preserves the communication flow easily. Its paintings are:

- Tracks the verbal exchange: It recalls what's been said before, constructing a context for the current discussion.
- Decides what's next: Based on what the consumer desires (their purpose), the DM determines how to reply. This could contain accessing data from other sources like databases or online offerings.

In quick, the DM is the part that makes sense of the communique and guides the CA to present the best possible response.

II.4.1.4 Generation of response

An NLG gadget can flip ideas into written text or translate question consequences into smooth-to-recognize language. It then promises these statistics to the person through textual content. There are 3 important ways to build NLG structures: Rule-based, Template-based, or Corpus-based.

1. The Rule-based method depends on pre-described systems to make sure of constant phrasing and correct grammar. This gives you a whole lot of manipulation over the answers.

2. Template-based total approach permits for greater natural communique using fashions which can create one-of-a-kind responses for every state of affairs. However, these methods are more difficult to control and might generate nonsensical textual content.

3. The Corpus-based total approach additionally produces various outputs, however much like the template-based approach, they may be hard to control and may generate unclear textual content.

II.4.1.5 Text to speech (TTS)

TTS technology can turn written text into audible speech. This is especially useful for visually impaired folks who can not examine textual content on their own. The TTS issue works in essential steps:

- NLP: This breaks down the text into personal sounds, just like the building blocks of speech.
- Speech Synthesizer: This takes the sounds from NLP and creates a speech audio signal that you can listen to.

II.4.2 NLP technology

NLP is an area of AI that teaches computers to recognize and work with human language. This consists of expertise in spoken words (like with voice assistants), written textual content (like with chatbots), or even producing human-like textual content itself.

One of the most crucial duties in NLP is Text Classification. This is in which a computer program sorts written text into predefined categories. In reality, many other NLP duties are a category problem. For instance, sentiment evaluation (figuring out if the textual content is wonderful or terrible) may be the idea of classifying text like "happy" or "sad"[77].

II.4.3 How NLP helps chatbots to understand user's query

NLP acts as a bridge between the complex world of human language and the rule-based world of computers, allowing chatbots to understand user queries in a more meaningful way. Here's how NLP helps chatbots become better conversation partners:

II.4.3.1 Data Preprocessing

Data preprocessing prepares raw data for analysis by cleaning (handling missing values, removing duplicates, correcting errors), transforming (scaling, encoding), and engineering features to capture relevant patterns. It also includes dimensionality reduction and data integration. These steps improve data quality and algorithm performance, making machine-learning models more effective.

II.4.3.2 Feature Extraction

Data preprocessing and feature extraction are closely related but distinct stages in data preparation. Data preprocessing involves cleaning, transforming, and integrating raw data to enhance quality and usability. Feature extraction, on the other hand, focuses on identifying and selecting the most informative features for modeling. Techniques for feature extraction include dimensionality reduction methods like PCA and methods specific to data types, such as word embeddings for text or image feature extraction for computer vision. In summary, data preprocessing ensures data is properly formatted and cleaned, while feature extraction identifies the most relevant features for modeling.

II.4.3.3 Classification Methods

1. Intent Classification

    - NLP techniques like tokenization (breaking down sentences into words) and stemming lemmatization (reducing words to their root form) are used to analyze user queries.
    - ML algorithms are trained on large datasets of labeled queries with specific intents (e.g., "book appointment", "check order status").

- Based on the extracted features and learned patterns, the NLP model classifies the user's intent, allowing the chatbot to respond appropriately.

2. Entity Recognition

- NLP can identify and classify specific entities within a user's query, such as dates, times, locations, or product names.
- This allows the chatbot to extract relevant information and personalize the response. For example, if a user asks ”What are your opening hours today?”, NLP can recognize ”today” as an entity and provide the specific schedule.

3. Sentiment Analysis

- NLP techniques can analyze the sentiment of a user's query, whether positive, negative, or neutral.
- This helps the chatbot tailor its response accordingly. For example, if a user expresses frustration, the chatbot can apologize and de-escalate the situation.

II.4.4 Existing Approaches to Conversational AI in Robots

Making robots that can talk like humans is a developing area. There are different ways to do this, so robots can have more natural talks with people.

- Rule-based Systems follow set rules to give responses based on what the user says. These can work for simple talks but cannot often handle complex conversations well.

- ML uses techniques to train models on data, so they can understand input and give responses. The models learn from many human conversation examples to respond more like humans.
- Deep learning uses special computer programs called neural networks. These programs can understand the context and have conversations that seem natural. Neural networks learn from examples to get better at chatting over time.
- RL helps robots learn by giving rewards. When robots interact well with people, they get rewards. By getting rewards for good interactions, robots slowly learn how to interact better.
- Many ways can make talking with robots better. Robots can see and hear things like speech, text, pictures, and body moves. When robots have cameras and sensors they can understand what people are doing and feeling. This helps robots talk better.
- Putting robots in the real world is important. AI in robots lets them see and interact as things happen. This means robots can get information through their bodies. Having a physical form makes robot talk more natural.
- Modern robot chat systems often use many techniques together. Simple questions can use rules while complex replies need deep learning models. Combining methods makes robot responses better and more reliable.

II.4.5 Model Types Used in Advanced Chatbots

Advanced chatbots leverage various types of models to manage and optimize interactions with users. This document outlines key model types used in developing sophisticated chatbots.

II.4.5.1 Seq2Seq Models (Sequence to Sequence)

Seq2Seq models utilize Recurrent Neural Networks (RNNs) or Long Short-Term Memory (LSTM) networks for encoding input sequences and decoding output sequences.

Applications

- Machine translation
- Text summarization
- Conversational agents

II.4.5.2 Transformers

Transformers use self-attention mechanisms to process input sequences in parallel, effectively handling long-range dependencies.

Examples

- GPT (Generative Pre-trained Transformer)
- BERT (Bidirectional Encoder Representations from Transformers)

Applications

- Advanced conversational agents
- Language Translation
- Text generation

II.4.5.3 Retrieval-Based Models

Retrieval-based models select the most appropriate response from a repository of predefined responses based on user input.

Techniques

- TF-IDF (Term Frequency-Inverse Document Frequency)
- BM25
- Dual-encoder networks

Applications

- Customer support
- FAQs
- Technical assistance

II.4.5.4 Generative Models

Generative models dynamically create responses using deep learning, relying on extensive training data to generate natural replies.

Examples

- OpenAI's GPT models
- Google's Meena

Applications

- Free-form conversation
- Creative writing assistants

II.4.5.5 RL Models

RL models optimize chatbot responses based on feedback from interactions, improving performance through rewards and penalties.

Techniques

- Q-learning
- Deep Q-Networks (DQNs)

Applications

- Adaptive learning systems
- Dynamic conversational agents

II.4.5.6 Memory-Augmented Models

Memory-augmented models use external memory components to handle long-term dependencies and retain context over extended conversations.

Examples

- Memory Networks
- Neural Turing Machines

Applications

- Complex dialogue systems
- Customer service bots

II.4.5.7 Hybrid Models

Hybrid models combine various approaches, such as rule-based systems with ML, to leverage the strengths of each.

Applications

- Complex customer interactions
- Systems requiring a balance between predefined responses and dynamic generation

II.5 related works

In this section, we will briefly talk about works related to our project and compare them as follows:

II.5.1 Mapping

In this paper [78], the authors used images acquired from cameras and other image sensors. The study showed different comparisons between many algorithms that treated an image itself or only features like outdoor, indoor and/or relatively stationary environments. It compared the results with Absolute Trajectory Error, GPU Power Consumption, Total Power Consumption, CPU Usage, and RAM Usage. Some methods were unsuitable for large environments due to excessive RAM usage, and others were unuseful in untextured environments with short and straight trajectories.

In another paper [10], researchers experimented with 2D Lidar and it made significant results. It showed two algorithms (Gmapping SLAM and Hector SLAM algorithms), and how they realize the robot's real-time positioning and indoor map creation. The issue that occurs when using these ideas is overlooking certain obstacles if fixed at a particular height or obstructed by components of a robot.

To avoid this issue [29] proposes a data fusion building method using single-line LiDAR and depth camera, which is more accurate than a single sensor for building maps and makes up for the deficiencies of incomplete detection and low accuracy. On the other hand, authors in [30] used 2D LIDAR but made sure to gather data from different angles, so it doesn't miss obstacles that can't be detected at certain heights using rotations. Through the experimental tests, the entire system can achieve the 3D reconstruction by two means, including incremental 2D scan along movement, or the 3D scan completed at fixed locations.

II.5.2 Localization

Localization can be achieved while mapping at the same time, using the Simultaneous Localization and Mapping (SLAM) methods, just as [10], [78], and [44] achieved the purpose of localizing in indoor areas using lidar and images. However, in a crowded and large area with many floors, like a hospital, this method can be confusing and ineffective for a robot.

In this paper [61], the authors made comparisons between different techniques, they used Odometry in the beginning, which utilizes motion sensors, to determine the change in position of an object relative to a previous position over time. But there is always the possibility of making an error resulting from wheel slippage for example. They also mentioned using image detection, one of the most difficult to implement, but when combined with AI, image recognition specifically, it produces very good results.

Other studies [64] used WIFI to signals. The proposed method combines Wi-Fi fingerprints and smartphone IMU data, using a binarization approach for composite fingerprints. It integrates turning landmarks from floorplan markings and IMU detection. Bayesian FAB-MAP matches locations, and a global pose graph estimates trajectories using composite fingerprints, PDR, and loop closing. This Achieved 1.75m and 1.77m average localization errors on real and public datasets, surpassing classic Wi-Fi methods, especially in multiuser scenarios.

II.5.2.1 Navigation

Navigation is a crucial aspect of robotic systems, enabling them to move autonomously in various environments. This section reviews significant contributions to the field of robot navigation.

Simultaneous Localization and Mapping (SLAM)

SLAM is a foundational technique in robot navigation, allowing robots to build a map of an unknown environment while simultaneously tracking their location within it. Various approaches have been developed, including particle filter-based methods [15], extended Kalman filter (EKF) approaches [66], and graph-based SLAM [20].

Path Planning Algorithms

Path planning involves finding an optimal path from a starting point to a destination. Algorithms such as A star (A*) [24] and Dijkstra's algorithm [13] are classical approaches that have been widely used. More recent advancements include Rapidly-exploring Random Trees (RRT) [37] and its variants, which are particularly effective in high-dimensional spaces.

Deep Learning for Navigation

Deep learning techniques have significantly impacted robot navigation, particularly in perception and decision-making. CNNs are used for visual navigation [19], while RL has been applied to learn navigation policies [65]. Combining SLAM with deep learning has also shown promising results in improving robustness and accuracy [43, 21].

Multi-Robot Navigation

Coordinating multiple robots to navigate in a shared environment introduces additional challenges. Research in this area includes distributed algorithms for collision avoidance [70], swarm robotics [56], and communication strategies for efficient coordination [71].

The field of robot navigation encompasses a wide range of techniques and approaches, each contributing to the advancement of autonomous systems. Continued research and development in SLAM, path planning, deep learning, and multi-robot systems are essential for the future of robotics.

II.5.3 Chatbots

From open-source solutions to commercial ones, there is a large range of build-in chatbots nowadays, in [77] the authors talked about four examples of pre-trained models of chatbots :

Watson Assistant, created by IBM, is part of the IBM Bluemix cloud services, utilizing a neural network trained on a billion Wikipedia words to understand intents, entities, and dialogues, supporting over thirteen languages and offering extensive developer tools.

Microsoft's Language Understanding Intelligent Service (LUIS) is integrated with Azure, featuring active learning technology and pre-built models from Bing and Cortana, and supports eighteen languages via a REST API.

Wit.ai, an open-source solution, requires a messaging platform to function, handling around seventy-five languages through its HTTP API or official clients.

Rasa NLU, also open-source, runs locally, providing intent classification and entity extraction with a wide language range, and offers the benefits of self-hosted software like adaptability and data control.

Platform	Developer	Hosting	Pricing	Language Support	Features	Integration
Watson Assistant	IBM	IBM Bluemix Cloud Services	Bluemix pricing schema	Over 13 languages	Neural network, intent understanding	REST APIs, SDKs
Luis	Microsoft	Microsoft Azure	Azure pricing schema	18 languages	Specific Domain, active learning	REST API
Wit.ai	Facebook	-	Open Source	75 languages	NLP, entities, intents, contexts, actions	HTTP API
Rasa NLU	Rasa	Local	Open Source	Vast range of languages	Intent classification, entity extraction	Local installation, GitHub

Table II.1: Comparison of existing pre trained models for chatbots

II.6 Conclusion

In conclusion, robotics, RL, and chatbots represent a powerful trio within AI, each fueling the advancement of the others. RL empowers robots with greater adaptability, while chatbots, often used for robot control, leverage NLP advancements to understand human commands. Robotics research on physical embodiment informs chatbot development, leading to a more holistic understanding of the world for AI.

Chapter III

System Design

III.1 Introduction

To gain a comprehensive understanding of this robot's inner workings and to figure out all the functions and classes necessary in the system. In this section, we will talk about conceptions of our robot's system. It starts with a general architecture of the whole system, then diving deep into its subsystems, using different types of diagrams to explain throughout every part.

III.2 General Architecture

This section, presents the general architecture of the proposed system, as shown in figure III.1. This architecture is designed to enable the robot to perform actions with accuracy and efficiency.

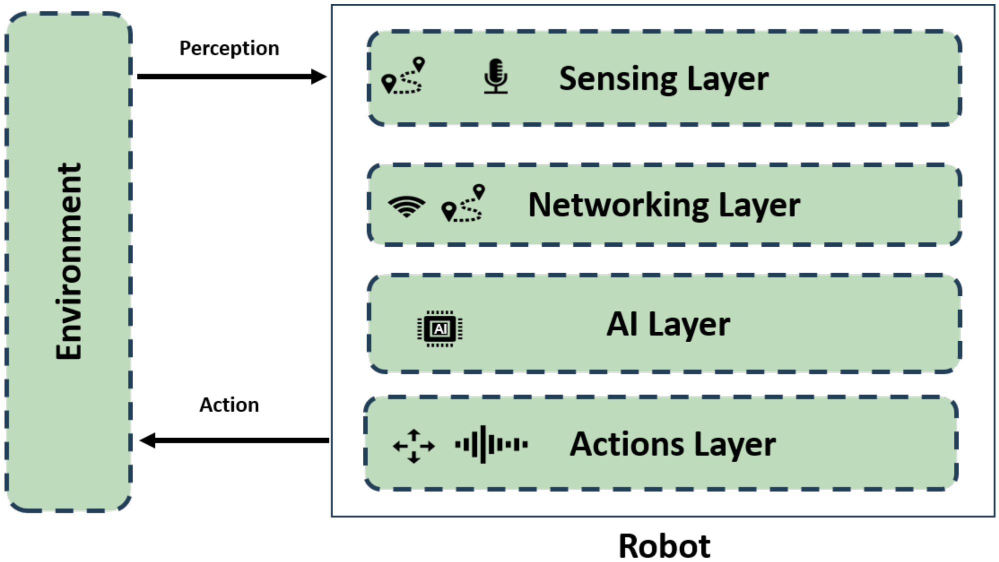

Figure III.1: General architecture of the system

- **Sensing Layer:** Utilizes various sensors to collect data about the environment, providing the robot with the necessary information to understand its surroundings.
- **Networking Layer:** Ensures connectivity between the robot's processing unit and external components, such as the ESP32 UWB WROOM CORE, to accurately determine its location.
- **AI Layer:** Hosts AI models for navigation and chatbot functionalities, representing the intelligent aspects of the system.
- **Action Layer:** Executes actions based on processed data, such as moving to a new location or responding to a question.

III.3 Detailed implementation architecture

The robot consists of six necessary modules, each with its own role. The first is a **perception module** that provides data about the robot's environment. This information along with information sent by the **location module** are processed by the **processing module**, which controls everything in the system, particularly the **movement module**, which is responsible for directing and moving the robot. The **communication module** is responsible for communication with people, and finally, the **energy module** is responsible for providing energy about the module and the state of the battery. The following Figure III.2 illustrates the detailed architecture of the system.

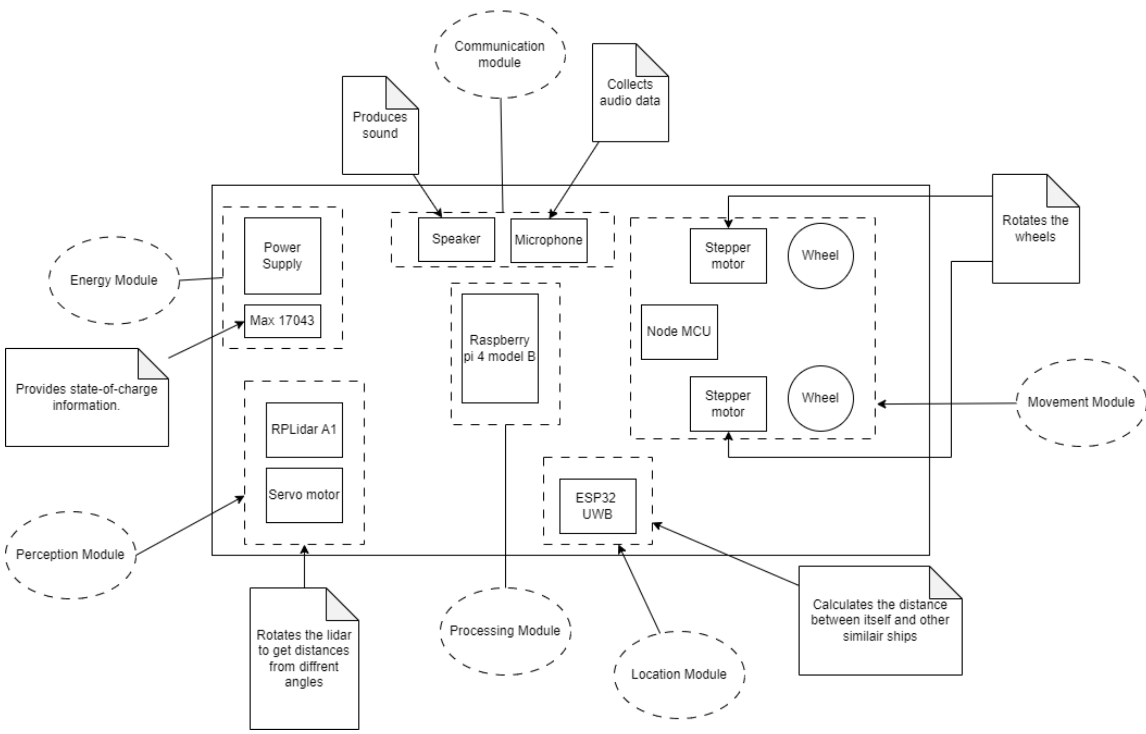

Figure III.2: Detailed architecture of the system.

III.4 Detailed system architecture

Our project's system architecture is comprised of three core components: the chatbot, the mapping and localization module, and the movement model. Each component is designed to work together, ensuring Well-integrated operation of the entire system.

III.4.1 Chatbot module

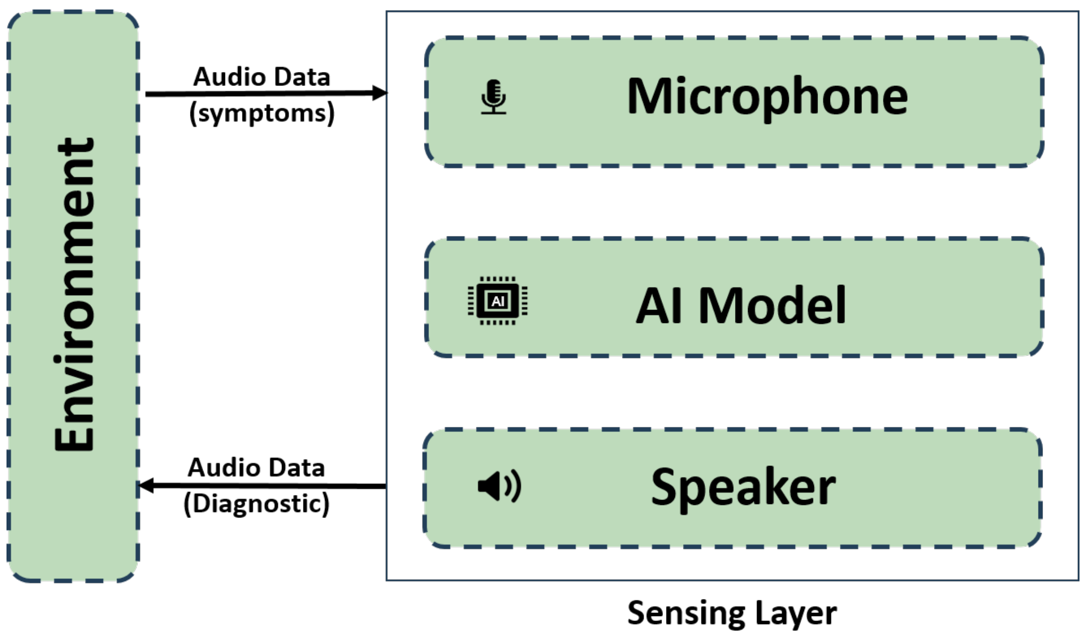

Figure III.3: The architecture of chatbot Module

The chatbot module includes a microphone that captures audio input, allowing the robot to record the user's question. The question is then converted from speech to text, processed by the AI model, and subsequently converted back to speech to be delivered through the speaker.

III.4.2 Mapping Module

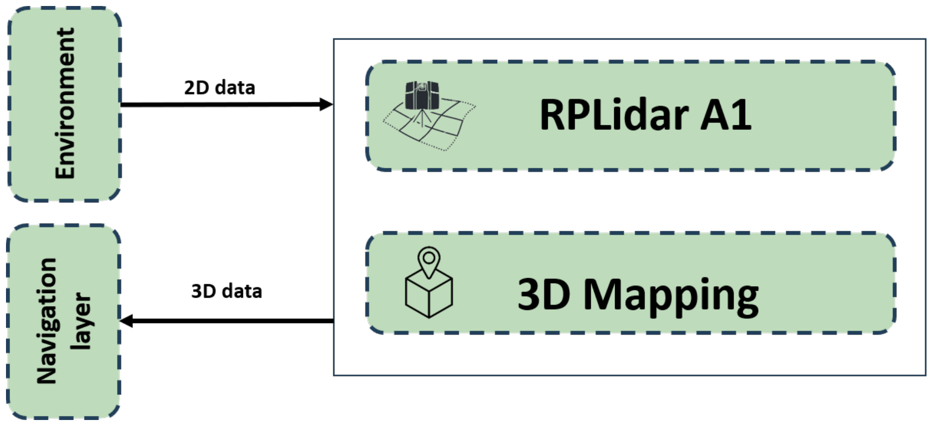

Figure III.4: The architecture of the mapping module.

The Mapping Module involves collecting 2D data using RPLidar, a rotating laser scanner that measures distances to surrounding objects. It also involves a servomotor that rotates the RPlidar to other angles to convert it into 3D data, creating a detailed representation of the environment. The conversion process involves algorithms that integrate multiple 2D scans to build a 3D map.

III.4.3 Navigation Model

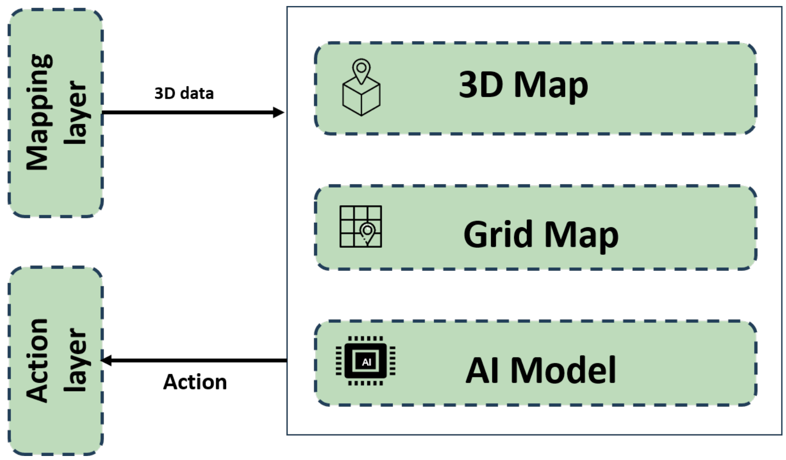

Figure III.5: The architecture of the navigation model.

The 3D map generated by the mapping model will be converted into a grid map. This conversion enables the navigation model to find a path from its current position to the goal destination.

III.5 Identification of actors

The actors are the extern entities that interact with the system, such as human beings, another system, or a robot. Our system interacts with two primary types of external actors:

- **Medical Staff:** These are the main actors, this category includes doctors and nurses who initiate interactions with the robot. They can give commands like directing the robot to move or asking questions about patient care or medical knowledge.

- **Patients:** These secondary actors interact with the robot by approaching it. Patients can request diagnoses by initiating conversations with the robot. The robot will then ask questions, provide a preliminary diagnosis, and potentially book appointments with the appropriate specialists within the hospital.

III.6 Methodologies

This section outlines the methods used to guarantee the robot's proper functionality and performance.

Simultaneous Localization and Mapping (SLAM) and RL are two foundational technologies driving the advancement of autonomous robot navigation. Each plays a distinct yet complementary role in enabling robots to effectively operate in complex environments.

III.6.1 Wake Word Interaction

Configuring the wake word detection system to identify specific vocal cues will allow the robot to initiate active listening for commands. This system enables seamless interaction between users and the robot, allowing for intuitive control through spoken instructions. Once the wake word is recognized amidst varying environmental conditions and accents, the robot swiftly processes the command, triggering predefined actions or responses.

Algorithme 1 Main Function for Wake Word Detection

```
function main():
    access_key = "**********************"
    custom_model_path = "***************"

    # Call function to create Porcupine instance wake word detection engine
    porcupine = create_porcupine_instance(access_key, custom_model_path)
    # Initialize audio stream using Porcupine parameters
    audio_stream = initialize_audio_stream(porcupine)
    print("Listening for wake word...");

    try:
        while (True) do # Continuously loop to process audio frames
            # Read audio frames from the audio stream
            pcm = read_audio_frames(audio_stream, porcupine.frame_length);
            result = process_audio(porcupine, pcm);# Process audio frames using Por-
cupine
            # Check if wake word detected; if so, invoke callback function
            if (result ≥ 0) then
                detected_callback();
            endif
        endwhile

    except KeyboardInterrupt
        print("Stopping...");

    finally# Clean up
        cleanup_audio_stream(audio_stream);
        cleanup_porcupine(porcupine);
```

III.6.2 Simultaneous Localization and Mapping

SLAM is a technique used in robotics and autonomous systems to construct a map of an unknown environment while simultaneously keeping track of an agent's location within it. SLAM enables robots and autonomous vehicles to navigate and operate in unknown environments autonomously.

III.6.2.1 Principles of SLAM

- **Mapping**:
  - SLAM involves creating a map of the environment using sensor data, such as laser range finders, cameras, or depth sensors.
  - This map can be represented in various forms, including occupancy grids, feature-based maps, or point clouds.
- **Localization**:
  - Localization is the process of estimating the robot's position and orientation within the map.
  - SLAM algorithms fuse information from sensors with motion models to estimate the robot's pose (position and orientation) relative to the map.
- **Simultaneous Estimation**:
  - SLAM algorithms solve the chicken-and-egg problem of simultaneously estimating the map and the robot's pose.
  - This involves integrating new sensor measurements with previous information to refine both the map and the localization estimate.

- **Loop Closure**:
  - Loop closure is a critical component of SLAM that detects and corrects drift in the robot's pose estimation.
  - It involves identifying and closing loops in the robot's path by recognizing previously visited places.

Algorithme 2 SLAM Algorithm

Initialize:

x_0 (Initial robot pose)

m_0 (Initial map of the environment)

while Robot pose $x_t \neq$ Target pose **do**

/* Prediction Step */

Predict the robot's pose:

$x_t = f(x_{t-1}, u_t)$

Update the covariance matrix:

$\Sigma_t = G_t \Sigma_{t-1} G_t^T + R_t$

/* Correction (Update) Step */

Sense the environment and update the estimate:

Compute measurement prediction:

$z_t = h(x_t, m_t)$

Compute Jacobian of the measurement model:

$H_t = \frac{\partial h(x_t, m_t)}{\partial x_t}$

Compute measurement covariance:

$Q_t = H_t \Sigma_t H_t^T + Q_t$

Compute Kalman gain:

$K_t = \Sigma_t H_t^T (H_t \Sigma_t H_t^T + Q_t)^{-1}$

Update pose estimate:

$x_t = x_t + K_t(z_t - h(x_t, m_t))$

Update covariance matrix:

$\Sigma_t = (I - K_t H_t)\Sigma_t$

/* Data Association Step */

Match current measurements to features in the map.

/* Loop Closure Detection Step */

Detect and close loops in the trajectory.

/* Map Update Step */

Integrate new landmarks or update existing landmarks:

$m_t = m_{t-1} \cup \{m\}$

end while

III.6.3 multi-class classification

A multi-class classification algorithm is a type of supervised learning algorithm used to classify instances into one of three or more classes. Unlike binary classification, which deals with two classes, multi-class classification handles problems where each instance can belong to one of many classes. Here's a general explanation:

III.6.3.1 Data Collection and Preprocessing

- **Data Collection**: Gather data that is labeled with one of the multiple classes. This data can come from various sources depending on the problem domain.
- **Data Cleaning**: Remove any noise or irrelevant information from the data.
- **Data Transformation**: Convert data into a suitable format for the algorithm. This may involve tokenization, lemmatization, and other preprocessing steps if dealing with text data.

III.6.3.2 Feature Extraction

- **Vectorization**: Convert textual or categorical data into numerical form. Techniques include Bag-of-Words, TF-IDF, or word embeddings.
- **Normalization**: Scale the numerical data to ensure that each feature contributes equally to the model.

III.6.3.3 Model Selection

Algorithm Choice: Choose a machine learning algorithm that supports multi-class classification. Popular choices include:

- **Logistic Regression**: With one-vs-rest (OvR) or one-vs-one (OvO) schemes.
- **Decision Trees**: Such as Random Forest or Gradient Boosting.
- **Support Vector Machines (SVM)**: Extended for multi-class problems.
- **Neural Networks**: Especially deep learning models that can handle complex and high-dimensional data.

III.6.3.4 Training the Model

- **Splitting Data**: Divide the dataset into training and validation sets.
- **Model Training**: Fit the model on the training data. For neural networks, this involves defining the architecture, loss function (e.g., categorical cross-entropy), and optimizer (e.g., SGD, Adam).
- **Validation**: Evaluate the model on the validation set to fine-tune hyperparameters and avoid overfitting.

III.6.3.5 Evaluation

- **Metrics**: Use metrics such as accuracy, precision, recall, F1-score, and confusion matrix to evaluate model performance.
- **Cross-Validation**: Perform k-fold cross-validation to ensure the model generalizes well to unseen data.

III.6.3.6 Prediction

- **Class Probabilities**: The model outputs probabilities for each class, and the class with the highest probability is selected as the prediction.

- **Thresholds**: Sometimes, decision thresholds are adjusted to balance between precision and recall.

III.6.3.7 Example Use Cases

- **Image Classification**: It analyzes and categorizes an image by comparing it to preexisting training data of labeled images. For instance, it can classify images of animals and label them as dogs, cats, or frogs.
- **Text Classification**: Assigning categories to text data, such as spam detection or sentiment analysis.
- **Medical Diagnosis**: Classifying medical images into different disease categories.

III.6.3.8 Advantages

- **Flexibility**: Can handle a wide range of classification problems.
- **Scalability**: Suitable for large and complex datasets.
- **Accuracy**: Often provides high accuracy with proper tuning and data preprocessing.

III.6.3.9 Challenges

- **Computational Complexity**: Training can be computationally intensive, especially for deep learning models.
- **Data Requirements**: Requires a large amount of labeled data for training.

- **Interpretability**: Complex models like neural networks can be difficult to interpret.

Multi-class classification algorithms are essential in many applications across various fields, making them a fundamental component of modern machine learning and artificial intelligence systems.

III.6.4 Ultra-Wideband Positioning Method

Ultra-wideband (UWB) positioning is a technique used for accurately determining the location of objects or devices within a defined space. It operates by measuring the time it takes for a signal to travel between UWB devices, typically referred to as nodes, and then using these time-of-flight measurements to calculate distances.

III.6.4.1 Principles of UWB Positioning

- **Time of Flight (ToF) Measurement**:
  - UWB devices exchange short pulses of radio waves that have extremely narrow pulses (hence the term ”ultra-wideband”).
  - These pulses are transmitted at very precise intervals and are measured with high accuracy.
  - By measuring the time it takes for these pulses to travel between devices, the distance between them can be calculated. This is known as time-of-flight (ToF) measurement.
- **Distance Calculation**:

  - To determine the distance between two UWB devices, the ToF measurement is multiplied by the speed of light (approximately $299,792,458$ m/s) and then divided by two.
  - This calculation accounts for the time it takes for the signal to travel from one device to the other and back.

- **Trilateration and Triangulation**:
  - Once distances between devices are known, positioning is achieved using trilateration or triangulation techniques:
  - **Trilateration**: Involves using distance measurements from at least three fixed reference points (known as anchors) to determine the position of a fourth point (the tag).
  - **Triangulation**: Uses angles, measured between at least two reference points and the unknown point, to calculate the position.

- **Accuracy and Precision**:
  - UWB positioning systems are known for their high accuracy and precision, often achieving sub-meter or even centimeter-level accuracy in optimal conditions.
  - This high level of precision makes UWB positioning ideal for applications such as indoor navigation, asset tracking, and industrial automation.

III.7 UML Diagrams

This section presents detailed UML diagrams to comprehensively understand the proposed system.

III.7.1 Use case Diagram

This figure presents a use case diagram to outline the key functionalities and interactions of the system.

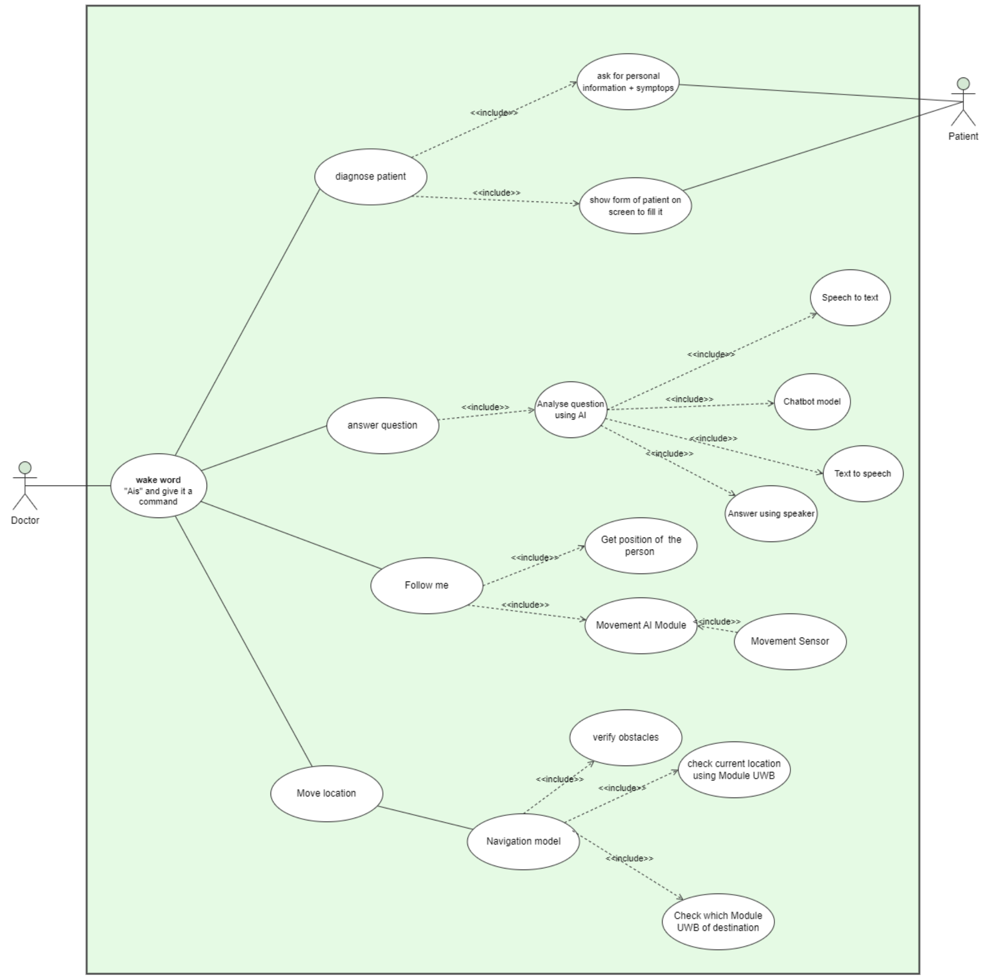

Figure III.6: Use Case Diagram.

III.7.2 Sequence Diagram

In this section, a sequence diagram is exhibited to provide a clear and concise illustration of the system's execution flow and the collaboration between components that was divided into 3 sub-diagrams.

- Figure III.7 shows the first sub-diagram that illustrates the process of asking the robot a question.

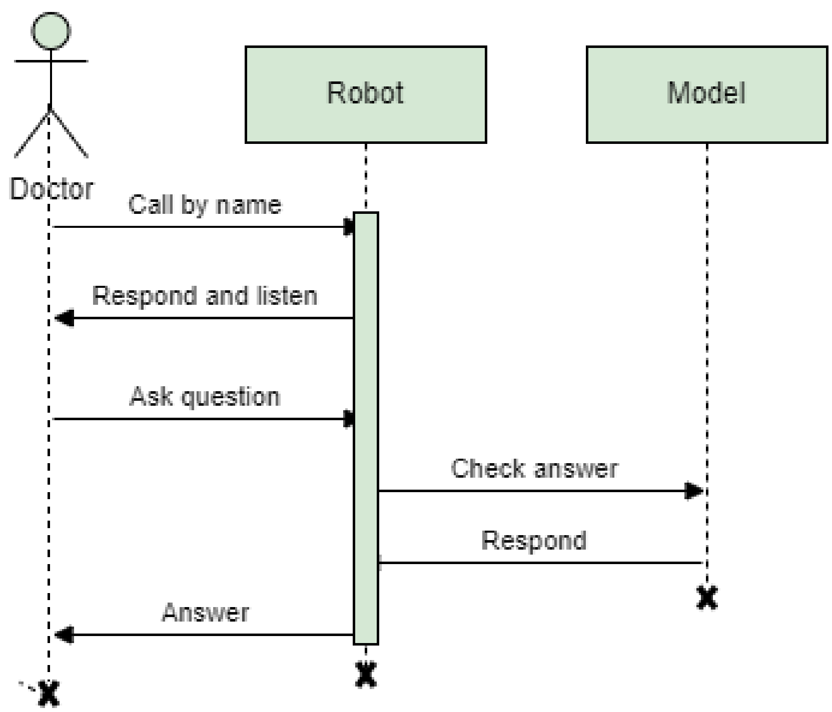

Figure III.7: Answer a question Diagram.

- Figure III.8 shows the first sub-diagram that illustrates the interaction with a patient.

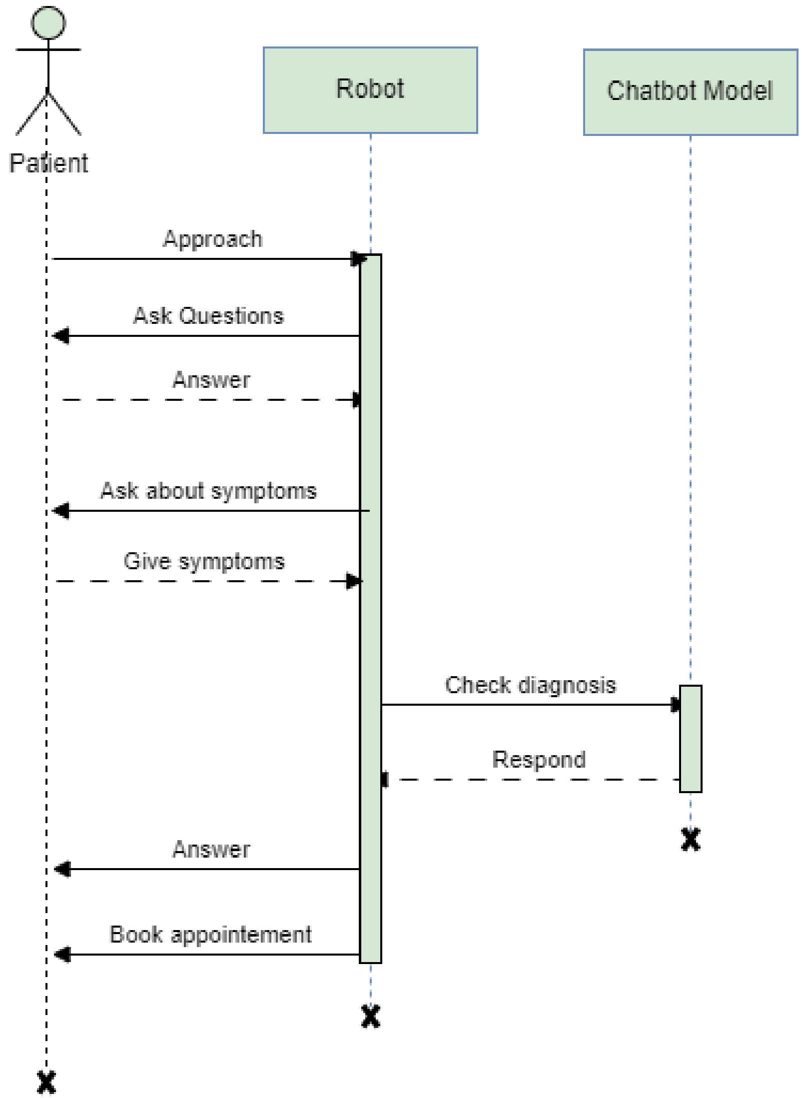

Figure III.8: Patient Diagnosis Sequence Diagram.

- Figure III.9 shows the first sub-diagram that illustrates the process of commanding the robot to go to a specific location.

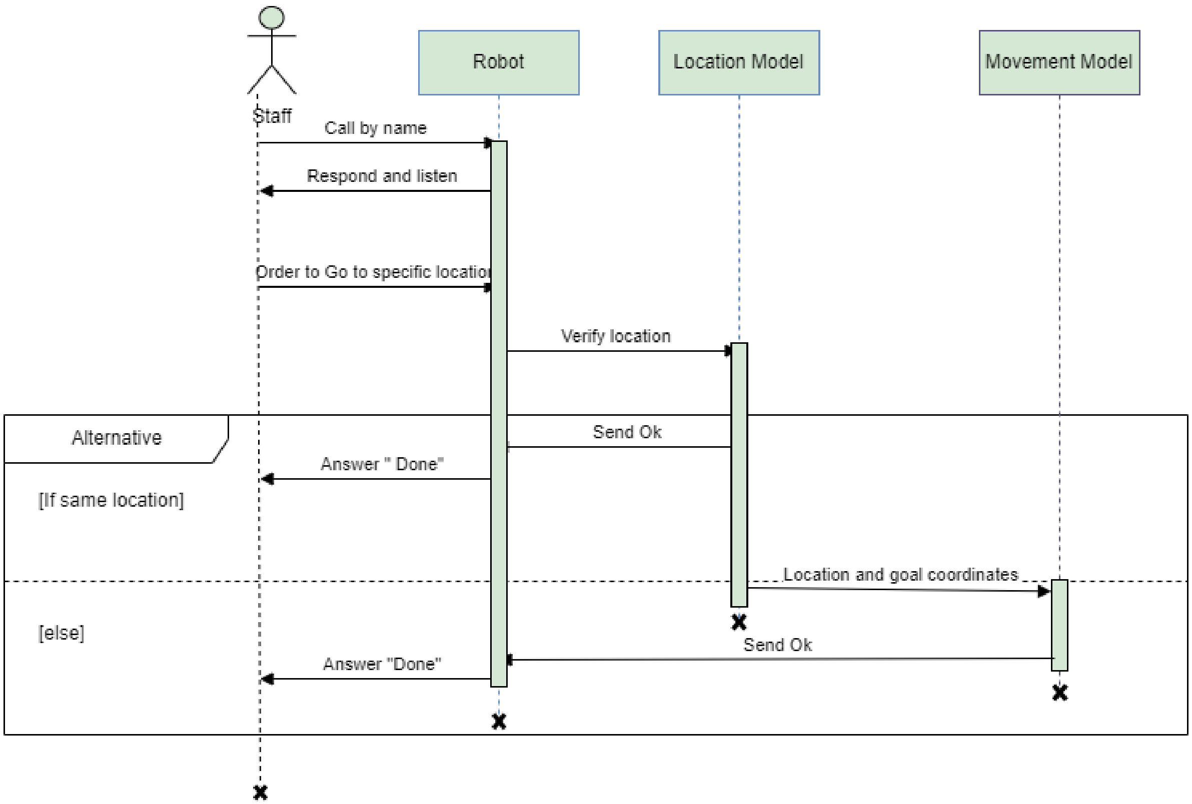

Figure III.9: Robot Movement Sequence Diagram.

III.8 Conclusion

In this chapter, we presented the general and detailed system architectures of our project and explained each of them separately. In addition, we have also explained in detail the algorithms used at each stage of the detailed architecture of the system. In the next chapter, we'll explain exactly what we did at each step.

Chapter IV

System Implementation

IV.1 Introduction

Building and assembling a physical robot is an exciting, complex, and important process that involves combining mechanical, electrical, and software components to create a machine capable of performing specified tasks. This chapter will overview the basic steps to transform a robot from a concept to a living reality.

IV.2 Environment of development

This section will explain the basic techniques for developing, building, and assembling the physical robot.

IV.2.1 Programming languages and frameworks

In this section, we will talk about the software used in our project like programming languages, and libraries.

IV.2.1.1 Python language

Figure IV.1: Logo Python

Python is the most popular open-source programming language among developers. It is a versatile object-oriented programming (OOP) language that is easy to learn and is widely used across various fields[28].

IV.2.1.2 C language

Figure IV.2: Logo C

C is a widely used open-source programming language favored by developers. It is a powerful and efficient language that forms the basis for many other programming languages. Its versatility makes it suitable for a wide range of applications [31].

IV.2.1.3 Numpy

Figure IV.3: Logo Numpy

NumPy is a fundamental package for scientific computing in Python. It supports large, multi-dimensional arrays and matrices and a collection of mathematical functions to operate on these arrays. NumPy is widely used in the scientific and engineering communities for tasks such as data manipulation, numerical computing, and data analysis[23].

IV.2.1.4 Tensorflow

Figure IV.4: Logo Tensorflow

TensorFlow is an open-source ML framework developed by Google. It provides a comprehensive ecosystem of tools, libraries, and community resources, allowing researchers and developers to build and deploy ML models effectively. TensorFlow supports various platforms, including desktops, servers, and mobile devices, making it widely used in ML and AI[1].

IV.2.1.5 Matplotlib

Figure IV.5: Logo Matplotlib

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. It is widely used for creating plots and charts to visualize data clearly and concisely. Matplotlib is highly customizable, allowing users to create a wide range of plots, including line plots, bar charts, histograms, scatter plots, and more. It is often used in conjunction with other libraries such as NumPy and pandas for data manipulation and analysis[27].

IV.2.2 Development Tools

In this section, We have used an extensive range of development tools and environments in our project. Here are the main tools and environments we used.

IV.2.2.1 Anaconda Environment

Figure IV.6: Logo Anaconda

Anaconda Environment is a feature of the Anaconda distribution that allows you to create isolated environments for your Python projects. These environments can have their own set of packages and Python versions, separate from the global Anaconda installation. This helps in managing dependencies and avoiding conflicts between different projects[2].

IV.2.2.2 Spyder

Figure IV.7: Logo Spyder

Spyder is an open-source Integrated Development Environment (IDE) for scientific programming in Python. It offers advanced editing, debugging, and profiling functionalities along with an interactive console and data exploration tools. Spyder is widely used by data scientists and engineers for its simplicity and powerful features tailored for scientific computing[55].

IV.2.2.3 Jupyter

Figure IV.8: Logo Jupyter

Jupyter is an open-source project that provides interactive notebooks for creating and sharing documents that contain live code, equations, visualizations, and narrative text. It supports multiple programming languages, including Python, R, and Julia, and is widely used in data science, scientific computing, and ML for its ability to integrate code execution with rich media output [33].

IV.2.2.4 ROS

Figure IV.9: Logo ROS

ROS is an open-source framework for writing robot software. It provides a collection of tools, libraries, and conventions to simplify the task of creating complex and robust robot behavior across various robotic platforms. ROS is widely used in academic research and industry to develop advanced robotic applications[53].

IV.2.2.5 Arduino IDE

Figure IV.10: Logo Arduino IDE

The Arduino IDE (Integrated Development Environment) is an open-source software platform used for writing, compiling, and uploading code to Arduino-compatible boards. It provides an easy-to-use interface for developing and debugging code, making it popular among hobbyists, educators, and professionals for creating interactive electronic projects[18].

IV.2.2.6 Fritzing

Figure IV.11: Logo Fritzing

Fritzing is an open-source hardware initiative that makes electronics accessible as creative material for anyone. It includes a software tool that allows users to design circuits, create schematics, and produce PCB layouts. Fritzing is widely used by hobbyists, educators, and professionals for prototyping and documenting electronics projects[34].

IV.3 Electronic Schema

In this section, we explain the steps we took to prepare the robot from choosing the electronic devices to connecting them as shown in figure IV.15.

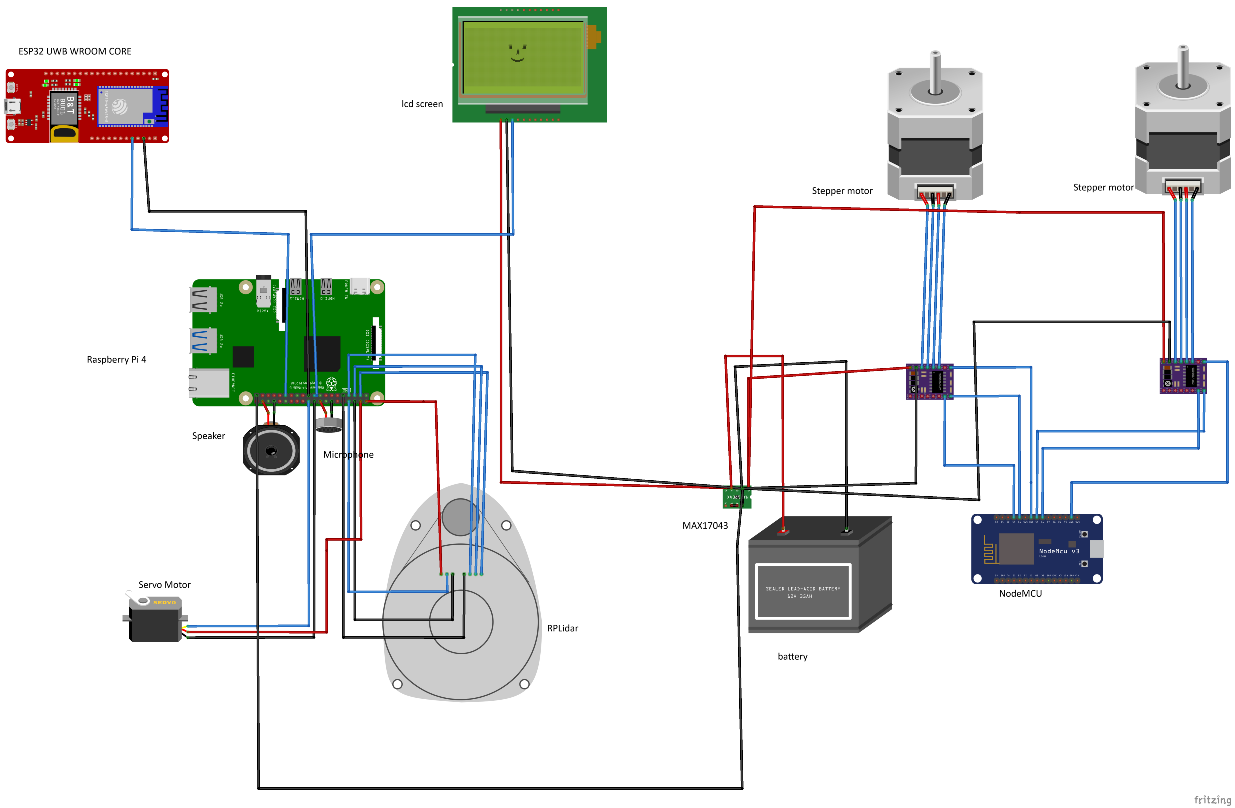

Figure IV.12: Electronic Schema

IV.3.1 Hardware Presentation

In this section, we present the process of assembling the hardware components of the robot. Assembling the hardware involves selecting and integrating various electronic and mechanical parts to create a functional platform for our project, which is:

IV.3.2 NodeMCU ESP8266

Figure IV.13: NodeMCU ESP8266

NodeMCU ESP8266 is an open-source development board and firmware based on the ESP8266 Wi-Fi module. It integrates a microcontroller with Wi-Fi capabilities, making it ideal for IoT projects. The NodeMCU firmware provides a platform for building applications with Lua scripting, while the hardware is compatible with the Arduino IDE for easy programming and prototyping.

IV.3.3 ESP32 UWB WROOM CORE

Figure IV.14: Enter Caption

The ESP32 UWB WROOM CORE is a development board integrating Ultra-Wideband (UWB) technology with the ESP32 microcontroller. This combination allows for precise distance measurement and positioning capabilities, making it suitable for applications in indoor navigation, asset tracking, and proximity detection. The board features Wi-Fi, Bluetooth, and UWB capabilities, providing versatile connectivity options for various IoT projects.

IV.3.4 Servo Motor

Figure IV.15: Servo Motor

The NodeMCU ESP8266 is an open-source development board and firmware built around the ESP8266 Wi-Fi module. This board combines a microcontroller with Wi-Fi capabilities, making it perfect for IoT projects. The NodeMCU firmware supports application development using Lua scripting, and the hardware is also compatible with the Arduino IDE, facilitating easy programming and prototyping.

IV.3.5 Raspberry pi 4

Figure IV.16: Raspberry pi 4

The Raspberry Pi 4 is a single-board computer developed by the Raspberry Pi Foundation. It offers significant improvements over its predecessors, including increased performance, more RAM options, multiple USB ports, dual HDMI outputs, and true Gigabit Ethernet. It is widely used for various applications, including IoT projects, home automation, media centers, and as a desktop replacement for basic computing tasks.

IV.3.6 RPLidar A1

Figure IV.17: RPLidar A1

The RPLidar A1 is an affordable 2D laser scanner created by SLAMTEC. It utilizes a laser triangulation measurement system to produce 2D point cloud data of its environment. The device emits a low-power infrared laser pulse (¡5mW) and calculates distances by measuring the time the reflected light returns. By rotating the laser beam, the A1 can scan a complete 360-degree horizontal plane around the sensor.

IV.3.7 MAX17043/44 IC

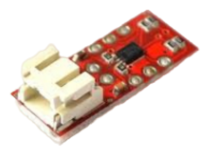

Figure IV.18: MAX17043/44 IC

The MAX17043/MAX17044 are the host-side fuel-gauge systems for lithium-ion (Li+) batteries in handheld and portable devices that are ultra-compact and low-cost. The MAX17043 is set up to work with one lithium cell, whereas the MAX17044 is set up to work with a two-cell pack [52].

IV.3.8 NEMA 17 stepper motor

Figure IV.19: NEMA 17 stepper motor

It is a popular type of electric motor known for its standardized dimensions defined by the National Electrical Manufacturers Association (NEMA). It breaks down a full rotation into discrete steps, allowing for precise control of the motor's angular position. Its features allow easy integration into various mechanical systems. These motors are widely used in 3D printers, Pick and place machines, robotics, and other motion control applications. Their versatility, reliability, affordability, and availability make them a good choice for various hobbyist and industrial applications.

IV.3.9 Hardware Installation

This section presents and goes into depth the hardware installation and sensors used in the system. In the basic components to achieve the project goal. The robot consists of three main units:

- Unit 1: Includes a Raspberry Pi 4 to which a microphone, speaker, radar, and screen are connected. The radar is strategically placed above the robot and below it is the screen.

- Unit 2: Includes a NodeMcu that connects two stepper motors at the back of the two wheels.
- Unit 3: It includes a unit ESP32 UWB WROOM CORE to determine the location.

Figure IV.20: our Robot

IV.3.10 Source Code Implementation

This section showcases the code used for developing and programming the robot.

main code

The primary code involves detecting voice activity by identifying human speech within an audio signal to activate the robot. Activation occurs when the user says "vision," triggering immediate responsiveness. After that, the robot executes one of the specified commands, which include "Follow me," "Follow the site," "Question," and "Ask the patient" in French.

IV.3.10.1 Environment Perception

Dataset

To create a high-performance and accurate chatbot model, we need a comprehensive and varied dataset. That's why we developed "Be-Rady," a website designed to collect and present data from various exams and medical courses.

Figure IV.21: Be-rady

Our primary goal was to gather multiple-choice questions (MCQs) from different educational sources, including college exams and medical books. Through the website, users could input and access a wide range of MCQs along with the correct answers, and comments if necessary.

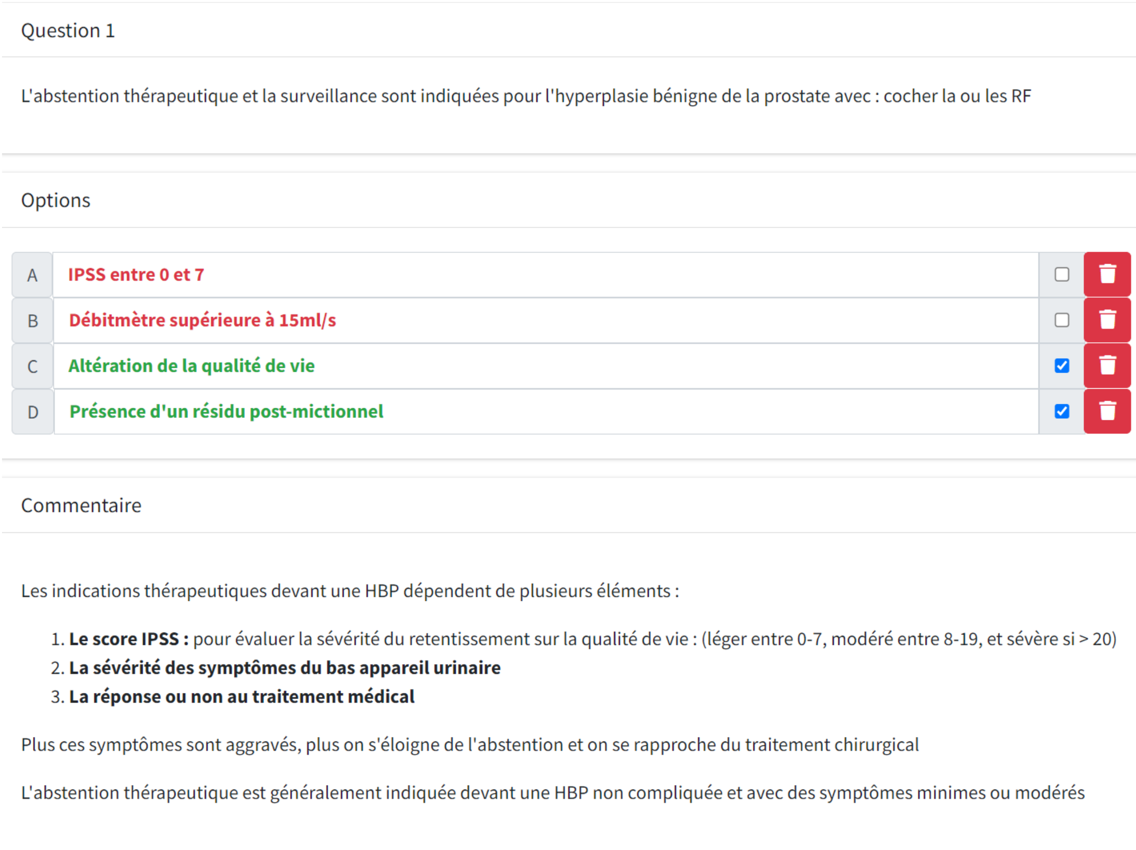

Figure IV.22: Question example structure in Be-rady website

After collecting the data, we transferred the data structure (as it shows in Figure IV.23) into the appropriate format (Figure IV.24) to allow seamless integration into the chatbot model.

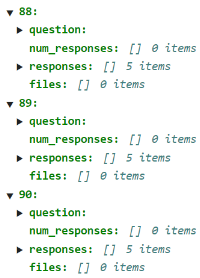

Figure IV.23: data structure

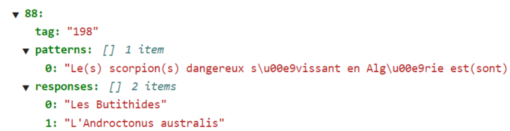

Figure IV.24: new data structure

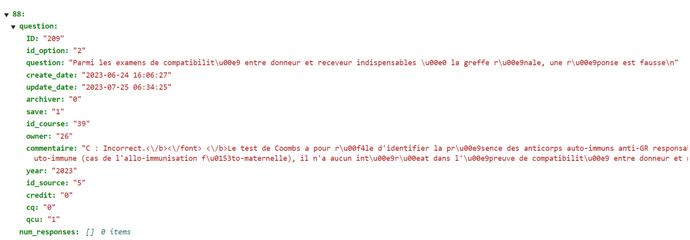

Figure IV.25: questions structure

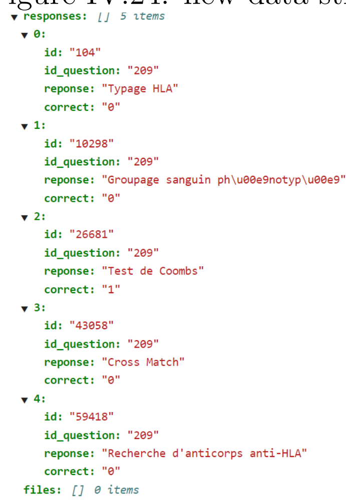

Figure IV.26: choices structure

Chatbot model

Figure IV.1 shows the hyper-parameters of the chatbot model:

Parameter	**Value**
Input Data	Preprocessed text data
Vocabulary Size (words)	`len(words)` (unique words)
Number of Classes (intents)	`len(classes)`
Input Shape	`(len(trainX[0]),)`
Output Shape	`(len(trainY[0]),)`
Hidden Layers	2
Neurons in Hidden Layer 1	128
Activation Function (Layer 1)	ReLU
Dropout Rate (Layer 1)	0.5
Neurons in Hidden Layer 2	64
Activation Function (Layer 2)	ReLU
Dropout Rate (Layer 2)	0.5
Output Layer Activation	Softmax
Loss Function	Categorical Cross-Entropy
Optimizer	SGD (learning rate=0.01, momentum=0.9, nesterov=True)
Metrics	Accuracy
Batch Size	32
Epochs	2000
Callbacks	TensorBoard
Training Data Size	`trainX.shape[0]` samples
Validation Data	Not specified

Table IV.1: Parameters and values of the chatbot model

With the parameters presented in the table IV.1 above, our model gave the following results. Figure IV.27 shows the accuracy of our model which kept increasing till it reached 0.93. As for the loss, Figure IV.28 shows it decreasing and the last value is 0.06. It is calculated by :

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} \quad \text{(IV.1)}$$

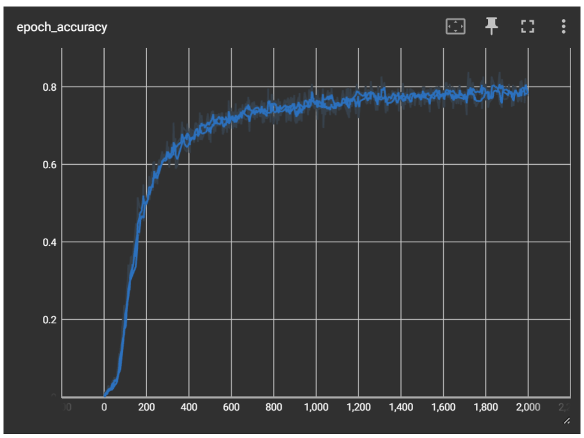

Figure IV.27: accuracy plot

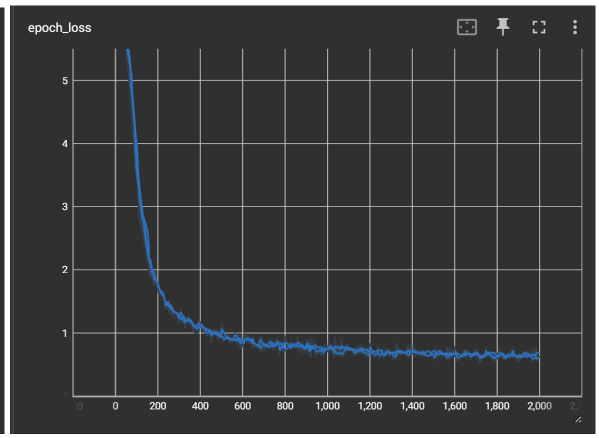

Figure IV.28: loss plot

IV.3.10.2 Mapping

To collect data from the environment, we used a 2D laser RPLidar A1. For collecting 3D data, we designed and 3D-printed a custom mount that allows the device to rotate to different angles using a servo motor. The servo motor is controlled programmatically, providing detailed adjustments and better environmental scanning. This setup improves our data collection capabilities by providing detailed 3D mappings, which are important for our application's spatial analysis and navigation functions.

Using the Lidar we were able to generate a map for the environment. In figure

IV.29, we have a 2D map in a 2D graph. And in figure IV.30 there is another 2D map in a 3D graph.

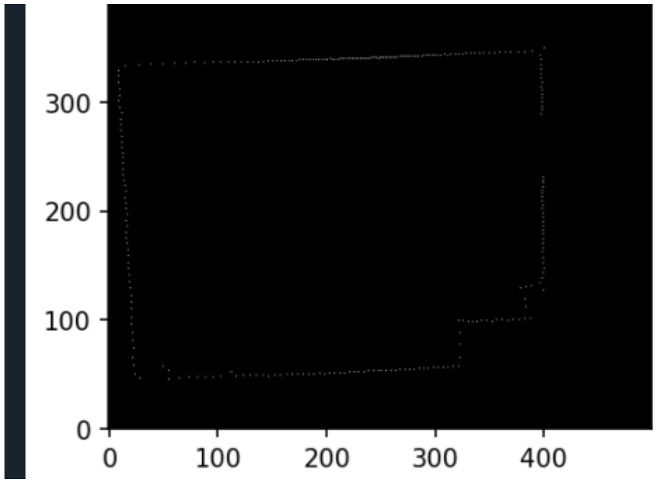

Figure IV.29: 2D map in 2D graph

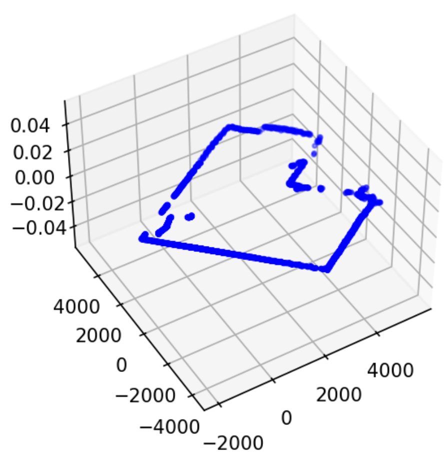

Figure IV.30: 2D map in 3D graph

Figure IV.31: RPLidarsupport

This is the 3D design of the Lidar mount:

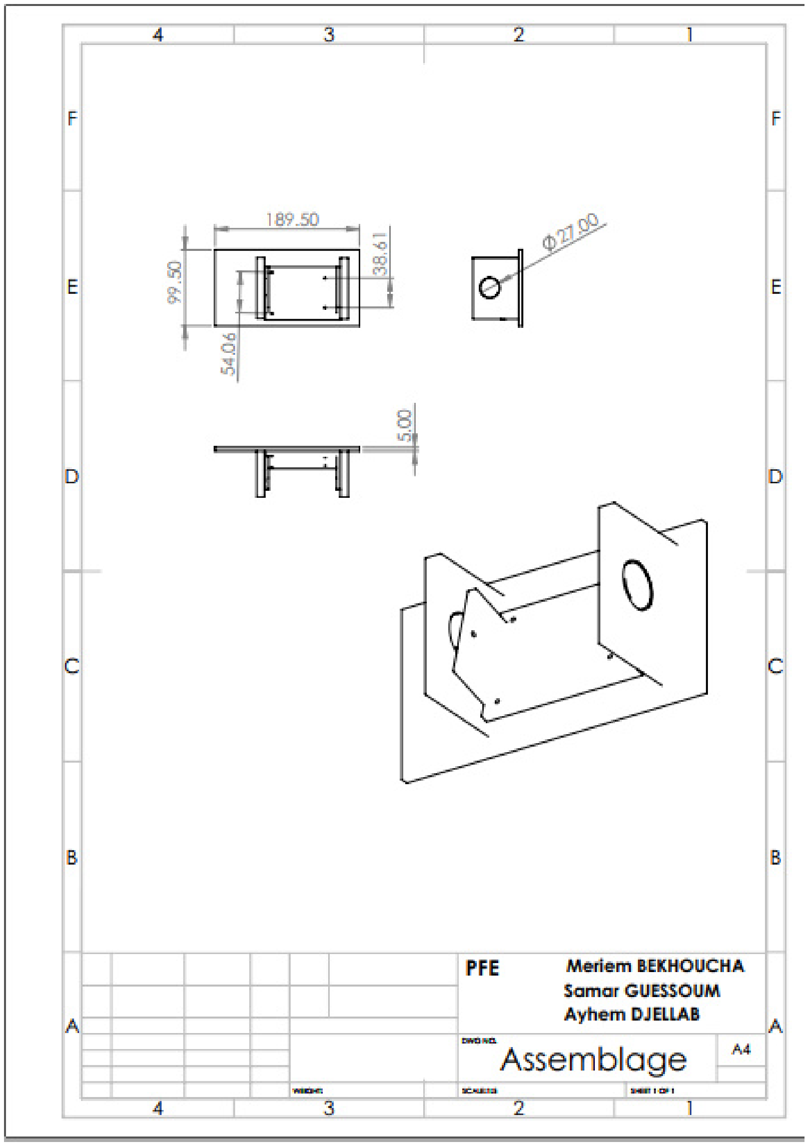

Figure IV.32: Design of Lidar Mount

This figure shows a 3D map using views(2D maps) from different angles.

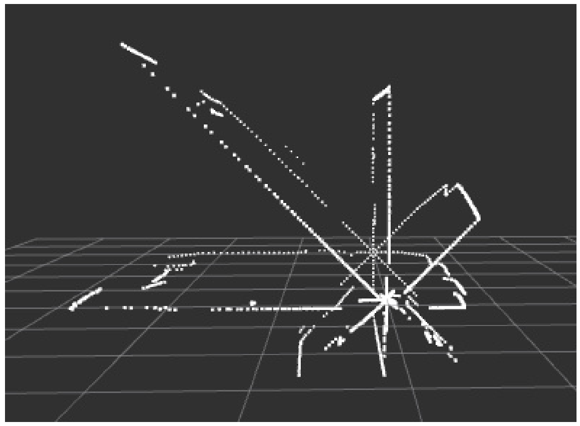

Figure IV.33: 3D map

IV.3.10.3 Navigation

To calculate the total reward of a model in reinforcement learning, start by defining the environment and identifying the reward signal the environment provides after each action taken by the agent. Train your model by running N episodes, where the agent interacts with the environment from an initial state until it reaches a terminal state. During each episode i, collect the rewards $r_t^{(i)}$ received at each time step t. Sum these rewards within an episode to obtain the total reward for that specific episode, given by $R_i = \sum_{t=0}^{T_i} r_t^{(i)}$, where T_i is the total number of time steps in episode i. To calculate the cumulative total reward across multiple episodes, sum the total rewards of each episode, given by $R_{\text{total}} = \sum_{i=1}^{N} R_i = \sum_{i=1}^{N} \sum_{t=0}^{T_i} r_t^{(i)}$. This approach helps evaluate the performance of the model over the training period by quantifying the cumulative rewards obtained. This is the Total Reward Graph of model navigation:

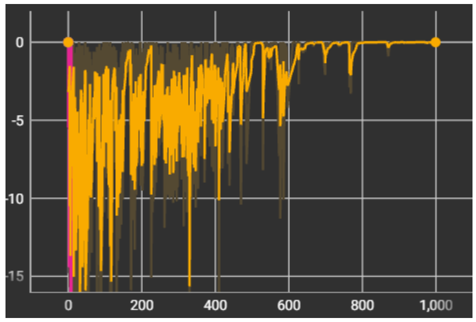

Figure IV.34: Total Reward

This Hyper-parameter of mode Navigation Ai:

Parameter	**Description**	**Value(s)**
`start_position`	Tuple defining the starting position of the agent.	Randomly chosen within (0-2, 0-5)
`obstacles`	List of tuples defining positions of obstacles.	[(1, 1)]
`optimizer`	Adam optimizer with a learning rate of 0.0001.	Adam(learning_rate=0.0001)
`gamma`	Discount factor (0.99).	0.99
`action_space`	List of strings representing available actions.	['up', 'down', 'left', 'right']
`states`	Tensor holding states used for training.	Randomly generated uniform tensor of shape (1, 30, 30)
`actions`	Tensor holding actions taken during training.	[[0]]
`rewards`	List holding rewards received during training.	[10.0]
`max_steps_per_episode`	Maximum number of steps per episode during training.	20
`batch_size`	Number of episodes collected before each training step.	16
`num_episodes`	Total number of episodes for training.	1000

Table IV.2: policynetwork parameters

IV.4 Conclusion

In conclusion, this chapter has given the knowledge of the essential tools, software, and hardware used to construct our system. It has provided a clear view of the code developed for each stage of the system's architecture, along with the methods for connecting electronic devices to build an autonomous robot model and defining each component individually. Finally, it has presented a glimpse of the successful outcomes achieved through the development process.

General Conclusions

In conclusion, this report comprehensively explored the integration of conversational AI into a bodily autonomy system for medical applications, covering four main chapters. The fundamental concepts of artificial intelligence (AI), the Internet of Things (IoT), and healthcare are discussed, highlighting their pivotal roles in reshaping healthcare systems. The overview, especially in medical contexts, has emphasized the advantages of integrating artificial intelligence with robotics to enhance the efficiency and accuracy of medical tasks. The concept of the system is detailed, outlining its architecture, components, and user interaction, providing a blueprint for effectively integrating conversational AI with autonomous mobility. The implementation phase was discussed, covering technical aspects, challenges encountered, and solutions implemented, culminating in a system that deftly interacts with patients and navigates medical environments autonomously.

The project achieved its goals, producing a functional system that enhances patient interaction and ensures reliable mobility. Future work includes developing natural language processing capabilities, integration with electronic health records, enhancing mobility and safety features, and ensuring scalability and deployment in diverse medical environments. The impact of integration on healthcare is profound, promising to improve patient care, increase efficiency and data-driven insights, and enhance access to healthcare services.

Overall, this project signals a significant advance in healthcare technology, showcasing the potential for smarter, more efficient, and patient-focused healthcare solutions.

Bibliography

[1] TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] Anaconda software distribution, 2020.

[3] Md Eshrat E. Alahi, Arsanchai Sukkuea, Fahmida Wazed Tina, Anindya Nag, Wattanapong Kurdthongmee, Korakot Suwannarat, and Subhas Chandra Mukhopadhyay. Integration of iot-enabled technologies and artificial intelligence (ai) for smart city scenario: Recent advancements and future trends. *Sensors*, 23(11), 2023.

[4] Mouadh Bali, Abdelkamel Tari, Abdullah Almutawakel, and Okba Kazar. Smart design for resources allocation in iot application service based on multi-agent system and dcsp. *Informatica*, 44:373–386, 10 2020.

[5] Belen Bermejo and Carlos Juiz. Improving cloud/edge sustainability through artificial intelligence: A systematic review. *Journal of Parallel and Distributed Computing*, 176:41–54, 2023.

[6] David Berry. The Limits of Computation: Joseph Weizenbaum and the ELIZA Chatbot. 11 2023.

[7] Olfa Boubaker. Chapter 7 - medical robotics. In Olfa Boubaker, editor, *Control Theory in Biomedical Engineering*, pages 153–204. Academic Press, 2020.

[8] Murray Campbell, A.Joseph Hoane, and Feng hsiung Hsu. Deep blue. *Artificial Intelligence*, 134(1):57–83, 2002.

[9] Yen-Kuang Chen. Challenges and opportunities of internet of things. pages 383–388, 01 2012.

[10] Yi Cheng and Gong Ye Wang. Mobile robot navigation based on lidar. In *2018 Chinese Control And Decision Conference (CCDC)*, pages 1243–1246, 2018.

[11] Ltd Co. *Artificial Intelligence Technology*. 01 2023.

[12] Christopher Collins, Denis Dennehy, Kieran Conboy, and Patrick Mikalef. Artificial intelligence in information systems research: A systematic literature review and research agenda. *International Journal of Information Management*, 60:102383, 2021.

[13] Edsger W Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.

[14] Saharnaz Dilmaghani, Matthias R. Brust, Grégoire Danoy, Natalia Cassagnes, Johnatan Pecero, and Pascal Bouvry. Privacy and security of big data in ai systems: A research and standards perspective. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 5737–5743, 2019.

[15] Hugh Durrant-Whyte and Tim Bailey. Simultaneous localization and mapping: Part i. *IEEE robotics & automation magazine*, 13(2):99–110, 2006.

[16] Oussama El Hamzaoui. Simultaneous localization and mapping for a mobile robot with a laser scanner : Coreslam. 09 2012.

[17] Emilio Ferrara. Fairness and bias in artificial intelligence: A brief survey of sources, impacts, and mitigation strategies. *Sci*, 6(1), 2024.

[18] Mohamed Fezari and Ali Al Dahoud. Integrated development environment ”ide” for arduino. 10 2018.

[19] Alessandro Giusti, Jerome Guzzi, Dan Ciresan, Fang-Lin He, Juan Pablo Rodriguez, Federico Fontana, Matthias Faessler, Christian Forster, Juergen Schmidhuber, and Gianni Di Caro. Machine learning for autonomous navigation of uavs with onboard sensors. *IEEE transactions on neural networks and learning systems*, 29(3):749–761, 2016.

[20] Giorgio Grisetti, Rainer Kummerle, Cyrill Stachniss, and Wolfram Burgard. A tutorial on graph-based slam. *IEEE Intelligent Transportation Systems Magazine*, 2(4):31–43, 2010.

[21] Walid Guettala, Ali Sayah, Laid Kahloul, and Ahmed Tibermacine. Real time human detection by unmanned aerial vehicles. In *2022 International Symposium on iNnovative Informatics of Biskra (ISNIB)*, pages 1–6. IEEE, 2022.

[22] Leo Gugerty. Newell and simon’s logic theorist: Historical background and impact on cognitive modeling. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 50:880–884, 10 2006.

[23] Charles R Harris, K Jarrod Millman, Stéfan J Van Der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebas-

tian Berg, Nathaniel J Smith, et al. Array programming with numpy. *Nature*, 585(7825):357–362, 2020.

[24] Peter E Hart, Nils J Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.

[25] H Mary Henrietta. *ARTIFICIAL INTELLIGENCE IN AGRICULTURE: A REVIEW OF CURRENT APPLICATIONS AND FUTURE TRENDS ARTIFICIAL INTELLIGENCE IN AGRICULTURE: A REVIEW OF CURRENT APPLICATIONS AND FUTURE TRENDS ARTIFICIAL INTELLIGENCE IN AGRICULTURE: A REVIEW OF CURRENT APPLICATIONS AND FUTURE TRENDS*, pages 1–5. 04 2024.

[26] Kun Mean Hou, Xunxing Diao, Hongling Shi, Hao Ding, Haiying Zhou, and Christophe de Vaulx. Trends and challenges in aiot/iiot/iot implementation. *Sensors*, 23(11), 2023.

[27] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.

[28] Aaquib Javed, Monika Zaman, M Monir Uddin, and Tasnova Nusrat. An analysis of python programming language demand and its recent trend in bangladesh. In *Proceedings of the 2019 8th International Conference on Computing and Pattern Recognition*, pages 458–465, 2019.

[29] Lu Jia, Zhengguang Ma, and Yongguo Zhao. A mobile robot mapping method integrating lidar and depth camera. *Journal of Physics: Conference Series*, 2402:012031, 12 2022.

[30] Xueyang Kang, Shengjiong Yin, and Yinglong Fen. 3d reconstruction assessment framework based on affordable 2d lidar. In *2018 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 292–297, 2018.

[31] Brian W. Kernighan and Dennis M. Ritchie. *The C Programming Language*. Prentice Hall, 1988.

[32] John Kerr and Kevin Nickels. Robot operating systems: Bridging the gap between human and robot. In *Proceedings of the 2012 44th Southeastern Symposium on System Theory (SSST)*, pages 99–104, 2012.

[33] Thomas Kluyver, Benjamin Ragan-Kelley, Fernando Pérez, Brian Granger, Matthias Bussonnier, Jonathan Frederic, Kyle Kelley, Jessica Hamrick, Jason Grout, Sylvain Corlay, Paul Ivanov, Damián Avila, Safia Abdalla, and Carol Willing. Jupyter notebooks – a publishing format for reproducible computational workflows. pages 87 – 90. IOS Press, 2016.

[34] André Knörig, Reto Wettach, and Jonathan Cohen. Fritzing: a tool for advancing electronic prototyping for designers. pages 351–358, 02 2009.

[35] Chayakrit Krittanawong, Albert Rogers, Kipp Johnson, Zhen Wang, Mintu Turakhia, Jonathan Halperin, and Sanjiv Narayan. Integration of novel monitoring devices with machine learning technology for scalable cardiovascular management. *Nature Reviews Cardiology*, 18, 12 2020.

[36] Andrew Kusiak. Smart manufacturing. *International Journal of Production Research*, 56(1-2):508–517, 2018.

[37] Steven M LaValle. Rapidly-exploring random trees: A new tool for path planning. In *Technical Report*. Iowa State University, 1998.

[38] sheik dawood M. Review on applications of internet of things (iot). 12 2018.

[39] Tao Mei, Minzhou Luo, Xiaodong Ye, Jun Cheng, Lan Wang, Bin Kong, and Rujin Wang. *Design and Implementation of a Service Robot for Elders*, pages 83–93. 12 2015.

[40] Giorgio Metta, Paul Fitzpatrick, and Lorenzo Natale. Yarp: Yet another robot platform. *International Journal of Advanced Robotic Systems*, 3, 03 2006.

[41] Brent Mittelstadt. 378Interpretability and Transparency in Artificial Intelligence. In *Oxford Handbook of Digital Ethics*. Oxford University Press, 12 2023.

[42] Tanvir Moin. Overview of computer vision, 06 2023.

[43] Raul Mur-Artal, JMM Montiel, and JD Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015.

[44] Ilyes Naidji, Ahmed Tibermacine, Walid Guettala, and Imad Eddine Tibermacine. Semi-mind controlled robots based on reinforcement learning for indoor application. In *ICYRIME*, pages 51–59, 2023.

[45] Andy Nguyen, Mario Kremantzis, Aniekan Essien, Ilias Petrounias, and Samira Hosseini. Enhancing student engagement through artificial intelligence (ai): Understanding the basics, opportunities, and challenges. *Journal of University Teaching and Learning Practice*, 21(06), 2024.

[46] Luminița Nicolescu and Monica Teodora Tudorache. Human-computer interaction in customer service: The experience with ai chatbots—a systematic literature review. *Electronics*, 11(10), 2022.

[47] Charaf Ouaddi, Lamya Benaddi, and Abdeslam Jakimi. Architecture, tools, and dsls for developing conversational agents: An overview. *Procedia Computer Science*, 231:293–298, 2024.

[48] Dr. Anish Pandey. Mobile robot navigation and obstacle avoidance techniques: A review. *International Robotics Automation Journal*, 2:1–12, 05 2017.

[49] Sharnil Pandya, Aanchal Thakur, Santosh Saxena, Nandita Jassal, Chirag Patel, Kirit Modi, Pooja Mehta, Sudhanshu Gonge, Kalyani Kadam, and Prachi Kadam. A study of the recent trends of ai in healthcare, immunology: Key challenges, domains, applications, datasets, and future directions. *Sensors*, 11 2021.

[50] Keyur Patel, Sunil Patel, P Scholar, and Carlos Salazar. Internet of things-iot: Definition, characteristics, architecture, enabling technologies, application future challenges, 05 2016.

[51] Anil Pise, Khalid Almuzaini, Tariq Ahanger, Ahmed Farouk, Kumud Pant, Piyush Pareek, and Stephen Nuagah. Enabling artificial intelligence of things (aiot) healthcare architectures and listing security issues. *Computational Intelligence and Neuroscience*, 2022:1–14, 08 2022.

[52] Mahmood H Qahtan, Emad A Mohammed, and Ahmed J Ali. Iot-based electrical vehicle's energy management and monitoring system. *Open Access Library Journal*, 9(7):1–15, 2022.

[53] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, Andrew Y Ng, et al. Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5. Kobe, Japan, 2009.

[54] Md Rahaman. A review on internet of things-iot- architecture, technologies, future applications challenges. *International Journal of Science and Business*, 14:80–92, 02 2024.

[55] Pierre Raybaut. Spyder-documentation. *Available online at: pythonhosted. org*, 2009.

[56] Erol Sahin. Swarm robotics: From sources of inspiration to domains of application. *Swarm robotics*, pages 10–20, 2005.

[57] Qusay Sarhan. Internet of things: A survey of challenges and issues. *International Journal of Internet of Things and Cyber-Assurance*, 1, 01 2018.

[58] Iqbal Sarker. Deep learning: A comprehensive overview on techniques, taxonomy, applications and research directions. *SN Computer Science*, 2, 08 2021.

[59] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[60] Jakub Semborski and Adam Idzkowski. A review on positioning techniques of mobile robots. *Robotic Systems and Applications*, February 2024.

[61] Jakub Semborski and Adam Idzkowski. A review on positioning techniques of mobile robots. *Robotic Systems and Applications*, 02 2024.

[62] Edward Shortliffe. The evolution of health-care records in the era of the internet. *Studies in health technology and informatics*, 52 Pt 1:suppl 8–14, 02 1998.

[63] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

[64] Jian Sun, Wei Sun, Xing Zhang, Jin Zheng, Chenjun Tang, and Zhongyu Chen. Multiuser smartphone-based localization by composite fingerprint descriptors in large indoor environments. *IEEE Sensors Journal*, 23(18):21882–21893, 2023.

[65] Lei Tai, Giuseppe Paolo, and Ming Liu. Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation. *arXiv preprint arXiv:1703.00420*, 2017.

[66] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic robotics*. MIT press, 2005.

[67] Sebastian Thrun et al. Robotic mapping: A survey. 2002.

[68] Spyros G Tzafestas. Synergy of iot and ai in modern society: The robotics and automation case. *Robot. Autom. Eng. J*, 31:1–15, 2018.

[69] Athanasios Valavanidis. Artificial intelligence (ai) applications. he most important technology we ever develop and we must ensure it is safe and beneficial to human civilization i. 1:1–49, 04 2023.

[70] Jur Van Den Berg, Ming Lin, and Dinesh Manocha. Reciprocal n-body collision avoidance. *Robotics research*, pages 3–19, 2008.

[71] Tianmiao Wang, Long Yang, Ruoyu Yang, Xianmin Zhao, and Jianwei Zhang. A review of multi-robot manipulation. *Annual Reviews in Control*, 47:35–53, 2018.

[72] Werbos. Backpropagation: Past and future. In *IEEE 1988 International Conference on Neural Networks*, pages 343–353. IEEE, 1988.

[73] Marco A Wiering and Martijn Van Otterlo. Reinforcement learning. *Adaptation, learning, and optimization*, 12(3):729, 2012.

[74] M. Wilson. Implementation of robot systems: An introduction to robotics, automation, and successful systems integration in manufacturing. pages 1–229, 01 2014.

[75] Mike Wilson. *Chapter 2. Industrial Robots*. 12 2015.

[76] Qiong Wu, Yanjun Liu, and Chensheng Wu. An overview of current situations of robot industry development. *ITM Web of Conferences*, 17:03019, 02 2018.

[77] Xiaonan Xu, Zheng Xu, Zhipeng Ling, Zhengyu Jin, and ShuQian Du. Comprehensive implementation of textcnn for enhanced collaboration between natural language processing and system recommendation. *arXiv preprint arXiv:2403.09718*, 2024.

[78] Ömer Yanık and Hakki Ilgin. A comprehensive computational cost analysis for state-of-the-art visual slam methods for autonomous mapping. *Communications Faculty of Sciences University of Ankara Series A2-A3 Physical Sciences and Engineering*, 65:1–15, 06 2023.