Cognitive Architecture for robots inspired by Quantum Computing

Yeshwanth Guru Krishnakumar

Department of Computer Science, Bioengineering, Robotics and System Engineering (DIBRIS)

University of Genova

Supervisor

Fulvio Mastrogiovanni
Alessandro Carfi

In partial fulfillment of the requirements for the degree of

Master of Science in Robotics Engineering

July 20, 2023

Acknowledgements

I would like to express my sincere gratitude and thanks to Professor Fulvio Mastrogiovanni, my supervisor and mentor, for his valuable advice, guidance, and unwavering support throughout the Master's Course and my thesis work. I am also grateful for his trust in me for this project.

Additionally, I want to extend my thanks to Professor Alessandro Carfi and Davide Lanza for their cooperation, assistance, and technical insights provided to enhance my work.

I would like to acknowledge and appreciate all the professors who have contributed to my academic journey, shaping my understanding of the subject matter and sharing their expertise.

To my friends Sanjay, Koushik, Rahul, and Nagsen I am deeply grateful for their unwavering support and encouragement, which has been a constant source of motivation for me during my international student life in Italy.

Lastly, I want to express heartfelt thanks to my dad, Mr.Krishnakumar, my mom, Mrs.Malar, and my sister Monisha for their unconditional love, support, understanding, and belief in my abilities.

I am immensely grateful to everyone who has played a part in my academic journey. Without their support, this achievement would not have been possible. And happy to be part of the Emarolab (TheEngineRoom) Family.

"Questa tesi è dedicata ai miei eroi: i miei nonni, i genitori e la famiglia del laboratorio EMARO e l'intera comunità di ingegneri. La loro incrollabile dedizione, la loro sconfinata curiosità e la ricerca senza tempo di plasmare un futuro migliore ispirano e aprono la strada a nuove possibilità. Sono profondamente grato per la loro profonda influenza e questa tesi rappresenta una testimonianza del loro prezioso contributo alla mia vita personale e professionale."

"This thesis is dedicated to my heroes—My Grandparents, parents and EMARO lab family, and the entire community of Engineers. Their unwavering dedication, boundless curiosity, and timeless pursuit of shaping a better future inspire and pave the way for new possibilities. I am profoundly grateful for their profound influence, and this thesis stands as a testament to their valuable contributions to my personal and professional life."

Abstract

Quantum robotics is the emerging sector that merges quantum physics principles with robotics, utilizing quantum computing, sensing, and control techniques to enhance robot systems. We are going to explore the field described combines quantum mechanics, cognition, and robotics and falls under the interdisciplinary fields of quantum cognition and quantum-inspired robotics. Quantum cognition explores how quantum mechanics can explain human cognition. Quantum-inspired robotics integrates quantum principles into robotic architectures. While quantum-inspired robotics does not involve actual quantum computing or quantum physical phenomena in most cases, it draws inspiration from the underlying principles of quantum mechanics to develop new algorithms, control strategies, and technologies for robotic systems

In recent years, there has been growing interest in applying quantum-inspired principles to cognitive science and robotics. This thesis explores the integration of quantum-like perception models in robotics for advancing cognitive abilities, specifically focusing on handover tasks to prove their ability. By leveraging the behavior of qubits to trigger the robot mechanism based on integrated multi-sensory information from the robot's environment, a novel cognitive architecture is proposed, enabling robots to perceive and process sensory data in a quantum-inspired manner. The results demonstrate enhanced Quantum-Inspired Decision-Making, closely resembling cognition processes observed in biological systems. Real-time deployment on a Tiago++ robot validates the practical applicability of the proposed architecture, showcasing its potential for advancing cognitive robotics for the strong belief state. By emulating biological cognition processes, these robots bridge the gap between artificial intelligence and human understanding, propelling research in both fields. This advancement brings us closer to a new era of intelligent robotics that closely resembles natural cognitive abilities.

Keywords: Quantum Mechanics, Cognitive Robotics, Quantum-Robotics, Cognitive Science, Software-Architecture for Robots.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Quantum physics and robotics are combined in the fascinating growing field of robotics. It can use quantum computing, sensing, and control techniques to enhance the capabilities of robot systems. We will research this area's subspecialty, which combines elements of quantum physics, cognition, and robotics to produce the interdisciplinary fields of quantum cognition and quantum-inspired robotics.

The field of Quantum cognition investigates how quantum mechanics may provide light on how people think. It investigates if specific instances in human judgment, problem-solving, and information processing may be accounted for by quantum principles. By applying quantum-like formalism to cognitive processes, researchers hope to better comprehend these processes. And many research has undergone on the quantum Cognition approach in recent years where some researchers compared Weak vs. strong quantum cognition[1]

Quantum-Inspired Robotics is the area of robotics where quantum mechanical ideas and principles are used. Even while most examples of quantum-inspired robotics do not include actual quantum computers or quantum physical phenomena, they are nonetheless inspired by the basic concepts of quantum mechanics in order to develop new robotic algorithms, control systems, and technologies.

If quantum-inspired concepts are used to help robots do jobs more effectively and efficiently, robotics may experience a revolution. Robotic data collection and comprehension may be improved with the use of quantum-inspired sensing techniques. By boosting the robot's ability to manipulate and interact with its environment, control systems influenced by quantum mechanics may enhance the robot's dexterity and autonomy.

Given its focus on quantum cognition and quantum-inspired robotics, quantum robotics generally has significant promise for improving robot capabilities.

A cognitive architecture for robotics is the underlying framework or structure that enables a robot to perform cognitive tasks and exhibit intelligent behavior. It outlines how to mix and organize various cognitive processes, such as perception, thought, learning, decision-making, and action, inside of a robotic system.

Interest in applying formalisms influenced by quantum mechanics to a range of fields, including cognitive science, has lately increased in engineering. Quantum theory-inspired cognitive science techniques have descriptive features that are excellent for understanding how perception, cognition, and decision-making function. These techniques provide a unique perspective on how information is handled and represented in complicated systems.

In our proposal, we emphasize a unique method for creating cognitive architecture for robots that are inspired by quantum mechanics, particularly in the context of handover duties. The suggested quantum-inspired design uses the actions of qubits, the cornerstones of quantum computation, to activate robotic systems based on the robot belief state. The design allows the robot to detect and interpret sensory input in a way that is inspired by quantum mechanics by combining multi-sensory information from the robot's surroundings and employing single-qubit measurement. This quantum-inspired robot perception paradigm developed by(Davide Lanza, Fulvio Mastrogiovanni, Paolo Solinas, [2][3]) is revolutionizing how robots perceive and interact with their surroundings where the authors suggest that simulation-oriented techniques should be preferred over actual implementations on quantum backends in order to investigate QL models for robot behavior, in our thesis work we have investigated the feasibility of the following approach in the real robot Tiago++ robot which is one of the main contributions and analyze the robot behavior in real-time.

The main contributions of this work lie in the development of a Traditional Cognitive Architecture Development with ROS-melodic for Tiago++ robot, Integration of qiskit(Qrobot package) developed Decision maker with single sensory fuzzy knowledge representation and multi-sensory fuzzy knowledge representation later followed with the analysis of single- and multi-qubit quantum systems as models for low-level perceptual data integrated into our architecture. By studying the behavior of qubits, researchers have gained a deeper understanding of how to represent and process sensory information in a quantum bit (Davide Lanza, Fulvio Mastrogiovanni, Paolo Solinas, [2][3]). The proposed architecture integrates a quantum-inspired model for decision-making as per the thesis experimental scenario, creating a comprehensive and adaptable framework for robot cognition.

Our research emphasizes the significance of multi-sensory integration from the robot world to the quantum-like perception model and gets the qubit behavior. This integration allows the robot to perceive and process information from various sensory modalities, resulting in a holistic understanding of its surroundings. The quantum state of the qubit, influenced by the integrated sensory information, guides the decision-making mechanism of the robot. And after that to prove the possibility of speeding up the Decision making by tuning the Query operator approach done in the following (Davide Lanza, Fulvio Mastrogiovanni, Paolo

Solinas[3]). This approach represents a departure from classical robot frameworks and introduces a more radical and biologically-inspired cognition process(sense-plan-act).

The results of this work demonstrate the effectiveness of the quantum-inspired cognitive architecture for robots in handover tasks where it provides a powerful framework for representing uncertainty and decision-making in a compact and efficient manner and making it possible for suitable cognitive science application. where the quantum-like perception model integrated model has the behavior to allow for the definition of query operators, which enable the inspection of any given world state. The answers provided by these query operators quantify the robot's degree of belief in the specific state being queried. Real-time deployment of the proposed architecture on a Tiago++ robot showcases its practical applicability with a handover task for the experimental proof of robot belief state based upon the adaptive decision making, but the approach logic can be used in any case for robotics application for cognitive purposes depending on its application use-case. The robots developed using this approach exhibit enhanced decision-making capabilities, closely resembling the cognition processes observed in biological systems. This advancement represents a significant step towards bridging the gap between robotics and artificial intelligence research.

The dissertation's opening chapter acts as an introduction to the remaining work.

Chapter 2, Describes Quantum robotics and cognitive robotics outline of its origin, Current state, future, and Pros and cons of the following fields.

Chapter 3, Describes the State of the art for the entire Development where the basics of Quantum computing principles can be studied and a detailed understanding of the thesis state-of-the-art work followed with the motivation.

Chapter 4, Describes the decision-making ability and its working comparison with classical bit and qubit.

Chapter 5, Describes the detailed description of the methodology usage working for the development.

Chapter 6 This chapter shows the traditional Cognitive architecture working mechanism.

Chapter 7, Quantum-inspired Cognitive Architecture description of single and multi-sensory quantum-like robot perception working.

Chapter 8, Realtime deployment of the complete architecture in Tiago ++ robot and its analysis(Experiments and results).

Chapter 9, Conclusion of the work and its future work possibilities.

Chapter 2

Robotics(Quantum Robotics - Cognitive Robotics)

2.1 Cognitive robotics

Cognitive Robotics is the field that integrates robotics, artificial intelligence, and cognitive science which has the possibility to create a robot that perceives, reason, learn and interact with its environment with human-like abilities, which draws inspiration from the cognitive sciences, which studies how human perceive, think and learn as well as from AI techniques which would enable the machine to have human-like intelligence. It has some critical components such as perception, Reasoning, Planning, learning and adaption, and Human-robot interaction [4][5][6][7][8].

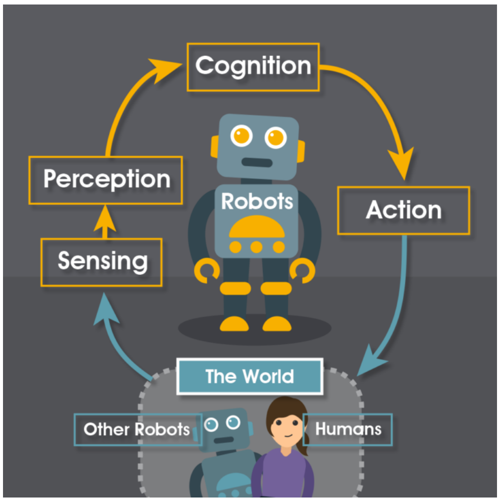


Figure 2.1: Cognitive-robotics

2.1.1 Cognitive Skills of Current Generation Robotics

Key components of cognitive robotics include the following list below:

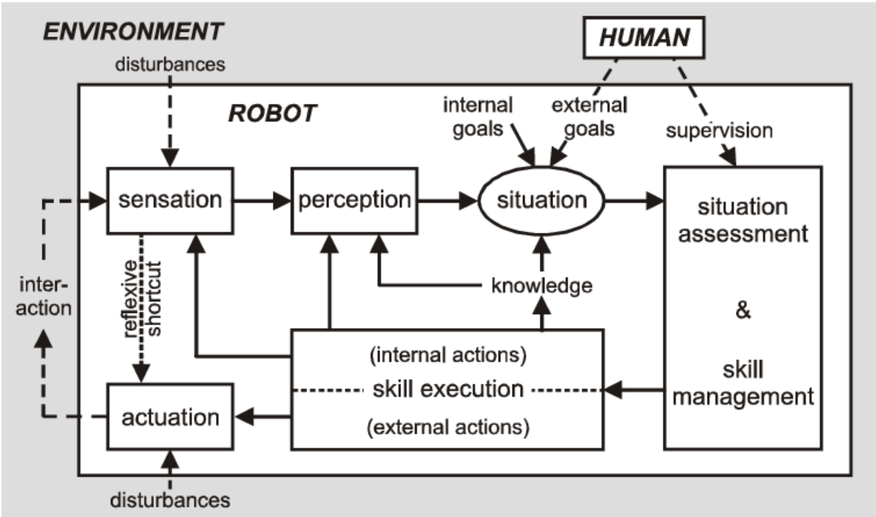


Figure 2.2: Example for Robotics Cognitive system component scenario

- Perception:

  Perception: In robotics which shows the ability of a robot to sense and use the information from its environment. In that scenario a robot involves the ability to perceive and understand its surrounding using various sensors which list such as Camera, LIDAR also called Light Detection and Ranging), Ultrasound, and tactile sensors. Meanwhile it a crucial for robots to interact with and navigate through their environment in an effective manner. which can be seen based on the key aspect of its ability description below.

  1. Sensing: Basically the robots use the different types of sensors to understand their environment. Vision sensors perceive their environment through their visuals, including shape, colors, and texture. Lidar sensors which are used use laser light to measure the distance and to create the 3d maps of their environment. Ultrasound and radars were used to measure distances and obstacle detection. Tactile is used to perceive information about the physical properties of the object by touch.
  2. Object Recognition: Robotics perception which has the ability to recognize and identify objects in their environment. Which involves the process of analyzing the sensor data that is recognized based on the

classifiers used in the specific applications.Where machine learning, Deep learning, and reinforcement learning are used to perceive this from the robot environment.

3. Localization and Mapping: Robotics perception has a major role in robot localization and mapping tasks. Where it has simultaneous localization and Mapping SLAM algorithms are used to create the map.
4. Obstacle Detection and Avoidance: By analyzing the sensor data, the robot can identify the distance and plan appropriate paths to navigate around them safely.
5. Scene Understanding: It makes the robot learn and understand the environment by utilizing sensors and other tools.

- Reasoning and Planning: Cognitive robots use reasoning and planning algorithms to process the information they perceive and do the decision-making. And majorly it will imply the techniques like probabilistic reasoning, symbolic reasoning, and logical reasoning reason to understand the environment, plan their activities and solve problems.

- Learning and Adaption: Cognitive robots with the learn and adaption techniques have the ability to experience and adapt to their behavior accordingly. They utilize the algorithms with machine learning techniques, deep learning techniques, and reinforcement learning techniques.

- Human-robot interaction: Cognitive robotics emphasizes the interaction between humans and robots where focusing mainly on developing intuitive and natural interfaces such as speech recognition, gesture recognition, and facial expression analysis for the possibility to enable seamless communication and collaboration between humans and robots.

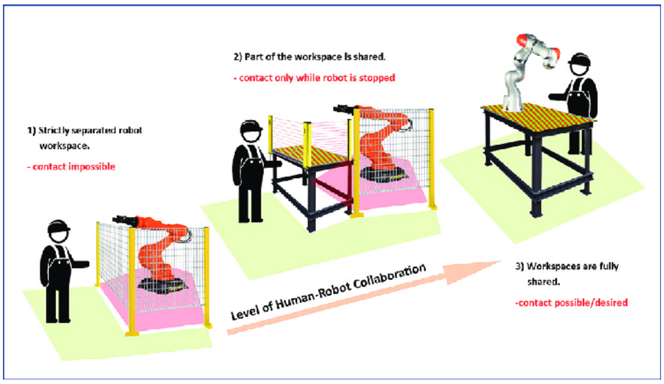


Figure 2.3: Human robot interaction

2.1.2 Advantages in Cognitive robotics

Cognitive robots have a number of benefits in several areas. In order to efficiently solve issues and make wise judgments, cognitive robots must first be able to see and comprehend complicated settings. Cognitive robots continually hone their problem-solving skills by adjusting to new circumstances and learning from their mistakes. The enhanced human-robot contact is a key additional benefit. Robots that are cognitively capable of understanding human gestures, words, and emotions can react correctly and collaborate and communicate with people in a natural way. The potential for applications in fields like customer service, healthcare, and education is expanded by this improved human-robot connection. Another significant benefit of cognitive robotics is its emphasis on learning and adaptability. By gaining new information and abilities via experience, these robots can enhance their performance over time. Cognitive robots improve their effectiveness and efficiency by evaluating data, identifying patterns, and making defensible judgments. Cognitive robots have several major benefits, including flexibility and autonomy. They are capable of navigating their surroundings on their own, identifying items and locations, and carrying out tasks without continual human assistance. With this degree of independence and adaptability, they may be used for a variety of tasks, including assisting in healthcare, automating production, and exploring dangerous places. An interdisciplinary approach that brings together knowledge from numerous domains, including AI, robotics, psychology, neuroscience, and philosophy, is another advantage of cognitive robotics. This interaction encourages the sharing of information and ideas, which advances each field and yields fresh discoveries. Cognitive robotics' interdisciplinary approach promotes creativity and aids in the creation of more advanced and effective robotic systems. In conclusion, cognitive robotics enables greater human-robot interaction, learning, adaptability, flexibility, and autonomy, as well as the advantages of a multidisciplinary approach. These benefits open the door for the creation and use of intelligent robotic systems in a variety of industries, ultimately benefiting society as a whole.

2.1.3 Dis-advantages in Cognitive robotics

- Cost and Complexity: Creating thinking robots requires a lot of time and resources. Intelligent systems need complex hardware, sophisticated algorithms, and a lot of computer power to be designed so that can sense, reason, and learn. The expense and complexity of this technology may prevent cognitive robots from being widely used.

- Ethical Issues: As cognitive robots develop in power, ethical issues come into play. Carefully addressing matters like privacy, security, and the possible influence on human employment is necessary. The area has difficulty ensuring that cognitive robots are employed responsibly and ethically.

- Ambiguity and Unpredictability: The actual world is frequently ambiguous, unexpected, and both. Given that they rely on models and algorithms that do not always deliver correct or comprehensive information, cognitive robots may find it difficult to manage such circumstances efficiently. In cognitive robots, dealing with uncertainty still poses considerable difficulties.

- Cognitive Restrictions: Despite their outstanding powers, cognitive robots are still unable to equal human intelligence. Robotic systems still need to catch up to human perception, comprehension, and decision-making in terms of complexity. In some circumstances, this constraint may have an impact on the effectiveness and dependability of cognitive robots.

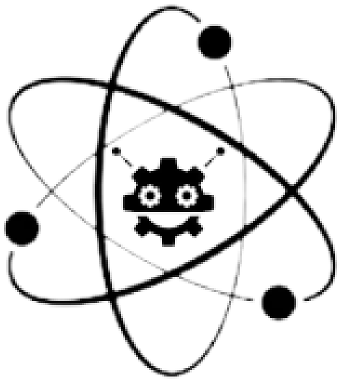
- Compatibility issues might arise when integrating several technologies and systems, which is a common task in cognitive robots. Utilizing components from several suppliers or research groups may result in interoperability problems when combining AI algorithms, robotic hardware, and cognitive models.

2.2 Quantum robotics

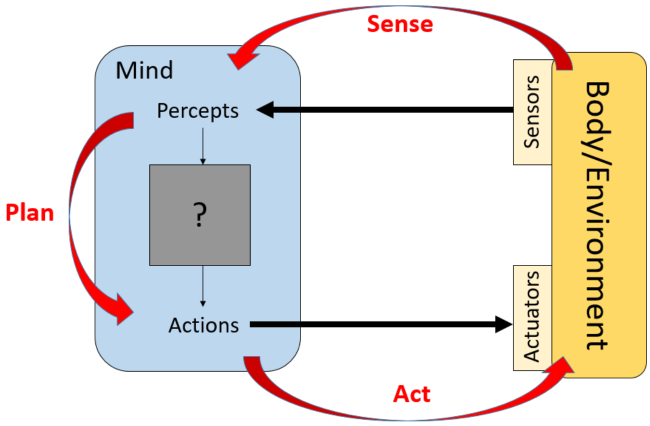
Quantum robotics (P. Tandon, S. Lam, B. Shih, T. Mehta, A. Mitev, and Z. Ong, [9]) is the field that integrates quantum mechanics and robotics (C. Petschnigg, M. Brandstötter, H. Pichler, M. Hofbaur, and B. Dieber,[10], A. Chella, S. Gaglio, M. Mannone, G. Pilato, V. Seidita, F. Vella, and S. Za- mmuto [11]), it focuses mainly on the possibility of implementing quantum systems to enhance the capability of robotics systems to advance the robot's ability. Below are the Quantum robotics emerging applications such as Quantum sensing and measurements, metrology, Quantum control and manipulation, Quantum machine learning, Quantum communication and networking, Quantum simulation, etc, etc ...

In our thesis we work mainly focus on Quantum robotics-related fields such as quantum cognition ((J. R. Busemeyer and Z. Wang[12])What is quantum cognition, and how is it applied to psychology?) (J. R. Busemeyer, P. Fakhari, and P. Kvam [13]) [14] and quantum-inspired robotics to prove the possibility of generating next-generation robots with respect to integrating the human-like quantum cognition [15] inspire model for robots with a quantum-like- perception model to organize the noisy uncertainty data from the robot environment and makes the decision making based on the following using a qiskit based python framework.

1



(a) Quantum-robotics



(b) Anatomy of Sense-plan-act

Figure 2.4: Thesis para diagram Concept

2.2.1 Quantum cognition for robotics

Quantum formalism is mainly a theoretical framework that uses principles of quantum mechanics to model human cognitive processes and decision-making.

[1]Quantum Robot. `http://www.quantum-robot.org/` Quantum Robot is a Python package for quantum-like perception modeling for robotics. The package exploits the Qiskit framework, implementing the models on quantum circuits which can be simulated on a classical computer or sent to a quantum backend (service provided by IBM Quantum Experience). The project was started in 2019 by Davide Lanza as a Master thesis research, with the help of Fulvio Mastrogiovanni and Paolo Solinas.

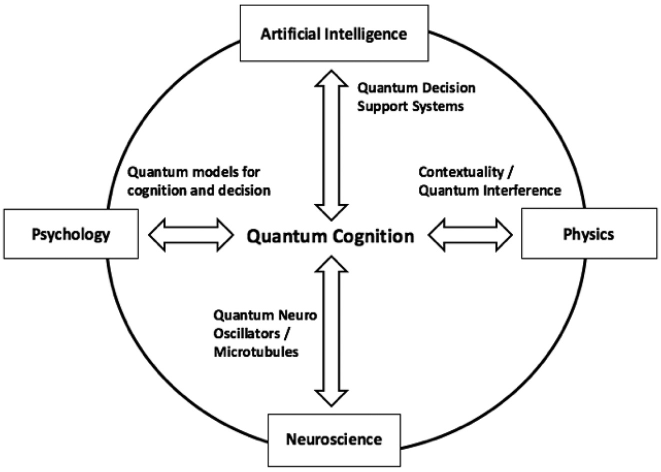


Figure 2.5: Quantum-cognition

This field suggests that human decision-making can be better described with a quantum-like model instead of classical models. But the application of quantum cognition is still ongoing research and exploration. And this field focuses on the development of intelligent machines performing tasks autonomously or with minimal human intervention, while there is significant progress in developing classical AI and Cognitive Architecture for robots. And the application of the quantum cognition field is in the earlier stage on the basis of the list below.

- Decision-making: Quantum-inspired Decision-making [2] model can pave the way to increase the decision-making capability of the robots. As per [2] [3] Quantum-like models can capture phenomena such as context-dependent decision bias and explore decision[16][17][18][19] [20][21].

- Uncertainty and Sensing:[2] [3] In this case, Quantum cognition has the potential to help the robot deal with uncertainties and incomplete information. Quantum-inspired algorithms such as quantum probability theory and quantum-like Bayesian networks can be used to model and update beliefs in situations where the sensory information is noisy or incomplete.

- Learning and Adaptation: Robots can learn in complicated and unpredictable contexts by using machine learning methods influenced by quantum mechanics. It is possible to investigate quantum machine learning models, such as quantum neural networks or quantum reinforcement learning, to enhance the generalization and adaptability of robots[16].

- Human-Robot Interaction: Quantum cognition models can be used to investigate and comprehend human cognition in human-robot interaction.

Robots may be able to adjust their behavior to better match human cognitive processes by taking into account the quantum-like nature of human decision-making, resulting in more logical and intuitive interactions[16].

2.2.2 Quantum-inspired robotics

Quantum-inspired robotics is the field that defines the inspiration from the concept and principles of Quantum mechanics to the field of robotics. It can t be directly robotics related to quantum computing or quantum algorithm in robot control or decision-making, and it draws inspiration from quantum principles to increase the capabilities of robotics systems[22][23][24][25][26][27][28].

- Sensing and perception: Quantum-inspired methods can be utilized to enhance robots' senses and faculties of perception. For instance, sensors inspired by quantum mechanics can offer improved sensitivity for identifying and measuring a variety of physical properties. Robots may be able to detect their surroundings more precisely and precisely thanks to quantum-inspired imaging systems like cameras and lidars[29][30].

- Optimization and Machine Learning: Robotic system optimization can be achieved using quantum-inspired algorithms, such as those inspired by quantum annealing or quantum-inspired optimization methods. These algorithms can assist robots in more effectively and efficiently resolving challenging optimization issues, such as path planning, workload distribution, or trajectory optimization[31].

- Decision-Making and Control: Quantum-inspired models of decision-making can be utilized to improve robots' capacity for making decisions. Robots can modify their behavior and make better-informed judgments in difficult situations by using quantum-inspired algorithms, such as quantum-inspired reinforcement learning and quantum-inspired Bayesian networks.

- Swarm robotics: The coordination and behavior of swarm robotic systems can be influenced by quantum-inspired ideas. For instance, efficient and secure communication amongst robots in a swarm can be facilitated using quantum-inspired communication protocols. Swarm algorithms influenced by quantum mechanics can also aid in improving a swarm of robots'[32] collective behavior and emergent characteristics[33][34][35][36][29][37].

- Hardware Inspired by Quantum Theory: Quantum-inspired robotics can also investigate the creation of hardware parts motivated by quantum theories. Improve robotic capabilities, this can entail using quantum-inspired sensors, actuators, or computational architectures that imitate certain properties of quantum systems, such as superposition or entanglement.

Although quantum-inspired robotics has the potential to advance the discipline, actual applications and practical implementations are still in the early phases of research. The future of quantum-inspired robots will be greatly influenced by current research and technology developments, which have only just begun to be completely understood.

2.2.3 Quantum-inspired robotics Pros And Cons

2.2.3.1 Advantage of Quantum-inspired robotics:

- Increased processing power: Compared to traditional computing techniques, quantum-inspired algorithms might provide considerable computational advantages. This may make it possible for robots to carry out intricate computations and optimizations more quickly, which would enhance their capacity for decision-making, planning, and control.

- Improved search and optimization algorithms: Quantum-inspired approaches, including quantum annealing or quantum-inspired search algorithms, may increase the effectiveness of addressing optimization issues. Robotic applications that need effective path planning, trajectory optimization, or resource allocation may find this useful.

- Increased sensor data processing: Processing of sensor data might be done more quickly and effectively thanks to quantum-inspired algorithms. Robots functioning in dynamic surroundings may benefit most from this since it will enable them to sense changes more quickly and react to them.

- Machine learning innovations: Quantum-inspired machine learning algorithms, such as quantum neural networks or quantum-inspired reinforcement learning, may lead to the development of novel methods for learning, pattern identification, and prediction. This could result in enhanced robot perception, adaptability, and autonomy.

2.2.3.2 Disadvantage of Quantum robotics:

- Technical difficulties: Quantum computing and algorithms influenced by quantum mechanics are still in their infancy. Significant technical obstacles, including those related to hardware requirements, algorithm design, and integration with current robotic systems, are associated with the actual application of these technologies in robotics.

- Limited availability of quantum hardware: At the moment, there are few costly quantum computers with enough qubits and reliable coherence times. Researchers and developers working on quantum-inspired robots may encounter obstacles, preventing the general use of these approaches, such as access to quantum hardware.

- Complexity and learning curve: Quantum-inspired algorithms and concepts can be difficult to grasp and need a strong foundation in quantum theory. Researchers and developers would need to become experts in both quantum computing and robotics in order to use these approaches, which may have a high learning curve.

- While quantum-inspired algorithms have great potential, it is still unclear how big of a competitive edge they will have over traditional techniques in various robotic applications. It's crucial to carefully assess and contrast existing conventional techniques with the performance enhancements and advantages of quantum-inspired robots.

Chapter 3

State of the Art

3.1 Quantum Computing Basics

3.1.1 Quantum Computing

Quantum Computing is an emerging field in computing that has the ability to use quantum mechanics to perform computations. While classical computers store and process information in bits, quantum computing uses quantum bits or qubits((A. Y. Kitaev, A. Shen, and M. N. Vyalyi,[38]), which can exist in a superposition of both $|0\rangle$ and $|1\rangle$ states simultaneously.

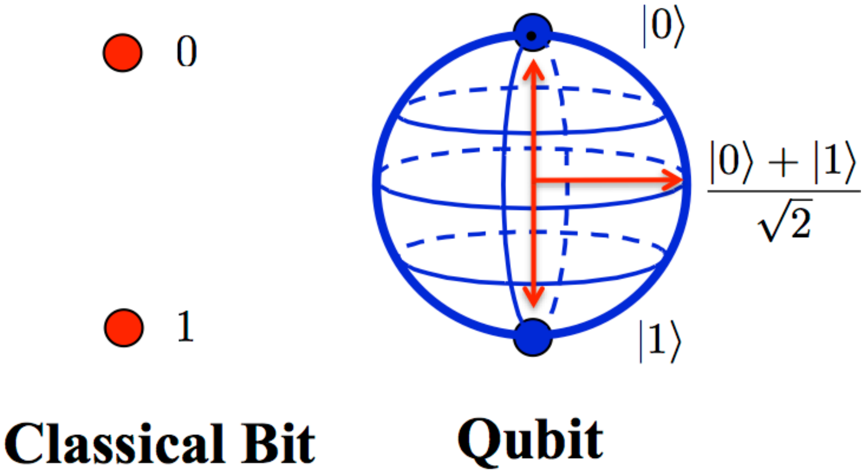


Figure 3.1: Classical-bit vs Qubit

In classical computing, binary logic gates that manipulate classical bits process data. Qubits, on the other hand, can reside in a state of superposition in quantum computing, which allows them to concurrently represent several states. Due to the special ability of qubits to execute parallel computations, quantum computers may be able to solve complicated problems with a significant advantage.

Quantum gates, which are comparable to conventional logic gates, are used in quantum computers. To carry out quantum operations and change the system's quantum state, quantum gates manipulate qubits. Quantum algorithms can be used to solve particular problems more quickly than classical algorithms by applying sequences of quantum gates.

3.1.1.1 Importance and Potential Applications

Due to its special characteristics, quantum computing is extremely important and has the potential to change many industries. Quantum computing has the potential to significantly impact a number of crucial fields, including:

Cryptography: Many of the current cryptographic schemes, like RSA and ECC (Elliptic Curve Cryptography), may be broken by quantum computers. On the other hand, using quantum key distribution algorithms that make use of the laws of quantum mechanics, quantum cryptography provides the opportunity for secure communication.

Real-world issues such as logistics, supply chain management, financial portfolio optimization, and traffic routing all require the optimization of complex systems. These optimization issues may be solved more effectively by quantum algorithms like the quantum approximate optimization algorithm (QAOA) and quantum annealing.

Simulation: Unlike conventional computers, quantum computers are capable of simulating and studying physical systems, such as chemical reactions, quantum materials, and biological processes, in great detail and accuracy. Research in areas like quantum physics, materials science, and drug development could advance more quickly thanks to quantum simulations.

Quantum machine learning seeks to use the capabilities of quantum computing to improve conventional machine learning techniques. Quantum algorithms have the potential to enhance pattern recognition, optimization, and data processing tasks. Examples of these algorithms include the quantum support vector machine and quantum neural networks.

Financial Modeling: By enabling quicker and more precise calculations for complicated financial models, portfolio optimization, and risk assessment, quantum computing has the potential to transform financial modeling and risk analysis.

These are but a few examples of the possible uses for quantum computing. New and intriguing opportunities are anticipated to arise as the area develops, influencing numerous fields and resulting in major improvements in computational capabilities.

3.1.1.2 Quantum principles

Fundamental ideas and laws known as quantum principles control how particles and systems behave at the quantum level. These ideas come from the study of quantum mechanics, a branch of physics that explains how matter and energy behave at very small scales.

The following are some essential quantum laws:

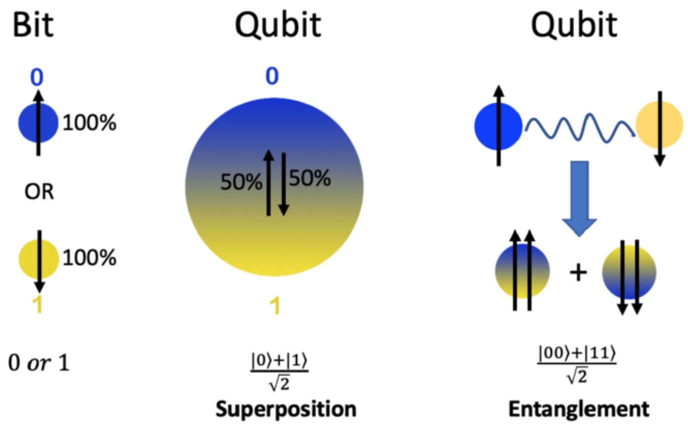


Figure 3.2: Qubit - superposition - Entanglement

- Superposition: The idea that quantum systems can exist in numerous states at once is known as superposition. To put it another way, a particle or quantum system can be in a state that is a combination of or a linear sum of all of its potential states. A qubit, for instance, maybe in a superposition of its 0 and 1 states.
- Wave-particle duality: According to quantum physics, particles like electrons and photons can behave both like waves and like particles. According to this theory, every particle has a wave nature that may be characterized by a mathematical concept known as a wave function.
- When a quantum system is measured, the wavefunction "collapses" into one of the potential states that is chosen at random by the probabilities inherent in the wavefunction. Because the process of measurement affects the result, this collapse is frequently referred to as the observer effect.
- Werner Heisenberg's uncertainty principle claims that there is a basic upper bound on the accuracy with which certain pairs of physical attributes,

such as position and momentum, can be simultaneously known. The ability to determine the other property accurately decreases as one property's measurement accuracy increases.

- Entanglement is a phenomenon in which the states of two or more particles become so coupled that they cannot be described independently from one another. When two particles are entangled, no matter how far apart they are, measuring one particle instantly changes the state of the other particle.
- Quantum tunneling: According to classical physics, particles should not be able to cross energy barriers, however, quantum tunneling allows particles to do so. It happens because particles behave like waves and can "tunnel" past obstacles to appear on the other side.
- Quantum interference: When particles behave like waves, quantum interference results. When two or more quantum waves overlap, they may interfere constructively, increasing the likelihood that particular events will occur, or destructively, reducing the likelihood of those occurrences or canceling them out.

  These ideas form the basis for comprehending and extrapolating the behavior of quantum systems, together with mathematical formalism and equations. Precision measurements, quantum computers, quantum cryptography, and a host of other phenomena have been made possible thanks to quantum mechanics, which has also proven very successful in explaining a wide variety of phenomena. But the quantum world frequently exhibits surprising behaviors, which puts our conventional intuitions to the test and necessitates new perspectives on the nature of reality.

3.1.1.3 Quantum Gates

Quantum gates are mathematical techniques used to change the states of qubits and carry out computations in a quantum computer. Quantum gates act on the quantum state of one or more qubits in a manner similar to that of classical logic gates.

Commonly used quantum gates include:

- Hadamard gate (H):
$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$
- Pauli-X gate (X):
$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

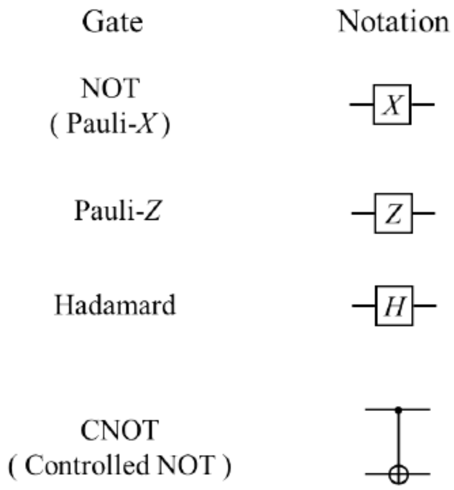


Figure 3.3: Commonly-used-quantum-gates

- Pauli-Y gate (Y):

$$Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$$

- Pauli-Z gate (Z):

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

3.1.1.4 Quantum Measurement

By making an observation, a qubit can be read for information through a technique known as quantum measurement. A qubit's state collapses to one of the base states $|0\rangle$ or $|1\rangle$ when it is measured, with a probability given by the superposition's amplitudes.

Applying a measurement operator to the qubit's state is how measurements are made in quantum computing. The measurement basis, which might be the computational basis ($|0\rangle$ and $|1\rangle$) or any other basis in which the qubit's state is expressed, corresponds to the measurement operator.

The qubit's superposition collapses to a classical state upon measurement. The squared magnitudes of the amplitudes in the superposition indicate the likelihood of receiving a particular measurement result.

For example, if a qubit is in the state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, the probability of measuring $|0\rangle$ is $|\alpha|^2$, and the probability of measuring $|1\rangle$ is $|\beta|^2$. The probabilities add up to 1, ensuring that the measurement outcome corresponds to one of the possible basis states.

Since the measurement procedure is irreversible, once a qubit has been measured, the knowledge of its initial state has been lost. By retaining their state after measurement, bits in classical computing do not have this property, which sets quantum computing apart from it.

3.1.1.5 Quantum Parallelism

Quantum parallelism is one of the fundamental tenets of quantum computing. It is used to describe how quantum computers can use the superposition property of qubits to carry out several computations at once.

In traditional computing, calculations are carried out in a sequential manner, with each operation depending on the results of the one before it. The processing speed and efficiency are constrained by this sequential nature. However, quantum parallelism enables quantum computers to investigate numerous computational avenues concurrently, enhancing their processing capacity enormously.

A quantum algorithm that uses quantum parallelism may process every potential input at once by encoding it in the superposition of qubits. This makes it possible to explore multiple computing paths at once, perhaps accelerating the resolution of some issues.

Consider a scenario where we must search for a certain item in an unsorted database to solve an issue. Until a match is found, each item would be systematically checked in traditional computing. In contrast, quantum parallelism enables a quantum computer to simultaneously search through all superpositions of things, potentially accelerating classical computations exponentially.

3.1.1.6 Quantum Interference

The phenomenon of quantum interference ((D. Aerts and S. Sozzo, [39]), which results from the superposition of qubits, is essential to quantum algorithms. It happens when various quantum states interact positively or negatively, leading to various measurement results.

Constructive interference happens when the superposition amplitudes of compatible states combine, increasing the likelihood of a particular measurement result. The desired solutions in quantum algorithms are amplified by this constructive interference, increasing the possibility that they will be measured.

On the other side, destructive interference happens when the superposition amplitudes of incompatible states cancel each other out, which lowers the likelihood of measuring particular outcomes. Ineffective solutions are suppressed via destructive interference, which boosts the effectiveness and precision of quantum algorithms.

Quantum algorithms control the probability amplitudes of several computing paths by taking use of interference phenomena. Interference can be used to increase the possibility of measuring the desired answers and inhibit irrelevant or wrong solutions by skillfully designing quantum gates and the superposition states.

3.1.1.7 Quantum algorithms

The special characteristics of quantum systems are exploited by quantum algorithms to tackle particular problems more quickly than conventional algorithms. Quantum algorithms are computational algorithms created to be implemented on quantum computers. These methods use parallel computations and may exponentially speed up some jobs by taking advantage of concepts like superposition, entanglement, and interference.

- Shor's algorithm 3.4: A quantum algorithm called Shor's algorithm is used to identifying the prime factors of a composite number through the process of integer factorization. Shor's technique uses the period-finding property of quantum algorithms to effectively factorize huge numbers and takes advantage of the ability of quantum computers to operate on multiple values concurrently through superposition. The possibility of breaching traditional public-key cryptography methods like the widely used RSA is significantly increased by this approach.

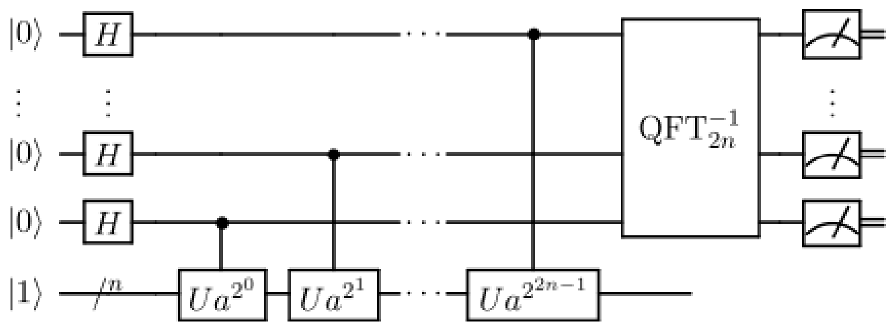


Figure 3.4: Shor's Algorithm

- Grover's algorithm 3.5: Grover's algorithm is a quantum search algorithm that effectively searches a database with N items that is unstructured in around N steps as opposed to the O(N) steps needed by traditional techniques. It takes advantage of the quantum parallelism and interference features to speed things up. Applications for Grover's algorithm include unstructured problem-solving, optimization issues, and database searching.

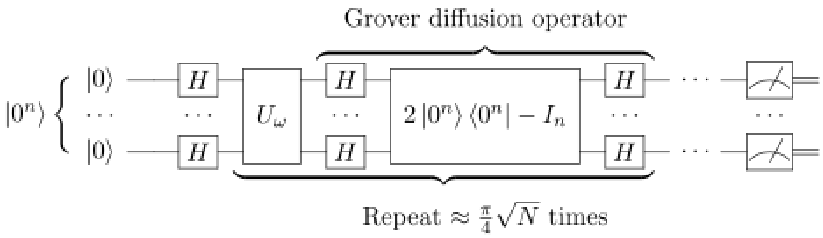


Figure 3.5: Grover's Algorithm

- Quantum simulation algorithms: Quantum computers are used to model the behavior of quantum systems using quantum simulation methods. Quantum simulation algorithms provide the opportunity to explore complicated quantum processes and materials because classical computers find it difficult to simulate large-scale quantum systems. Quantum simulation methods include the Variational Quantum Eigensolver (VQE) and Quantum Phase Estimation (QPE).

- Quantum Fourier Transform (QFT) 3.6 : Shor's method and many other quantum algorithms make use of this core subroutine. It efficiently computes periodic characteristics and quantum state changes by converting the input of a quantum state into its frequency domain representation.

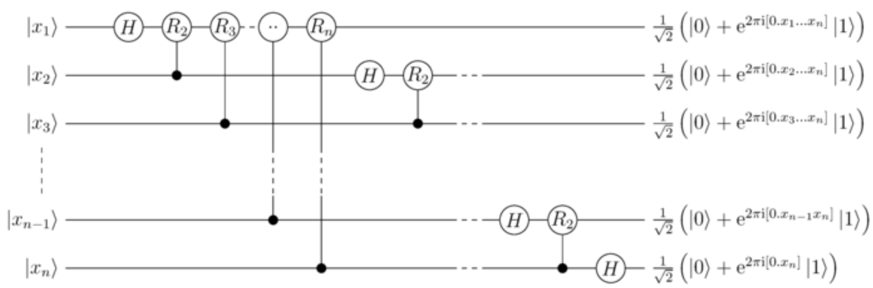


Figure 3.6: Quantum Fourier Transform Algorithm

- Quantum approximate optimization algorithm (QAOA) 3.7: Algorithm for quantum approximate optimization (QAOA) Combinatorial optimization issues are solved using the hybrid quantum-classical optimization approach known as QAOA. By changing the parameters of a quantum circuit, it can approximate solutions to optimization problems by combining classical optimization techniques with quantum computing capabilities.

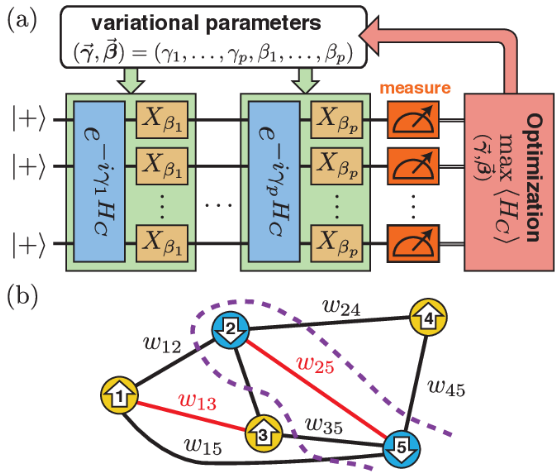


Figure 3.7: Quantum approximate optimization algorithm

3.1.2 Single Qubit Calculation

Let's consider a single qubit system.

3.1.2.1 Initialization

We start with the qubit in the state $|0\rangle$.

3.1.2.2 Hadamard Gate

We apply the Hadamard gate (H gate) to the qubit, which transforms it into a superposition state:

$$H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

Now, the qubit exists in a superposition of both 0 and 1 states. As we are going to use the Qiskit framework to work with the thesis let's have a look at its working with a Qiskit program Fig:3.8 with an example script. And have a short breakdown of the script in detail about the single qubit working below,

- One qubit is used to build a quantum circuit, and a Hadamard gate is used to put the qubit in a superposition state.
- The qubit is then measured, and the measurement's outcome is subsequently stored in a conventional bit.
- The applied measurements and gates are printed on the circuit.

- To get a single measurement result, the circuit is simulated using the Qiskit Aer simulator with shots=1.
- To display the results of the measurement, the measurement result is printed.

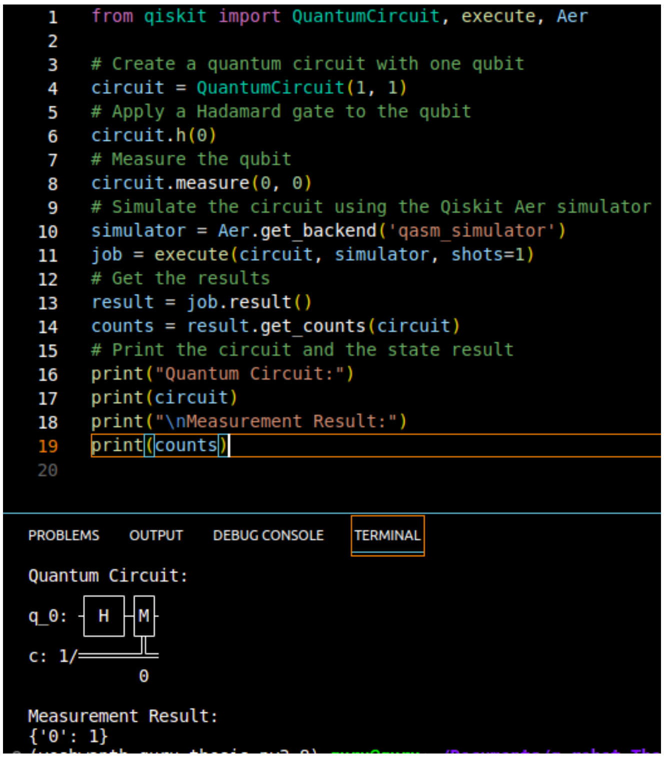


Figure 3.8: Single-qubit using Qiskit

3.1.3 Multi-qubit system Calculation

If we consider a single qubit that is coupled in three dimensions, it means that the qubit has interactions or couplings with three independent systems or degrees of freedom. This can be represented as a three-qubit system, where each qubit corresponds to one dimension.

Let's denote the basis states of the individual qubits as $|0\rangle$ and $|1\rangle$. The state of the three-qubit system can be represented as a combination of all possible combinations of the basis states:

$$|a\rangle \otimes |b\rangle \otimes |c\rangle$$

Here, $|a\rangle$ represents the state of the first qubit, $|b\rangle$ represents the state of the second qubit, and $|c\rangle$ represents the state of the third qubit.

```python
# Import necessary libraries
from qiskit import QuantumCircuit, Aer, execute
from qiskit.visualization import plot_bloch_vector

# Create a quantum circuit with one qubit
circuit = QuantumCircuit(1, 1)

# Apply rotations around different axes of the Bloch sphere
circuit.rx(0.3, 0)  # Rotation around the x-axis

# Measure the qubit
circuit.measure(0, 0)

# Simulate the circuit using the Qiskit Aer simulator
simulator = Aer.get_backend('qasm_simulator')
job = execute(circuit, simulator, shots=1)
result = job.result()
counts = result.get_counts()

# Get the measurement result
measurement_result = list(counts.keys())[0]

# Print the circuit and the measurement result
print("Quantum Circuit:")
print(circuit)

print("\nMeasurement Result after Rx(0.3):")
print(measurement_result)

# Plot the Bloch vector representation
plot_bloch_vector([1, 0, 0])  # Initial state |0>

# Apply the next rotation
circuit.ry(0.6, 0)  # Rotation around the y-axis

# Measure the qubit
circuit.measure(0, 0)

# Simulate the circuit again
job = execute(circuit, simulator, shots=1)
result = job.result()
counts = result.get_counts()

```

Figure 3.9: Multi-qubit using Qiskit(Part-a)

```python
measurement_result = list(counts.keys())[0]

# Print the circuit and the measurement result
print("\nQuantum Circuit:")
print(circuit)

print("\nMeasurement Result after Rx(0.3) + Ry(0.6):")
print(measurement_result)

# Plot the Bloch vector representation
plot_bloch_vector([0, 1, 0])  # State after Rx(0.3) + Ry(0.6)

# Apply the final rotation
circuit.rz(0.9, 0)  # Rotation around the z-axis

# Measure the qubit
circuit.measure(0, 0)

# Simulate the circuit once more
job = execute(circuit, simulator, shots=1)
result = job.result()
counts = result.get_counts()

# Get the measurement result
measurement_result = list(counts.keys())[0]

# Print the circuit and the measurement result
print("\nQuantum Circuit:")
print(circuit)

print("\nMeasurement Result after Rx(0.3) + Ry(0.6) + Rz(0.9):")
print(measurement_result)

# Plot the Bloch vector representation
plot_bloch_vector([0, 0, 1])  # State after Rx(0.3) + Ry(0.6) + Rz(0.9)

```

Figure 3.10: Multi-qubit using Qiskit(Part-b)

Since the three qubits are coupled in three dimensions, there can be interactions or correlations between their states. The specific details of these interactions depend on the nature of the coupling and the physical system being considered.

To perform operations on the three-qubit system, we can apply quantum gates or operations that act independently on each qubit or gates that allow for interactions between qubits.

For example, let's apply the Hadamard gate (H gate) to the first qubit:

$$H(|0\rangle \otimes |b\rangle \otimes |c\rangle)$$

Applying the H gate to the first qubit, we get:

$$(H|0\rangle) \otimes |b\rangle \otimes |c\rangle$$

Using the Hadamard gate transformation we discussed earlier, this becomes:

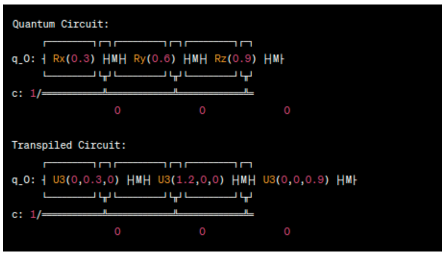


Figure 3.11: Multi-qubit using Qiskit - result(Part-c)

$$\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |b\rangle \otimes |c\rangle$$

Expanding the expression, we have:

$$\frac{1}{\sqrt{2}}(|0\rangle \otimes |b\rangle \otimes |c\rangle + |1\rangle \otimes |b\rangle \otimes |c\rangle)$$

This represents a superposition of eight basis states: $|0bc\rangle$, $|1bc\rangle$, $|0bc\rangle$, $|1bc\rangle$, $|0bc\rangle$, $|1bc\rangle$, $|0bc\rangle$, and $|1bc\rangle$. Here, $|abc\rangle$ denotes the state where the first qubit is in state $|a\rangle$, the second qubit is in state $|b\rangle$, and the third qubit is in state $|c\rangle$.

By applying various gates and operations to the three-qubit system, we can manipulate its state and perform computations that utilize the entanglement and correlations between the qubits.

It's important to note that the specific interactions and couplings in the three-dimensional system would determine the nature of the computations and the entanglement effects that can be achieved. The above explanation provides a general understanding of how a multi-qubit system can behave when coupled in multiple dimensions. Apart from that can have a reference of an example multi-qubit system working in qiskit 3.9 3.10.

3.1.4 Measurement Mechanism in a qubit

We perform a measurement on the qubit to determine its final state. The measurement collapses the qubit into one of the basis states, either 0 or 1, with specific probabilities.

Let's assume that upon measurement, we obtain the outcome 0. This means the qubit has collapsed into the state $|0\rangle$. For a detailed understanding of the measurement techniques in qubit let's have a look in detail at an example scenario with the quantum mechanics calculations for better understanding purposes.where it starts with the single qubit in a superposition state: Let's consider a qubit in a superposition state:

$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$,

where α and β are complex probability amplitudes, and $|0\rangle$ and $|1\rangle$ represent the basis states.

To calculate the probabilities of obtaining different measurement outcomes, we need to determine the squared magnitudes of the probability amplitudes. In this case, the probability of measuring $|0\rangle$ is given by $|\alpha|^2$, and the probability of measuring $|1\rangle$ is given by $|\beta|^2$.

However, it's important to note that the measurement process in quantum mechanics is inherently probabilistic. When we measure the qubit, it collapses into one of the basis states with a probability given by the squared magnitude of the corresponding probability amplitude.

For example, let's say we measure the qubit $|\psi\rangle$ and obtain the outcome $|0\rangle$. In this case, the qubit collapses to the $|0\rangle$ state, and subsequent measurements will always yield the same outcome. Mathematically, we can represent this as:

$|\psi\rangle = |0\rangle$ (with probability $|\alpha|^2$), $|\psi\rangle = |1\rangle$ (with probability $|\beta|^2$).

The measurement outcome provides information about the state of the qubit, but it also alters the state itself. After the measurement, the qubit is no longer in a superposition but in a definite state corresponding to the measured outcome.

It's worth noting that the probabilities $|\alpha|^2$ and $|\beta|^2$ must add up to 1 because the measurement outcome must be one of the basis states. Therefore, we have the normalization condition:

$|\alpha|^2 + |\beta|^2 = 1$.

In practice, to calculate measurement outcomes and probabilities, we typically start with a qubit in a known quantum state, apply appropriate mathematical operations or physical processes, and then evaluate the probabilities associated with different measurement results.

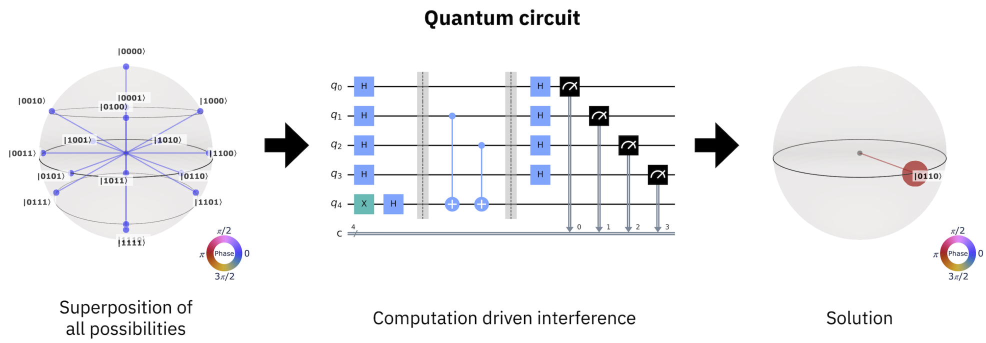


Figure 3.12: Qubit working process visuals

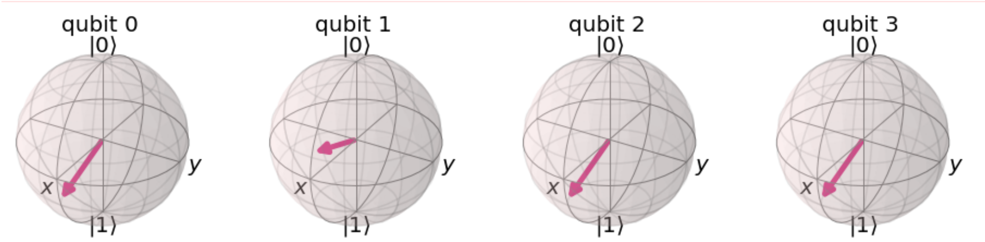


Figure 3.13: Qubit example of plotting results

3.1.5 Challenges and Limitations

- Decoherence:A significant obstacle to quantum computing is decoherence. It alludes to the degradation of quantum states and the loss of coherence brought on by interactions with the outside world. Quantum systems are highly susceptible to noise, disturbances, and interactions with the electromagnetic or particle fields around them.

  It is challenging to maintain the integrity of qubits and carry out accurate computations because decoherence causes the delicate quantum states, such as superposition and entanglement, to decay quickly. Qubit interactions with their surroundings cause undesired noise to entangle with quantum information, introducing mistakes and inaccuracies into quantum processes.

  Researchers use a variety of methods, including error-correcting codes, quantum error correction, and quantum fault tolerance (described in the next paragraph), to reduce decoherence. To lessen the effects of decoherence, quantum hardware designs concentrate on lowering ambient interactions and enhancing isolation.

- Scalability:In quantum computing, scalability presents a serious challenge. The number of qubits and the complexity of computations that can be handled by contemporary quantum computers are limited, despite the fact that they have the ability to solve some problems more quickly than classical computers.

  The difficulties of upholding coherence, minimizing mistakes, and controlling quantum operations multiply exponentially as the number of qubits rises. Numerous technological challenges must be overcome in order to increase the number of qubits, including precise control of qubit interactions, reducing crosstalk between qubits, and assuring high-fidelity quantum gates.

  Furthermore, solving the qubit connectivity problem is necessary for scaling quantum systems. It gets harder to make sure that every qubit can connect to every other qubit as the number of qubits increases. Building large-scale quantum computers capable of addressing complicated problems requires establishing long-range interactions and maintaining high-quality connections between qubits.

- Quantum Error Correction: In order to address the inherent fragility of qubits and mitigate the effects of errors brought on by decoherence and other sources, quantum error correction is a crucial approach. It entails redundantly storing quantum information across several qubits, enabling error detection and repair.

  Ancilla qubits, which are supplementary qubits used in quantum error correction algorithms, are used to store and handle incorrect data. Errors can be found and fixed without directly measuring the encoded quantum states by taking measurements on the ancilla qubits and using quantum operations.

  The fundamentals of quantum error correction codes, such as the surface code, and the stabilizer codes, serve as the foundation for quantum error correction algorithms. These codes add redundancy to the qubit encoding, enabling faults to be found and fixed by observing and adjusting the ancilla qubits.

  However, due to the increasing resource needs, such as more qubits, higher connectivity, and sophisticated gate operations, implementing quantum error correction is difficult. Building dependable and scalable quantum computers requires overcoming error correction constraints and attaining fault-tolerant quantum processing.

3.1.6 Quantum Supremacy

Quantum supremacy is the point at which a quantum computer is able to carry out a computational task better than the most potent classical computers. It illustrates the moment at which a quantum computer can do a task much more quickly than any classical machine, proving the superiority of quantum computing in some applications.

By showing a quantum computer that could do a certain computation that would take even the most powerful classical supercomputers an impracticable amount of time, Google's research team claimed to have attained quantum supremacy in 2019. This accomplishment demonstrated how far quantum computing has come and how it has the ability to execute some tasks better than classical computers.

3.1.7 Industry Players and Developments

Significant improvements and participation from numerous industry participants, including tech firms, academic organizations, and startups, have been seen in the field of quantum computing. In the world of quantum computing, some famous participants are:

- IBM: IBM has been at the forefront of research and development for quantum computing. They actively work to advance quantum hardware, software, and applications and provide access to their quantum computers via the IBM Quantum Experience platform.
- Google: Google is credited with making important advances in quantum computing, including the assertion that it has attained quantum dominance. They keep researching hardware improvements, error correction methods, and quantum algorithms.
- Microsoft: Microsoft is actively working on developing and researching quantum computing. They are concentrating on creating Azure Quantum, a platform for scalable quantum computing that seeks to offer cloud access to quantum tools, software, and hardware.
- IonQ: Leading provider of trapped-ion quantum computing is IonQ. They have made strides in creating high-fidelity qubits, lowering mistakes, and scaling their quantum systems. They use trapped ions as qubits.
- Building hybrid quantum-classical computing systems is the main emphasis of Rigetti Computing. They are dedicated to creating useful applications for quantum computing and making their quantum processors available through the Forest platform.

  These are only a few of the several groups working to advance quantum computing. With ongoing research, technological developments, and partnerships aimed at overcoming obstacles and achieving the potential of quantum computing for a variety of sectors and applications, the subject is still evolving quickly.

3.2 Literature Review

3.2.1 Additional basics for the thesis work

In this Section, we are about to discuss the Motivation(State of the art) of the work before that we are going to have a small discussion about some terms and techniques which play some part of the role in the thesis work. First, the discussion and overview of quantum Computing have been explained in the section 3.1 apart from that some of the following will be listed below

3.2.1.1 Binocular rivalry

Binocular rivalry: When competing visual information is simultaneously delivered to each eye, a process known as binocular rivalry takes place, which causes a perceptual alternation or competition between the two pictures. It is a perceptual phenomenon that emphasizes how several sensory inputs compete with one another and work together in the brain. Dichoptic presentation is a method used in binocular rivalry tests where participants see two separate pictures, one for each eye. For instance, one eye may be shown a red picture while the other is shown a green one. The observer's vision often switches between viewing the red picture and the green image, as if they were oscillating back and forth, rather than experiencing a combination of the two hues. The switching between the two perceptions may happen naturally, with no outside influence, or it can be impacted by a variety of things including attention, the potency of the stimulus, and cognitive processes. Although the precise processes driving binocular rivalry are still under investigation, it is thought that they entail connections between several visual cortical regions and other brain regions in charge of perception and attention. Numerous studies have used binocular rivalry as a method to examine visual perception, brain processing, and awareness. It offers an understanding of how various visual representations vie for dominance in the brain and how the brain resolves competing sensory information. The neural correlates of awareness, attention, and the influences of numerous elements on perception, such as the impact of emotional content or contextual information, have all been studied using binocular rivalry. Understanding binocular rivalry has consequences for areas including neurology, psychology, and vision studies and may give information on basic processes of perception and cognition. Researchers want to acquire a better understanding of how our brain creates our subjective perception of the world based on sensory data by investigating the processes underpinning binocular rivalry. The basic description of the following is to understand the approach described by the (E.manousakis[40]) in his research about the Quantum formalism to describe binocular rivalry and its working about the formalism to understand it which is part of the thesis working will be discussed in the architecture working section.

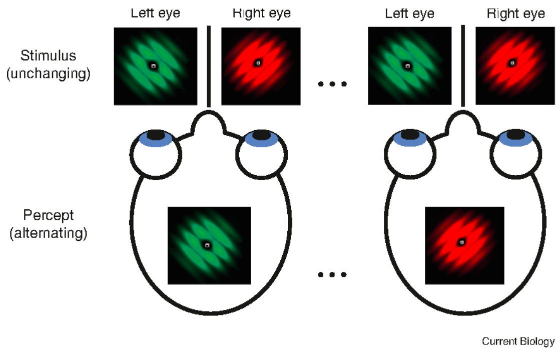


Figure 3.14: Binocular rivalry

3.2.1.2 Ambiguous figures

Ambiguous figures: Reversible or multistable forms, commonly referred to as ambiguous figures [41], are visual cues that may be interpreted in several ways. They are inherently ambiguous, allowing the spectator to see them in a variety of ways, either spontaneously or after some effort. Typically, these images include contrasting visual clues or components that may be arranged and interpreted in several ways. As a consequence, even if the actual input doesn't change, the viewer's impression of the figure fluctuates between several readings of it. The spectator may perceive a subjective "flip" or a smooth transition between the different interpretations as a result of the rapid or gradual alterations in perception.

The Necker cube, which can be seen as a three-dimensional cube that flips in orientation, Rubin's vase, which can be seen as either a vase or two facing profiles, and the famous duck-rabbit illusion are some famous examples of ambiguous figures. Depending on the viewer's perception, these figures can be seen as either a duck or a rabbit.

Because they show how active perception is and how interpretation shapes our visual experience, ambiguous figures are a useful tool for understanding how vision works. By stressing the impact of top-down processes, such as expectations, prior experiences, and contextual clues, on our perception of the environment, ambiguous figures cast doubt on the notion that physical inputs and perception correspond one to one.

Researchers have used ambiguous figures to look into a variety of perception-related topics, such as the neural correlates of multistable perception, the mechanisms underlying perceptual switches, the function of attention in perception, the impact of culture and individual differences on interpretation, and the mechanisms underlying perceptual switches. Additionally, research on binocular rivalry, visual illusions, and visual awareness have all made use of these figures.

In conclusion, ambiguous figures are visual cues that may be interpreted in a number of different ways, causing perceptual uncertainty and switching between several interpretations. They provide insightful information on the mechanics of perception and the intricate interplay between top-down cognitive processes and bottom-up sensory data(E.Conte[42]).

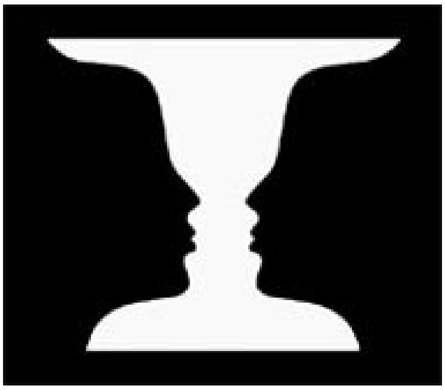


Figure 3.15: Ambiguous figures

3.2.1.3 Fuzzy logic for robot Knowledge representation

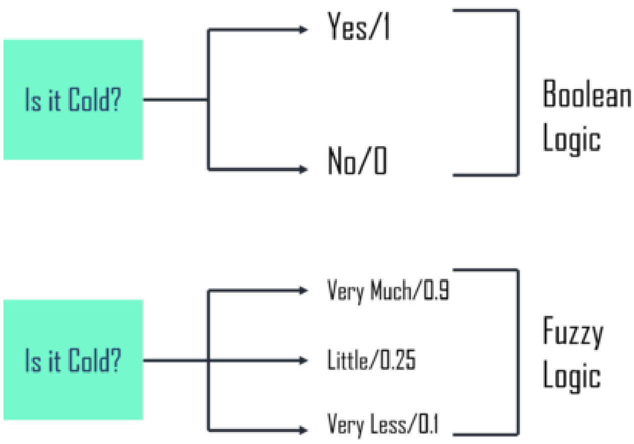


Figure 3.16: Fuzzy comparison

Fuzzy logic is a branch of logic that deals with deliberation and judgment in the face of ambiguity and imprecision. Fuzzy logic allows for intermediate values between true and false, expressing degrees of truth or membership, as opposed to standard binary logic, which only accepts values of either true or false that are clear-cut and exact.

In fuzzy logic, variables may have a range of values to describe degrees of truth or membership, often between 0 and 1. This makes it possible to reason and make decisions in circumstances that are unclear, ambiguous, or imprecise.

Numerous industries and applications, such as control systems, artificial intelligence, decision support systems, pattern recognition, and expert systems, often employ fuzzy logic. It is especially helpful in circumstances when conventional reasoning may be insufficient since it offers a way to represent and manage uncertainty and imprecision.

To collect and handle data that is intrinsically fuzzy or ambiguous, fuzzy logic uses linguistic variables, fuzzy sets, fuzzy rules, and fuzzy inference. Fuzzy logic makes use of these methods to allow the representation and manipulation of intricate real-world ideas as well as to support more adaptable and flexible thought and decision-making processes.

To deal with ambiguity and imprecision in quantum computing, fuzzy logic may be used to models inspired by quantum mechanics. The idea of fuzzy quantum sets, which blends fuzzy logic with quantum theory, is one instance.

In conventional quantum computing, measurements provide exact results with probabilities, and quantum states are defined using complicated probability amplitudes. The knowledge that is at hand, nevertheless, can be insufficient or ambiguous in certain instances. In contrast to rigid probabilities, degrees of membership provide a more flexible and reliable representation of quantum states in fuzzy quantum sets.

Let's use quantum-inspired optimization as an illustration of how fuzzy quantum sets are used in quantum-inspired models in this context. Quantum-inspired optimization algorithms, such the Quantum-Inspired Genetic Algorithm (QIGA), seek to use the concepts of quantum physics to enhance optimization procedures.

Qubits are often used to represent possible solutions in a traditional quantum optimization technique, and quantum gates may be utilized to effectively search for the best solutions. However, it's possible that the quantum state of qubits is not exactly understood owing to elements like noise, decoherence, or a lack of resources. Fuzzy quantum sets may be used to describe this uncertainty.

A quantum state with different degrees of membership is represented by a fuzzy quantum set. A qubit may have a degree of membership in each state rather than being wholly in the states of 0 or 1, for example. To illustrate uncertainty or superposition between the two states, a qubit may, for instance, have a degree of 0.7 in state 0 and a degree of 0.3 in state 1.

Fuzzy quantum sets may be used in quantum-inspired optimization to describe ambiguous or imprecise states of qubits while the optimization process progresses. This makes it possible to explore the solution space more thoroughly while accounting for any ambiguities or errors in the quantum state representation.

Fuzzy quantum sets enable quantum-inspired optimization algorithms to handle partial or uncertain data more skillfully, perhaps producing better optimization outcomes in real-world situations where quantum systems may experience noise or flaws.

It's crucial to remember that research on quantum-inspired computing is still ongoing and that there are many methods for adding fuzzy logic and uncertainty management into models inspired by quantum mechanics. Depending on the method and issue area, the specifics and applications may change ((V. M. Peri and D. Simon[43],I. Konukseven[44],L. A. Zadeh[45]),L. A. Zadeh[46], Z. Toffano and F. Dubois,[47] . In many different contexts, including modeling uncertain settings, fuzzy logic is often used to express and manage uncertainty. Fuzzy logic may be used to describe an uncertain environment in the following ways:

Linguistic Variables: Fuzzy logic enables the representation of qualitative and imprecise data using linguistic variables. The different qualities or circumstances that form an uncertain environment may be described using language variables, such as "high probability," "low confidence," or "medium risk."

The degrees of membership or truth connected to linguistic variables are captured using fuzzy sets. Fuzzy sets may be used to describe the ambiguous states or circumstances in an uncertain environment. A fuzzy set labeled "High Probability" can, for instance, contain degrees of membership that range from 0.7 to 1.0, which indicates a high degree of uncertainty.

Membership Functions: The degree of membership or truth connected to each element in a fuzzy set is defined by the membership functions. The inputs (such as sensor readings or data) are translated by these functions into membership levels within the fuzzy sets. Membership functions may be created to manage various degrees of uncertainty in the case of an uncertain environment, enabling flexible and adaptive modeling.

Fuzzy Rules: In a fuzzy logic system, fuzzy rules provide the connections between the input and output variables. The heuristics or expert knowledge that direct behavior or decision-making in an uncertain situation are captured by these principles. By taking into account diverse combinations of input variables and the degrees of membership, fuzzy rules may be created to address ambiguity.

Fuzzy Inference: Fuzzy inference combines the inputs and fuzzy rules to provide suitable outputs or judgments. Fuzzy inference mechanisms may use degrees of membership and fuzzy rules to reason and make conclusions that account for uncertainty in an uncertain environment. Fuzzy sets that reflect the inferred states or actions based on the uncertain environment might be the outputs.

An uncertain environment may be successfully described and handled by using these fuzzy logic approaches. In order to reason and make judgments in the face of ambiguity, fuzzy logic allows for the modeling of imprecise and uncertain information. Due of this, fuzzy logic is a useful tool for handling ambiguous situations in a variety of applications, including robotics, decision support systems, and control systems.

3.2.2 Motivation to the thesis work

In our Thesis, we are going to integrate a Quantum-like perception model with a Robots Cognitive Architecture of a classical system where the quantum-like perception model makes robots learn the surrounding and trigger the mechanism based on the following. We will discuss the Cognitive-Architecture (D. P. Benjamin, [8]) (J. E. Laird, [7]) (C. Burghart, [6]) development which is the main sector of the thesis where the scenario is developed for a Handover task with a human. At first, the traditional cognitive Architecture was designed based on the behaviors discussed in the chapter section 2.1 with the current preferred scenario. As per the (E.manousakis[40]) In order to formalize the subjective process of perception logically, the study suggests a formalism based on the quantum theory of measurement. The probability distribution of dominance durations in binocular rivalry is explained using this technique. The hypothesis explains why dominance duration lengthens under periodic stimulus interruption in a way that is consistent with experimental evidence and makes sense. Additionally, it offers testable hypotheses on the distribution of perceptual change across time. The study offers a thorough framework for comprehending the dynamics of subjective experiences in perception as a whole. And the following paper's findings and the formalism related to the Subjective process of perception have potential implications for cognition in robotics. Based on the following understanding and mathematically describing how humans perceive and experience the world and the study approach of the (E. Conte[42]) Quantum interference which we have seen in the section 3.1.1.6 effects are seen in quantum mechanical processes. Because they include an extra term known as the quantum interference term, quantum probabilities in this situation end up being different from classical ones. The author examines the existence of quantum interference and if violations of the classical probability field may be seen during perception-cognition by human participants using ambiguous pictures. The 256 people who participated in the tests provide conclusive proof that there is such a quantum effect. As a result, it seems that mental processes during the perception and understanding of ambiguous figures adhere to quantum physics. This information may be used by researchers to create cognitive robotic systems. Robotic systems' capacity to detect and understand sensory data may be improved by including formalism, which might result in more complex and nuanced cognitive capacities. This would make it possible for robots to interact with their surroundings more than humans do, which might enhance their capacity for complicated scenario decision-making and adaptation. The following idea (Davide Lanza, Fulvio Mastrogiovanni, Paolo Solinas [2]) gave the possibility of a Quantum-like perception model which has the application of quantum theory-based formalisms in cognitive science is discussed in the study, with a focus on their descriptive properties such state superposition and probabilistic interference. It states that it has been discovered that the dynamics of biological or cognitive systems and quantum dynamics are isomorphic. The purpose of the research is to investigate whether a robot with restricted sensing capabilities may successfully use a quantum-like (QL) perception paradigm. The (Davide Lanza, Fulvio Mastrogiovanni, Paolo Solinas[2]) approach provides a preliminary case study and recognizes the drawbacks of applying quantum models to real quantum devices, especially in imbalanced circumstances when mistakes may occur. Instead, they advocate the use of simulation-oriented methods to research QL models for robot behavior and analyze the benefits of QL methods for knowledge representation and processing in robotics. According to the paper's conclusion, simulations are preferable to actual implementations on quantum hardware for studying robot behaviors and exploring the advantages of QL approaches, even though quantum theory-based models may be advantageous for robot perception and cognition. (Davide Lanza,Fulvio Mastrogiovanni Paolo Solinas[2])The use of quantum-like (QL) techniques in cognitive science is discussed, along with how well they work for perception, cognition, and decision-making. It claims that quantum theory-inspired formalisms have been used in this sector for a long time. To determine the viability of a QL perception model for a robot with constrained sensing capacities, the authors performed a preliminary investigation.The research proposes an extended model for multi-sensory inputs that builds on their earlier work and enables the construction of a multidimensional representation of the environment from sensor data. To show how this model gives a clear and elegant representation, they concentrate on a 3-D case study. It incorporates aspects that are especially useful for modeling uncertainty and decision-making processes.The QL paradigm also makes it easier to define query operators that allow for the analysis of any given world state. These operators provide numerical measurements of the robot's belief or degree of confidence in relation to that particular condition.For robots with limited sensory capabilities, the research provides a QL perception model. It highlights the model's benefits, such as its capacity to manage ambiguity, aid in decision-making, and provide condensed world representations. The approach also enables measurement of the robot's level of confidence in various world situations using query operators.The author developed the qiskit-based quantum-like model with the above-discussed abilities, which we are going to see the following integration with the classical cognitive Architecture where the proposed models of the (Davide Lanza,Fulvio Mastrogiovanni, Paolo Solinas[2] [3]) has been integrated and comparing the following (Classical and Quantum inspired Cognitive Architecture working in simulation and in the real-time robot Tiago++) to learn about the implementation of the quantum-like approach in the classical robot hardware and followed with the analysis of the robot belief state based on the Query operators approach as proposed in the (Davide Lanza,Fulvio Mastrogiovanni, Paolo Solinas[2] [3]) where the fields of Cognitive robotics ,Quantum cognition for robots ,Quantum robotics,Quantum inspired robotics has been focused with the implementation of the techniques in a Handover task scenario with a human results the Human robot interation will be in line for the Thesis output as Cognitive Architecture For robots inspired by Quantum Computing which is a novel basis like the approaches (X. Li, Z.-y. Liu, V. Zhang, and B. Xu,[48] ,Y. Hu and C. K. Loo [49],M. Mannone, V. Seidita, and A. Chella,[50] , Q. Zhang [51], to prove the next generation robot development. [1]

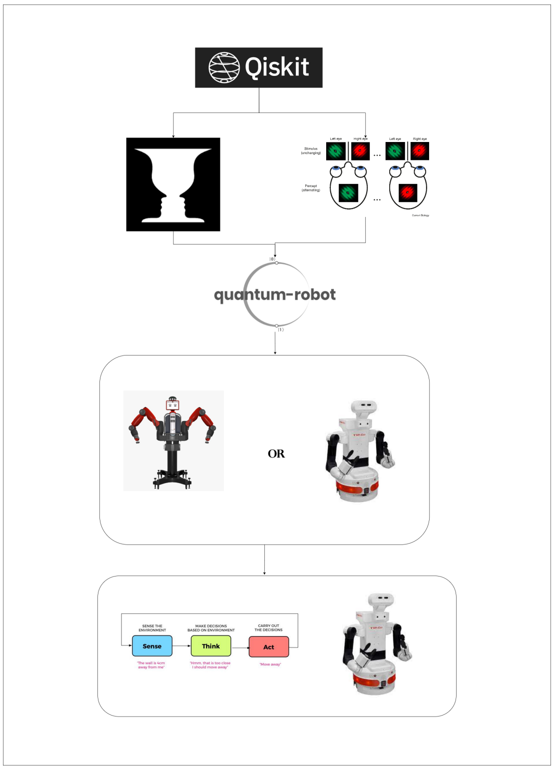


Figure 3.17: State-of-the-art

[1]The *quantum-robot* package is a Python package for quantum-like perception modeling in robotics. It utilizes the Qiksit framework and implements the models on quantum circuits, which can be simulated on a classical computer or executed on a quantum backend (provided by IBM Quantum Experience). The *quantum-robot* website can be accessed at `http://www.quantum-robot.org/`.

Chapter 4

Decision making

Decision-making is the process of selecting a course of action among various alternatives. It is an essential component of human existence that affects situations both personally and professionally. Making choices entails a number of processes and considerations, whether it's determining what to eat for breakfast, selecting a college to attend, or handling complicated business situations. The above description is to mention that Decision-making in cognitive architectures is the process through which a robot's cognitive system examines data, evaluates the environment, and decides which course of action to take to accomplish its objectives. Robots see processes and decide depending on the information provided when using cognitive architectures, which are frameworks that give these processes a structure. There are many aspects in the robot Cognitive system but besides decision-making, other aspects have been described in the 2.1.As the Decision -making is the main role in the thesis work. So Chapter 4 has been separated to describe and compare the following between the Classical bit and the quantum bit. And its difference based on the behavior of decision-making(D. Kalogeras, [17],J. M. Yearsley[18],T. Kovalenko[19],L. Rosendahl[26],S. Han and X. Liu[27]).First here are the key aspects of the decision-making of a robot's Cognitive Architecture:

- Perception and Sensing: Robots use a variety of sensors, including cameras, microphones, and touch sensors, to see and sense their surroundings. These sensory inputs are processed by perception modules, which then provide a picture of the robot's environment. Decisions are made using this knowledge as a foundation.

- Knowledge Representation: Cognitive architectures provide systems for storing and representing knowledge about the environment, task domain, and past experiences of the robot. Numerous formats, including symbolic

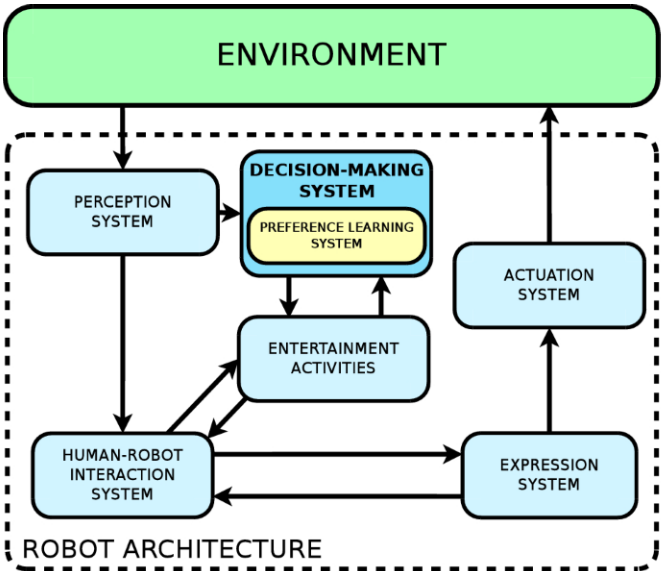


Figure 4.1: Decision-Making Architecture of an Example-1 Cognitive Robot System

representations, ontologies, and statistical models, may be used to store this information.

- Reasoning and Planning: Robots utilize planning processes to organize the information at hand and draw judgments. This entails making inferences based on the robot's knowledge base, using probabilistic reasoning, and applying logic. Then, planning algorithms are used to create action sequences to accomplish certain objectives or address issues.

- Management of objectives and Tasks: Robots make decisions based on specified objectives and tasks. These objectives may be established by human operators or acquired automatically. These objectives are managed and organized by the cognitive architecture, which also prioritizes them and assigns resources appropriately.

- Algorithms for Making Decisions: Cognitive architectures include decision-making algorithms that take into account a number of variables, including the robot's present state, the objectives to be attained, the resources at hand, and the limitations. These algorithms may use machine learning methods, rule-based systems, or a mix of the two.

- Adaptability and Learning: Robots can adjust their decision-making processes in response to new information and changing situations thanks to cognitive architectures. Over time, they may develop their decision-making skills, refresh their information, and learn from previous mistakes.

- Execution and Monitoring: After a choice has been made, the cognitive architecture converts it into the proper motor instructions to direct the robot's movements. The architecture also has controls for checking on how these activities are being carried out, giving feedback, and making changes as necessary. Cognitive architectures may include modules for human-robot interaction, allowing robots to take into account human input, comprehend orders given in normal language, and cooperate with people in making decisions.

So in the thesis, decision-making is the main role where the following will be proved with the integration of a Quantum-like perception model in the Cognitive Architecture and calculating the comparison performance based on the behaviors of the robot will be discussed in the upcoming chapters.

4.1 Comparison between the Qubit and Classical bit working

Qubits provide the benefit of simultaneous exploration and probabilistic outcomes, which might be useful for decision-making in some situations. However, qubits' practicality is constrained by their mistake sensitivity and implementation difficulties. Classical bits are a dependable option for decision-making in most situations because they provide predictable outputs, are more error-resistant, and are simple to apply.

For the real-time comparison to understand the decision-making in humans and robotics in a detailed manner refer [1] to the image 4.2

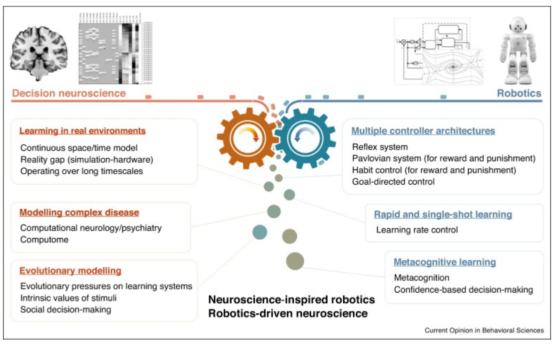


Figure 4.2: Decision Making Human Vs Robots

Table 4.1: Comparison of Qubits and Classical Bits

Aspect	**Qubits**	**Classical Bits**
Representation	Qubits can represent complex combinations of states through superposition, allowing for parallel exploration of multiple possibilities.	Classical bits represent only one state at a time, limiting decision-making to the sequential exploration of alternatives.
Exploration of Solutions	Qubits can explore multiple solutions simultaneously, potentially enabling faster evaluation of various options.	Classical bits require sequential exploration of solutions, which may take longer for problems with a large number of possibilities.
Probabilistic Results	Measurement of qubits provide probabilistic results, offering insights into the likelihood of different outcomes.	Classical bits provide deterministic results, yielding a definite value for decision-making.
Error Sensitivity	Qubits are highly sensitive to errors caused by noise and decoherence, requiring error correction techniques to maintain the accuracy of results.	Classical bits are more robust and less prone to errors, making classical computing systems more reliable for decision-making.
Algorithmic Advantages	Quantum algorithms, leveraging the properties of qubits, can provide exponential speedup for specific problems, offering a potential advantage in decision-making for those cases.	Classical algorithms are well-established and efficient for a wide range of decision-making tasks, making classical computing a reliable choice for many applications.
Implementation Challenges	Developing and maintaining quantum systems capable of manipulating qubits is technologically challenging, requiring precise control and mitigation of noise and error sources.	Classical bits can be implemented using various technologies with well-established methods, making them more accessible and easier to integrate into existing systems.

Chapter 5

Methodology

5.1 Tiago ++ Robot and its specification

In the following Chapter, we are going to have a detailed description of the Hardware and software which has been used in the complete thesis work. Before getting into the discussion thanks to pal robotics for the invention which makes the researchers explore The bimanual TIAGo service robot was created to operate in confined spaces and is known as TIAGo++. When ambidextrous manipulation is needed for jobs such as ambient assisted living or light industrial, TIAGo++ characteristics make it the right platform for study. It combines capabilities for movement, perception, manipulation, and human-robot interaction with the intention of assisting people and assisting researchers all around the globe in pushing the boundaries!

Figure 5.1: Tiago ++ robot Gallery

Key characteristics of TIAGo++ include:

- Bimanual Manipulation: TIAGo++ has two robotic arms, allowing it to carry out ambidextrous manipulation-required activities. The robot can handle things and carry out intricate activities that call for the coordinated use of both arms thanks to this function.

- Mobility: With its wheeled mobile base, TIAGo++ maintains the mobility feature of the original TIAGo robot. This makes the robot adaptable to various interior conditions by enabling it to travel independently or be teleoperated.

- Perception: Through sensors like cameras, depth sensors, and laser scanners, TIAGo++ adds perception capabilities. These sensors provide the robot the ability to understand its surroundings, find items, and efficiently interact with the environment.

- Human-Robot Interaction: TIAGo++ is intended to make it easier for people and robots to communicate. It has functions and user interfaces that facilitate easy user interaction and cooperation. In ambient assisted living situations, where the robot could help people with regular duties or provide assistance in a nice way, this element is especially crucial.

The overall goal of TIAGo++ is to operate as a research platform that integrates the skills of mobility, perception, manipulation, and human-robot interaction. By allowing investigations and experiments that require ambidextrous manipulation and activities related to ambient assisted living or light industries, it aims to help individuals and support researchers in expanding the area. And the Tiago ++ robot fig:5.1 has a lot more hardware available for TIAGo, such as a variety of sensors and end-effectors as well as an extension panel for adding other accessories and devices. In addition to Whole-Body Control, is based on the Ros-based architecture. In which the thesis working is based on the Cognitive Architecture Development for robots using the Tiago++ robot for a handover task in a Quantum Computing inspired way. So in our case, we are going to use ROS-Melodic for controlling the Tiago++ robot. The rest of the hardware and software will be described in the below sections

5.2 Experimental Softwares

Cognitive Architecture was developed firstly in the simulation approach once that has been proved, followed by the real-time deployment with the robot(Traditional and quantum-inspired). So the current section describes the software and in detail develops the scenario and the architecture,

5.2.1 ROS and its purpose

ROS stands for Robot Operating System. It is an open-source framework for building software for robotic systems. It is an open-source platform for creating robotic system software. In order to make the process of creating robot applications more straightforward, ROS offers a variety of tools, libraries, and standards. It is a middleware layer rather than an actual operating system that runs on top of a host operating system (like Linux) and offers a selection of services that may be used by robotic applications.

The distributed design of ROS makes it possible for multiple robot system components to communicate with one another. It offers a message system that enables communication between several nodes (single program modules). Nodes may be programmed in a variety of programming languages (such as C++ and Python) and can operate on a variety of computers, which makes it easier to create large robotic systems with scalable and modular architectures.

The vast ecosystem of pre-existing packages, which are reusable software components that perform common functionality, is one of the primary characteristics of ROS. Perception, control, mapping, localization, simulation, and other topics are all covered by these programs. These packages enable developers to streamline the creation process and concentrate on more complex duties unique to their robots.

Due to its adaptability, community support, and accessibility to a wide variety of resources, ROS has grown in popularity in both academic and industrial contexts. It has been extensively used to build and operate a range of robots, from tiny mobile robots to humanoid robots and even autonomous vehicles. It is also utilized in research, education, and commercial applications. For the current application, we are using the ROS-melodic that supports the Tiago++ robot. At first, I had a comparison study to implement the thesis work among the Baxter or Tiago++ robot . As I had a study about the navigation-based approach of (Davide Lanza [2]) which motivated me to choose the Tiago ++ robot to prove the current thesis task where the future work of the Architecture will be like the Mobile-manipulation based tasks with the implementation of the quantum-like perception model. So I chose the Tiago++ robot for the thesis work. So currently we are using the Melodic version of ros which is supported in Ubuntu 18.04. Because ROS Melodic was launched while Ubuntu 18.04 was the most recent Long-Term Support (LTS) version of Ubuntu, ROS Melodic is officially supported on Ubuntu 18.04. ROS Melodic was created and examined, particularly for Ubuntu 18.04 compatibility.

The Robot Operating System (ROS) is dependent on a number of system-level libraries, tools, and dependencies. Extensive testing, validation, and maintenance activities are necessary to support numerous operating system versions. To provide the greatest stability and compatibility for consumers, the ROS development team often concentrates on certain Ubuntu LTS versions.

Even though ROS Melodic may be installed on other Ubuntu releases or even Linux distributions, doing so can need extra steps and manual setup. The greatest community support and the smoothest experience come from sticking with the officially endorsed pairing of ROS Melodic with Ubuntu 18.04.

Figure 5.2: Ubuntu and ROS

5.2.1.1 Gazebo

One of Gazebo's significant advantages is its ROS integration. With Gazebo's seamless integration with ROS, which offers a framework for creating robotic applications, you can transfer information and instructions between simulated parts and actual systems. Because of this integration, Gazebo is a great option for designing, testing and verifying robotic algorithms and systems before putting them into use on actual robots.

- Physics Simulation: With the help of Gazebo, you can describe the dynamics and interactions of objects in a virtual setting thanks to its precise physics modeling capabilities. It supports a variety of physics engines, such as ODE, Bullet, and DART, each of which offers a unique set of simulation properties and performance trade-offs. To give robots and objects realistic behavior, Gazebo simulates forces, torques, collisions, friction, and other physical qualities.

- Sensor Simulation: You may mimic a variety of robotics-related sensors using Gazebo, including cameras, lidars, sonars, IMUs (Inertial Measurement Units), and more. Sensors' attributes and features, like their resolution, range of vision, noise, and distortion, may be customized. Gazebo sensor data may be used to test perception algorithms, create sensor fusion methods, and assess robotic system performance.

- 3D Visualization: A visually beautiful and engaging 3D simulation environment is offered by Gazebo. It depicts the virtual environment, robots, and items so you can see and examine their conditions and actions. To increase the realism of the simulation, the visualization might incorporate realistic lighting, shadows, textures, and material attributes. Real-time rendering is supported by Gazebo, allowing you to see the simulation play out and interact with the digital world.

- Robot Control: By providing control algorithms and behaviors, Gazebo allows you to command robots inside the simulation. Writing robot controllers allows you to program actuators like wheels and joint motors to move in response to instructions. You may get sensor readings and utilize them in your control algorithms by interacting with the simulated robot's sensors using Gazebo's APIs and tools. With this capacity, you may test and improve control tactics without running the risk of causing physical robots harm or racking up costly real-world expenses.
- Customization and Extension: Gazebo may be easily customized and expanded to meet different simulation requirements. It has a plugin-based design that enables you to add your own custom plugins to increase its capability. Using plugins, one may build unique robot models, specify new sensor types, apply unique physics models, and include more features in the simulation. To make customizing and automating simulation operations easier, Gazebo enables scripting in languages like C++, Python, and XML. And in our simulation scenario, we are using the Gazebo9 version that supports for the ubuntu 18.04 ros-melodic.

Figure 5.3: Gazebo

5.2.1.2 Rviz

A popular 3D visualization tool for ROS (the Robot Operating System) is called RViz. You can see and play around with many parts of a robot's vision, planning, and control. For showing sensor data, robot models, and other visualization components in a 3D environment, RViz offers a graphical user interface (GUI). The following are RViz's main characteristics and abilities:

- Visualization of sensors: The viewing of sensor data from several sources, including cameras, lasers, point clouds, and other sensors, is supported by RViz.It may provide sensor readings in a variety of 2D and 3D representations, such as point clouds and depth maps, as well as 2D views (such as camera pictures and laser scans).To customize the visualization to your requirements, RViz lets you define sensor-specific features like color mapping, the field of view, and filtering choices.

- Visualization of robots: You may see robot models in RViz, together with their joints, linkages, and other geometric features. You can precisely import and show the robot model thanks to its support for well-liked robot description formats like URDF (Unified Robot Description Format). For manipulating the robot's joints and seeing its kinematic configuration in real-time, use RViz's interactive controls. To better comprehend the locations and orientations of certain robot parts, such as grippers, end-effectors, or sensors, you may view and annotate those parts.

- Navigation and Planning Visualization: The planning and navigational elements of a robot system may be seen with RViz. It may provide planning trajectories, which depict the route a robot intends to take to arrive at a certain area. With the aid of RViz, you can see navigation-related components including occupancy grids, local cost maps, and global maps to better understand how the robot perceives its surroundings and the route it intends to take.

- Interactive Markers and Controls: You may interact with the shown data using RViz's interactive controls and markers. Through the GUI, you may choose out things, move them around, or change their characteristics. The development of interactive markers is also supported by RViz, which may be used to create unique user interfaces for engaging with the robot or the surrounding environment.

- Setup and Personalization: To customize the visualization to particular requirements, RViz provides a wide range of configuration choices. You may rapidly switch between several visualization settings thanks to the ability to store and load configuration files. With the use of plugins, RViz may be made to suit a particular need by adding new visualization features or extending current ones. Due to RViz's close integration with ROS, it is simple to connect to and view data that has been released on ROS-related issues. So this software is used for the robot visualization study in the thesis.

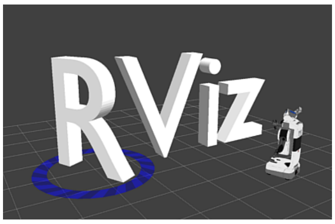


Figure 5.4: Rviz

5.2.1.3 Moveit

MoveIt is a widely used software framework that gives robots the ability to plan their motion planning and manipulate objects and is developed on top of the ROS (Robot Operating System). It is intended to make it easier to build complicated robot manipulation tasks, such as choosing and arranging things, gripping objects, and moving in an area.

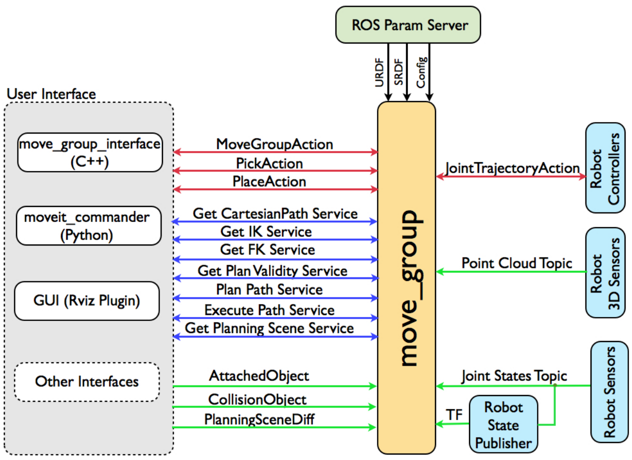


Figure 5.5: Moveit architecture-working

- Motion Planning: MoveIt offers tools and algorithms for creating robot motion plans that avoid collisions. Planning viable and optimal trajectories involves taking into consideration the robot's kinematics, environmental barriers, and intended job goals.
- Manipulation: It has grip and moving capabilities for use with robot grippers and arms. It offers high-level interfaces and resources for creating trajectory generation, inverse kinematics, and grasping techniques.

- Integration of perception: Moveit learns more about the world, it may interface with other perception systems, including cameras or depth sensors. Following that, this data is used for scene comprehension, object identification, and collision verification.

- Visualization and Simulation: It enables developers to view and test their robot's motion plans before implementing them on a real robot and supports simulation environments like Gazebo. This facilitates the process of developing and debugging.

- ROS integration: MoveThe connection with various robot parts, sensors, and actuators is made possible by its smooth integration with other ROS tools and packages. It utilizes ROS messaging and services for data transmission and coordination, and its design is based on ROS.

5.2.1.4 Thesis simulation Scenario outlook :

Thanks, pal robotics for the simulation data sets such as the robot, table, grasp object, etc. For a better understanding let's have a look at the thesis simulation scenario. First, we have discussed the gazebo workings in the section:5.2.1.1, and the reference to the simulation environment has been attached below fig:5.6.The fig:5.7 refers to the visualization using rviz. where the following software specifications were discussed in the section:5.2.1.2. And Moveit which has been described in the section 5.2.1.3. There for main components used in the simulation scenario have been discussed in detail. And further work on traditional architecture and Quantum-inspired architecture will be discussed in the above upcoming chapters.

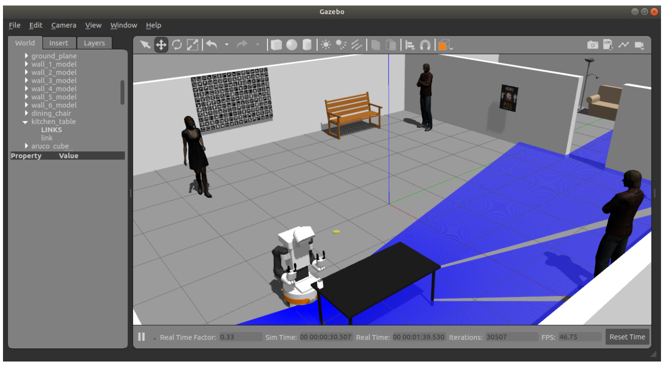


Figure 5.6: Gazebo-Thesis simulation scenario

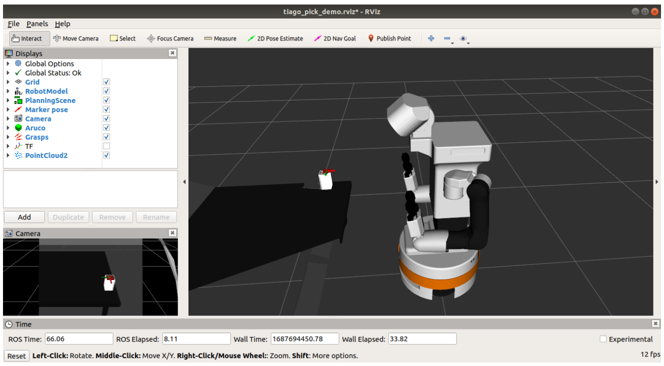


Figure 5.7: Rviz-Thesis simulation scenario

5.3 Quantum robot Package and its working Purpose as quantum-like perception model

In our thesis, a Python module called quantum-robot allows for the simulation of quantum-like perception in robotics. The software leverages the Qiksit framework to construct the models on quantum circuits, which may either be routed to an IBM Quantum Experience-provided quantum backend or emulated on a conventional computer. Davide Lanza began the project in 2019 as part of his master's thesis study, assisted by Fulvio Mastrogiovanni and Paolo Solinas. Currently, Davide Lanza is in charge of keeping it. Before getting into the detail

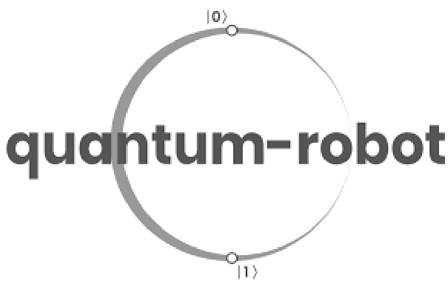


Figure 5.8: Qrobot

about the Quantum Robot Package let's have an explanation about the concept behind it with a real-time scenario.

- A lady is in her garden and is looking at a butterfly that is perched on a flower. The woman questions whether the butterfly is actually dead or just resting because it doesn't seem to be moving. The butterfly makes delicate motions, which she attentively watches, but they are too subtle for her to know exactly how it is feeling. She thus experiences a superposition of conscious states and is unsure of the butterfly's state—whether it is living or dead. Her awareness shifts back and forth between the two options as time goes on. If the butterfly isn't moving, does that mean it has died? Or is it just resting? she wonders. She is enthralled by the butterfly's delicate beauty and the intrigue surrounding its current state. A sudden wind gust blows through the garden, shaking the flower and upsetting the butterfly's perch. When disturbed, the butterfly flaps its wings to show that it is still alive. As the woman's superposition of conscious states dissolves, she is convinced that the butterfly was real and was only taking a nap.

- The initial superposition state is given by:

$$|\psi_0\rangle = \alpha|D\rangle + \beta|A\rangle$$

  Here, α and β are the complex probability amplitudes associated with the butterfly being dead ($|D\rangle$) or alive ($|A\rangle$).

- After the Hadamard gate operation, the state becomes:

$$|\psi_0\rangle = \alpha|0\rangle + \beta|1\rangle$$

  The Hadamard gate (H) is applied to the initial state, resulting in a new state represented by $\alpha|0\rangle$ (butterfly dead) and $\beta|1\rangle$ (butterfly alive). Here, $|0\rangle$ and $|1\rangle$ represent the respective qubit basis states.

- The measurement outcome is represented by the state $|\psi'\rangle$, which collapses to the butterfly being alive ($|1\rangle$).

With respect to reproducing the following behavior in the robot is the concept behind the qrobot python package, The robot uses a simulated quantum system to represent its knowledge based on the perceptual stimuli it has received. The robot's current "conscious" state is the result of a measure, which causes the system to collapse to a predetermined state. Consequently, a measurement takes place after gathering sensory data for a particular amount of time. There are the following components of the qrobot package such as the Qunit, Sensorial unit, Model, Burst, and the author gave the representation of an image to understand its working,Davide lanza[2] gave us a component diagram to understand its working.

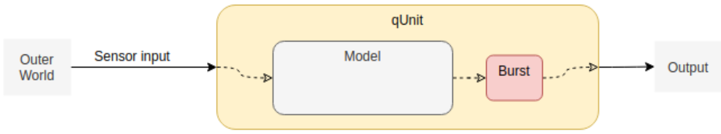


Figure 5.9: Qrobot components image

5.3.1 Components of the Qrobot Python package

5.3.1.1 SensorialUnit

The sensorial unit in the qrobot python package has the ability to get the sensorial inputs from the external environment, in this case, the qubit can handle the fuzzy logic-based knowledge representation because of qubit nature behavior. And the following knowledge is encoded in the qunit to its subcomponent called Model.

5.3.1.2 Qunit

From the sensorial unit, the values encoded in the Qunit subcomponent "Model"=The model collects binary input inside a given temporal frame, rotates the state vector to encrypt it, and then performs a measurement to produce a binary result based on the quantum measurement probability. And there are two models in the qunit class **Linearmodel** and **Angularmodel**. The angular model is mainly used in the thesis work as we are going to design the multi-sensory integration Davide, Fulvio, paolo[3], which means it represents the multi-qubit system and so sensorial input based on any dimension from the outer world the Angular model is used. Once after the representation by the Bloch sphere we are going to use burst techniques to result in the outcome state. For a better understanding of the qrobot working mechanism refer to the 5.10. The above components are important that need to be known before getting into further development. A short discussion has been derived here and further work with respect to the thesis scenario will be explained in the upcoming chapters. And thank you for the author to provide the Quantum-like [2]perception model library.

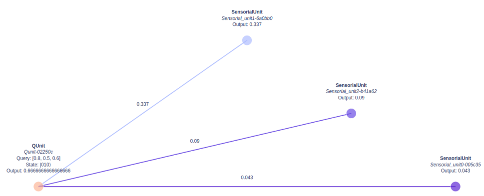


Figure 5.10: Sensory encoding and outcome mechanism in Qrobot Package

5.3.2 Qiskit and its purpose

A platform for creating open-source software called Qiskit enables anyone to create and run quantum algorithms on quantum computers. It is created by IBM and offers a collection of resources, including libraries and APIs, for working with quantum circuits, simulating quantum systems, and conducting research on IBM's quantum hardware.

Python is one of the many programming languages supported by Qiskit, which also provides both low-level control over individual quantum gates and a high-level interface for creating quantum circuits. It offers a variety of functions for designing, simulating, improving, and visualizing quantum circuits.

A component named Terra in Qiskit also provides the framework for creating quantum programs. It offers resources for modifying quantum states, building quantum circuits, and carrying out quantum operations. It also provides a range of quantum algorithms and error correction methods.

In addition, Qiskit provides a module called Aer, a powerful simulator for quantum circuits. It offers numerous simulation backends for varying degrees of accuracy and speed and enables users to replicate the behavior of quantum circuits on classical computers.

Last but not least, IBM Quantum, a module included in Qiskit, gives users access to IBM's cloud-based quantum computers. These tools let users submit their own quantum programs and run them on genuine quantum hardware, allowing for experimentation and investigation of quantum computing in practical settings.

In conclusion, Qiskit provides a comprehensive framework that enables scientists, programmers, and hobbyists to interact with quantum computers, from creating and simulating quantum circuits to actually running them on quantum devices. which would help the researchers to prove the possibilities of quantum-like approaches for future implementation. In the previous chapters, we have discussed the Quantum computing basics, in our thesis work, we are using the quantum computing simulator wrap-up library called qrobot 5.3.

5.4 Experimental hardwares

We have discussed the major components used in the thesis works and its methodology of usage in the working apart from the following thesis handover task scenario robot has been described in the section 5.1 and the software for the simulation and real-time has been discussed in the 5. And some other hardware like additional hardware for the handover task can be visualized in the realtime scenario description in upcoming chapters.

Chapter 6

Cognitive Architecture For Robots

6.1 Cognitive Architecture for Robots

After the detailed discussion about the basics and the motivation behind the work now we are going to see the implementation of those in the experiment at first in the current chapter refers to the development of the traditional cognitive architecture and its working in the simulation environment. At first, the architecture design is focused on the handover task. Where there is a human-robot-interaction scenario particularly picking an object from the table and handover to the human based on the Aruco-based pose estimation for object detection. The simulation scenario glimpse has been seen in the 5.6. we have used an aruco cube in the simulation scenario to grasp but handover goals have been predefined to a certain location where the following can be fulfilled by synchronizing the handover scene in real-time with a human.

6.1.1 Component Diagram of the Architecture and its working

6.1.1.1 Three nodes for object detection and grasppose

The following nodes Aruco_ros, Tiago_grasp pose detection, Tiago_aruco_detection in the Architecture were employed for the Aruco_detection of the object where the outcome of the following nodes results in the detection of the object and its 6-dof pose. And **/grasp_pose** of the detected object. At first, the **Aruco_ros** node detects the **/aruco_single/pose** is detected and passed the following topic to the node.**Tiago_aruco_detection** node where the detected aruco has been transformed to the robot **/base_footprint**(where the strategy to estimate the

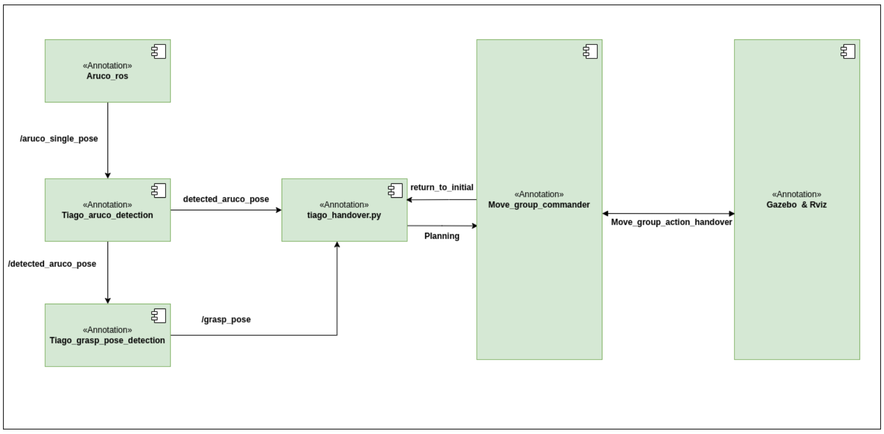


Figure 6.1: Traditional-Cognitive architecture for the robot (Component Diagram)

pose with respect to the robot) in which the tf2_ros library plays the role here to transform. The below calculation represents the calculation strategy with the thesis scenario as they play a major role.

- Transformation from TF2 frame to base link:

  Let's denote the pose of the ArUco tag in the TF2 frame as $(x_{\text{tf}}, y_{\text{tf}}, z_{\text{tf}}, q_{x_{\text{tf}}}, q_{y_{\text{tf}}}, q_{z_{\text{tf}}}, q_{w_{\text{tf}}})$, where $(x_{\text{tf}}, y_{\text{tf}}, z_{\text{tf}})$ represents the translation, and $(q_{x_{\text{tf}}}, q_{y_{\text{tf}}}, q_{z_{\text{tf}}}, q_{w_{\text{tf}}})$ represents the quaternion orientation.

  This transformation involves translating the position and rotating the orientation based on the relative pose of the TF2 frame with respect to the base link. Let's denote the translation and rotation of the TF2 frame with respect to the base link as $(x_{\text{bl}}, y_{\text{bl}}, z_{\text{bl}})$ and $(q_{x_{\text{bl}}}, q_{y_{\text{bl}}}, q_{z_{\text{bl}}}, q_{w_{\text{bl}}})$, respectively.

  The transformed pose of the ArUco tag in the base link frame can be calculated as:

$$
\begin{aligned}
x_{\text{bl\_tf}} &= x_{\text{bl}} + x_{\text{tf}}, \\
y_{\text{bl\_tf}} &= y_{\text{bl}} + y_{\text{tf}}, \\
z_{\text{bl\_tf}} &= z_{\text{bl}} + z_{\text{tf}}, \\
q_{x_{\text{bl\_tf}}} &= q_{x_{\text{bl}}} \cdot q_{x_{\text{tf}}}, \\
q_{y_{\text{bl\_tf}}} &= q_{y_{\text{bl}}} \cdot q_{y_{\text{tf}}}, \\
q_{z_{\text{bl\_tf}}} &= q_{z_{\text{bl}}} \cdot q_{z_{\text{tf}}}, \\
q_{w_{\text{bl\_tf}}} &= q_{w_{\text{bl}}} \cdot q_{w_{\text{tf}}}.
\end{aligned}
$$

- Transformation from base link to base footprint:

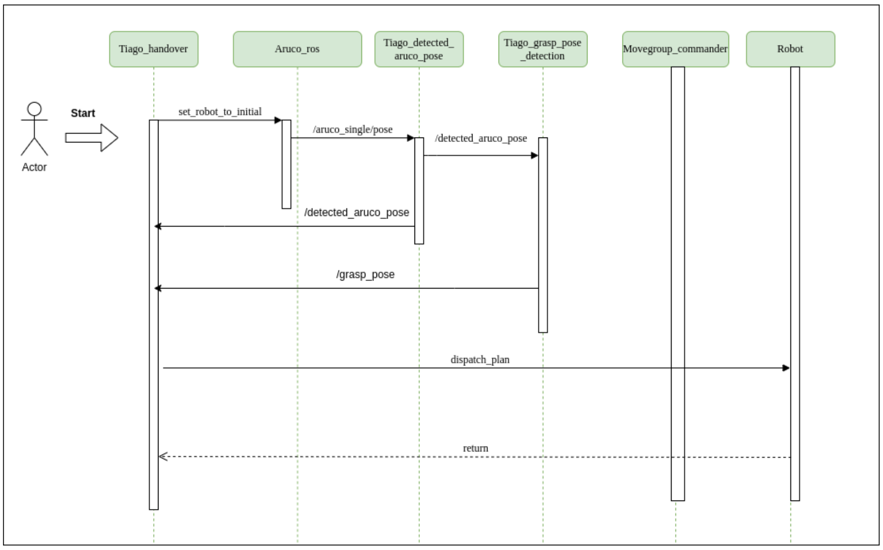


Figure 6.2: Traditional-Cognitive architecture for robot realtime (Sequence Diagram)

Similarly, you'll need to account for the relative pose of the base link with respect to the base footprint. Let's denote the translation and rotation of the base link with respect to the base footprint as $(x_{\text{bf}}, y_{\text{bf}}, z_{\text{bf}})$ and $(q_{x_{\text{bf}}}, q_{y_{\text{bf}}}, q_{z_{\text{bf}}}, q_{w_{\text{bf}}})$, respectively.

The final transformed pose of the ArUco tag in the base footprint frame can be calculated as:

$$\begin{aligned}
x_{\text{bf_bl_tf}} &= x_{\text{bf}} + x_{\text{bl_tf}}, \\
y_{\text{bf_bl_tf}} &= y_{\text{bf}} + y_{\text{bl_tf}}, \\
z_{\text{bf_bl_tf}} &= z_{\text{bf}} + z_{\text{bl_tf}}, \\
q_{x_{\text{bf_bl_tf}}} &= q_{x_{\text{bf}}} \cdot q_{x_{\text{bl_tf}}}, \\
q_{y_{\text{bf_bl_tf}}} &= q_{y_{\text{bf}}} \cdot q_{y_{\text{bl_tf}}}, \\
q_{z_{\text{bf_bl_tf}}} &= q_{z_{\text{bf}}} \cdot q_{z_{\text{bl_tf}}}, \\
q_{w_{\text{bf_bl_tf}}} &= q_{w_{\text{bf}}} \cdot q_{w_{\text{bl_tf}}}.
\end{aligned}$$

The following nodes publish the `/object_pose` and `/grasp_pose` to the node `Tiago_handover`.

6.1.1.2 Tiago handover

Tiago_handover is the node that intakes the pose of the object where the following uses the Moveit library to control the robot to execute the pick and handover of the object, the node uses the movegroup commander to control the

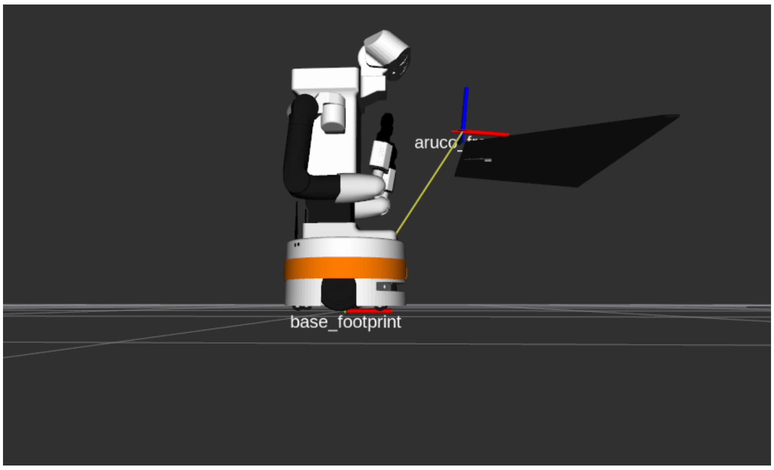
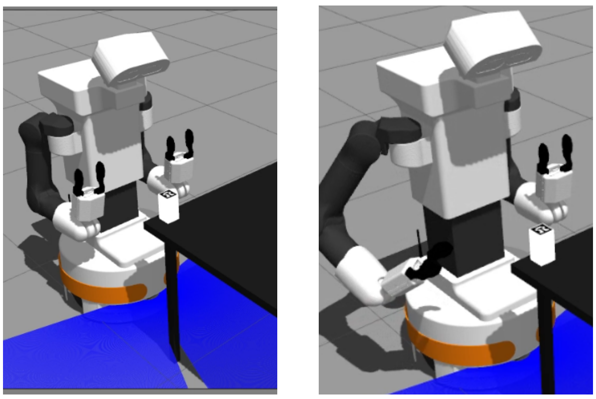


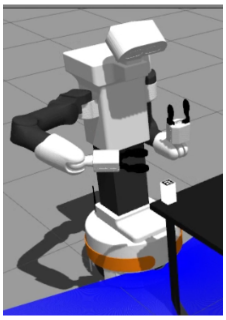
Figure 6.3: Visuals for understanding the working of the tf2_ros library

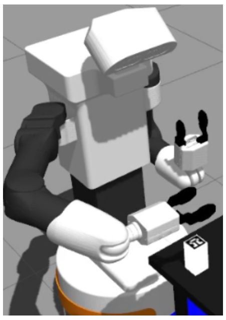
arm_right_torso to control, the robot. At first, the poses were subscribed where the moveit plan executes the robot to reach the z-axis offset of the received pose and make the target reach the following and make the /grasp_pose topic as the final to grasp the object. Once after that, it reaches the certain joint goal for the handover_pose in real time and drops the object in a certain place, and comes back to the certain pose.

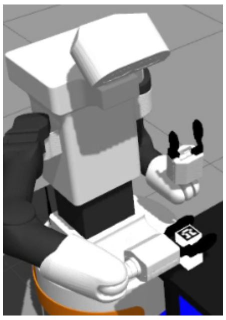
6.1.2 Simulation result of traditional Architecture

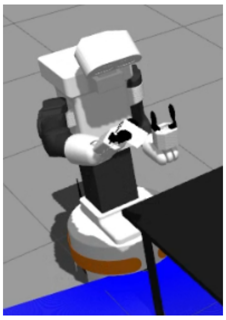


(a) Initial　　(b) Robot-trigger


(a) Plan 1


(b) Plan 2


(a) Plan 3


(b) Plan 4


(a) Plan 5


(b) Plan 6

Figure 6.7: Visuals for understanding ROS architecture

Chapter 7

Quantum-Inspired Cognitive Architecture for robots

7.1 Novel - Cognitive architecture for robots inspired by quantum computing

7.1.1 Single-sensory architecture(Design and working)

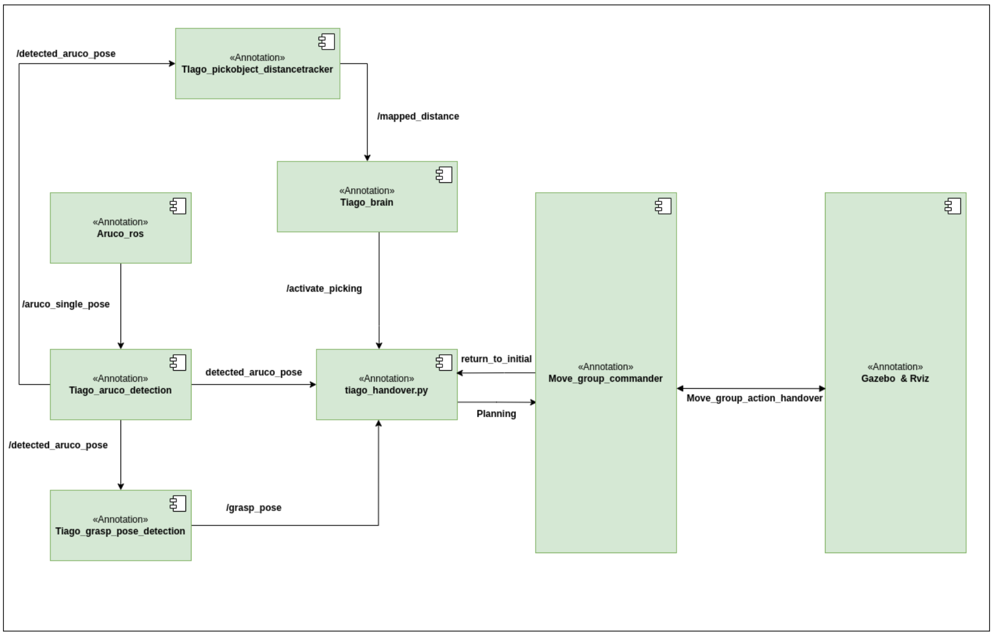


Figure 7.1: Quantum inspired-Cognitive architecture for the robot (Component Diagram)

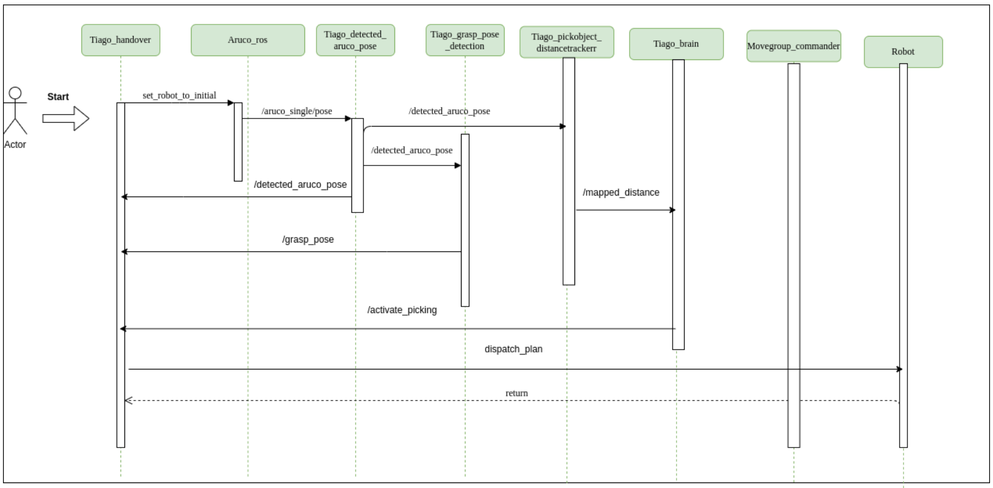


Figure 7.2: Quantum inspired-Cognitive architecture for robot (Sequence Diagram)

7.1.2 Architecture component Description

This chapter describes the implementation of Decision making node in traditional architecture which is the main core of the thesis. And the entire chapter will describe the implementation of single-sensory-based decision-making and Multi-sensory based decision-making with respect to the simulation scenario and its result. For the traditional architecture working just it can be referred to the chapter- 6. But the below subsection describes the additional nodes for the quantum-inspired architecture [52].

7.1.2.1 Sensorial-modality1(`pickobj_dist_calcul`)

This node describes the linear distance calculation of the detected object as the nature of the qubit in the decision-making node has the behavior to represent the uncertain data of the environment with fuzzy logic representation. The script initializes a ROS node named `pickobject_distance_calculator`, sets up a TransformListener, and subscribes to the topic `/detected_aruco_pose`.

In the callback function, it transforms the position from the `xtion_rgb_frame` to the robot's `base_footprint` frame and the received pose to the `base_footprint` frame using TF2.

Then, it calculates the Euclidean distance between the two points in 3D space using the distance formula:

$$\text{distance} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$

Where (x_1, y_1, z_1) are the coordinates of the `xtion_rgb_frame` position, and (x_2, y_2, z_2) are the coordinates of the received pose in the `base_footprint` frame.

The script sets the minimum distance (min_distance) and maximum distance (max_distance) values, which define the range for mapping the calculated distance.

To map the distance to a value between -1 and 1 with a center point of 0, it performs the following mapping:

$$\text{mapped\_value} = \frac{\text{distance} - \text{min\_distance}}{\text{max\_distance} - \text{min\_distance}} \times 2 - 1$$

This maps the distance between min_distance and max_distance to the range $[-1, 1]$, with 0 being the center point. It then takes the absolute value of the mapped value to ensure it remains positive and clamps it between 0 and 1.

Finally, the script publishes the fuzzy representation of the object mapping value on the topic `/pickobject_mapped_distance` and the original distance on the topic `/distance`. Now let's have a understanding about its working that the

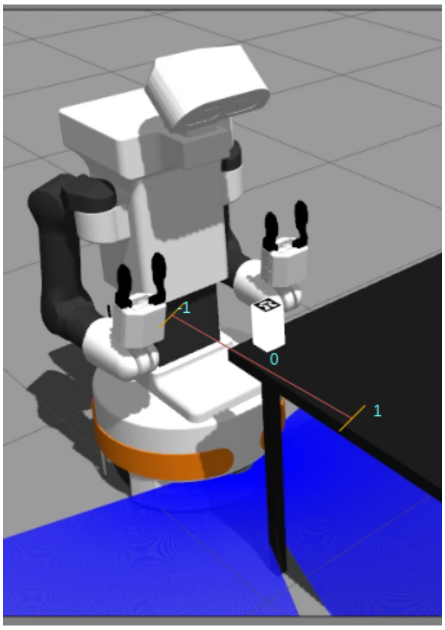


Figure 7.3: Linear distance calculation -Fuzzy based Knowledge representation

7.1.2.2 Decision Node working (`Tiago_brain`)

The decision node where the following is developed with the Q-robot python package which has been described in the 5.3. The current description is about

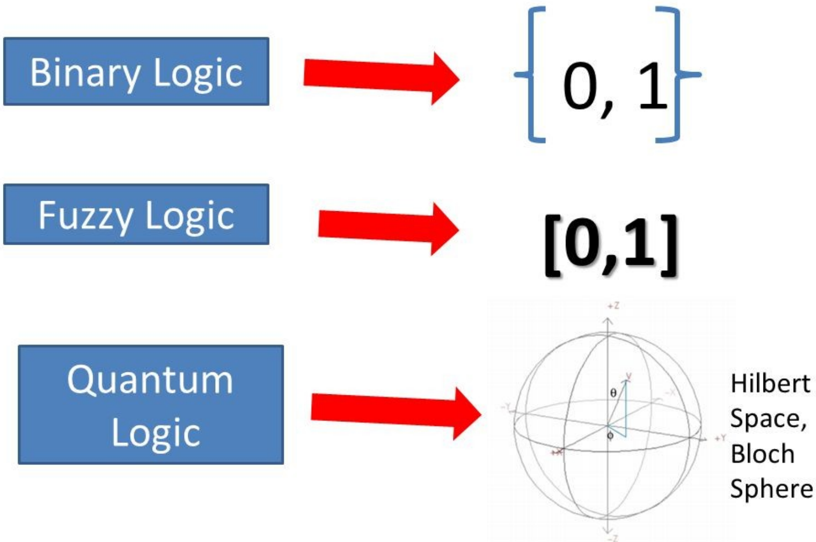


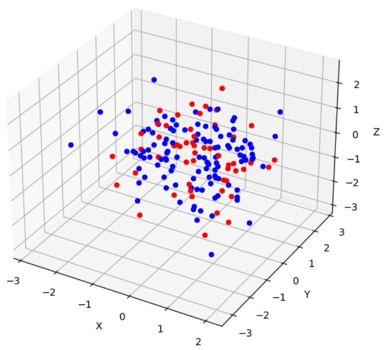
Figure 7.4: Encoding Fuzzy data logic

the single sensory modality working where we have encoded the uncertain data of the detected aruco object from the previous node 7.1.2.1. The following information is encoded in the qubit Hilbert state based on the fuzzy knowledge representation. Once the encoding data is, the superposition measurement has been taken with the single-shot measurement and the decision has been made with the state representing the presence and absence of the object with qubit output. Once the decision has been made as we have used the boolean-based approach in the script True or false is published from the `Tiago_brain` node to the `Tiago_handover node`. And once after that, the robot gets triggered and does the following work once the decision is received. Let's have a look based on the quantum mechanic's calculation for understanding purposes. In this document, we will discuss a quantum-inspired decision-making scenario based on the provided Python script.

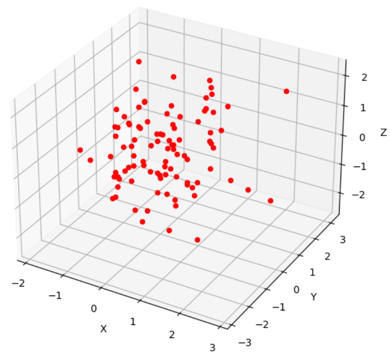
The fuzzy value encoding is represented as follows: If the mapped distance is between 0.1 and 0.3, the fuzzy value is 0.3:

$$|0\rangle \text{ (ket 0) represents the fuzzy value "0.3"}$$
$$|1\rangle \text{ (ket 1) represents the fuzzy value "0.5"}$$
$$|\psi\rangle = \sqrt{0.7}|1\rangle + \sqrt{0.3}|0\rangle \text{ (for the fuzzy value "0.7")}$$

The script measures the quantum unit output after processing the input fuzzy value using the AngularModel. The decision-making threshold is set to 0.5, which means that if the measured quantum unit output is less than or equal to 0.5, the script decides that the object is present (True), otherwise, it decides the object is not present (False).When the quantum unit output is less than or equal to 0.5, it means the measured fuzzy value is closer to "0.3" (which is the fuzzy value representing "Object is not present"). When the quantum unit output is greater than 0.5, it means the measured fuzzy value is closer to "0.7" (which is the fuzzy value representing "Object is present").So, based on the measurement outcome, the script publishes either True (if the object is present) or False (if the object is not present) to the topic `/activate_picking`.



(a) Qubit Behavior

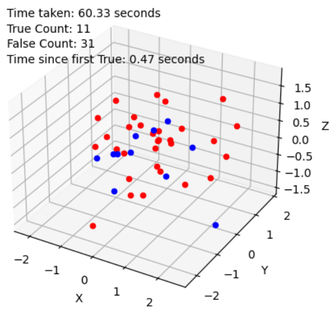


(b) Data-plot for the absence of object

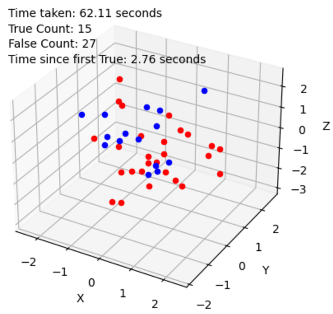
Figure 7.5: Decision-making node data analysis

7.1.3 Result of the single sensory Decision node working in simulation data analysis

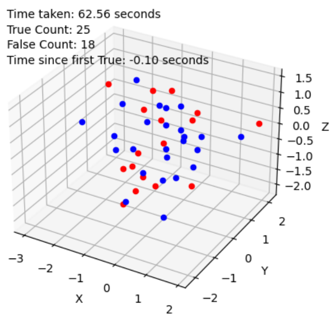
And in the current subsection, we are going to see the working of the decision node as per the [Through the superposition] “The two alternatives exist at the perceptual-cognitive level. Then, they pass at the decisional and conscientious level toward a selection of the two subsisting alternatives. An alternative logical structure is delineated, a structure of the simultaneous YES and NO” (Elio Conte). Now we are going to see the following in our Thesis scenario while the simulation works with the different data analysis to visualize the qubit behavior in decision-making with a 3-dimensional plot.



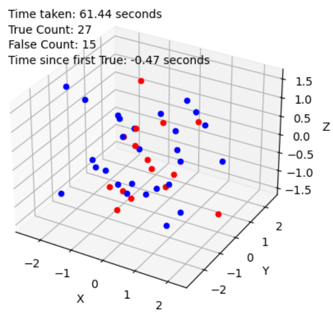
(a) (Query[0.0])



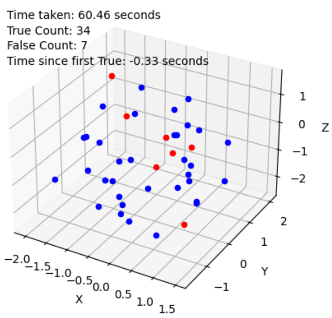
(b) (Query[0.1])



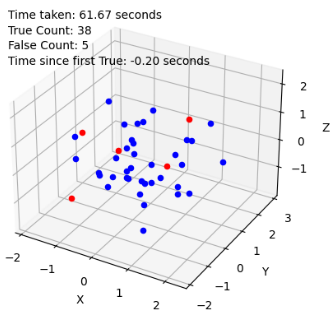
(c) (Query[0.2])



(d) (Query[0.3])

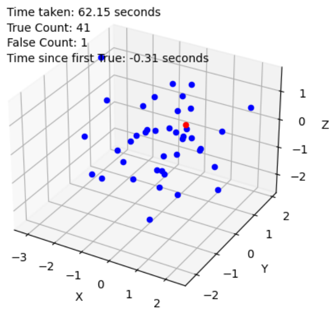


(e) (Query[0.4])

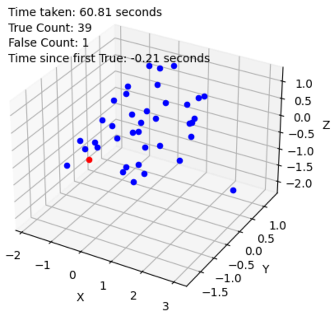


(f) (Query[0.5])

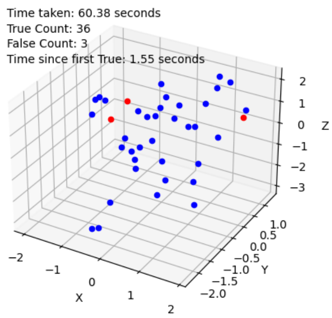
Figure 7.6: [Qubit behavior based on the Query approach -1]



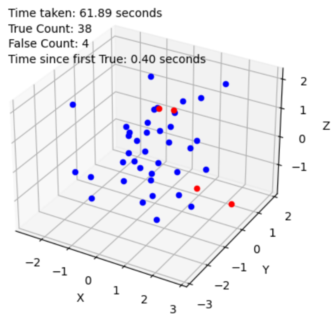
(a) (Query[0.6])



(b) (Query[0.7])



(c) (Query[0.8])



(d) (Query[0.9])

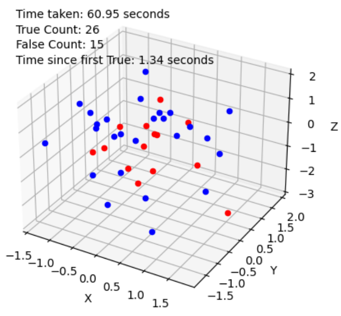


Figure 7.7: [Qubit behavior based on the Query approach -2]

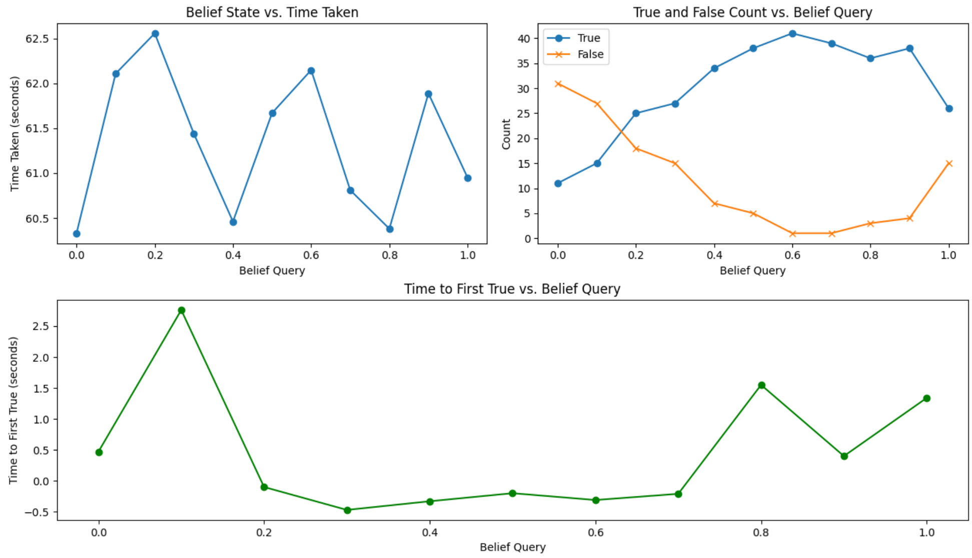


Figure 7.8: Robot belief state sim(Single-sensory architecture)

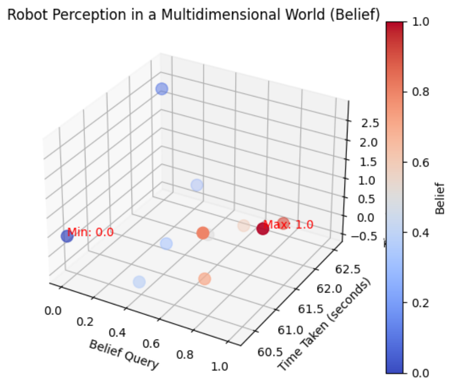


Figure 7.9: Robot belief state sim 3d world (Single-sensory architecture)

7.2 Multi-sensory architecture-Design

This section describes the implementation of Decision making node in traditional architecture which is the other main core of the thesis. And the entire section will describe the implementation of Multi-sensory based decision-making with respect to the simulation scenario and its result. For the traditional architecture working just it can be referred to the chapter- 6. But the below subsection describes the additional nodes for the quantum-inspired architectures.

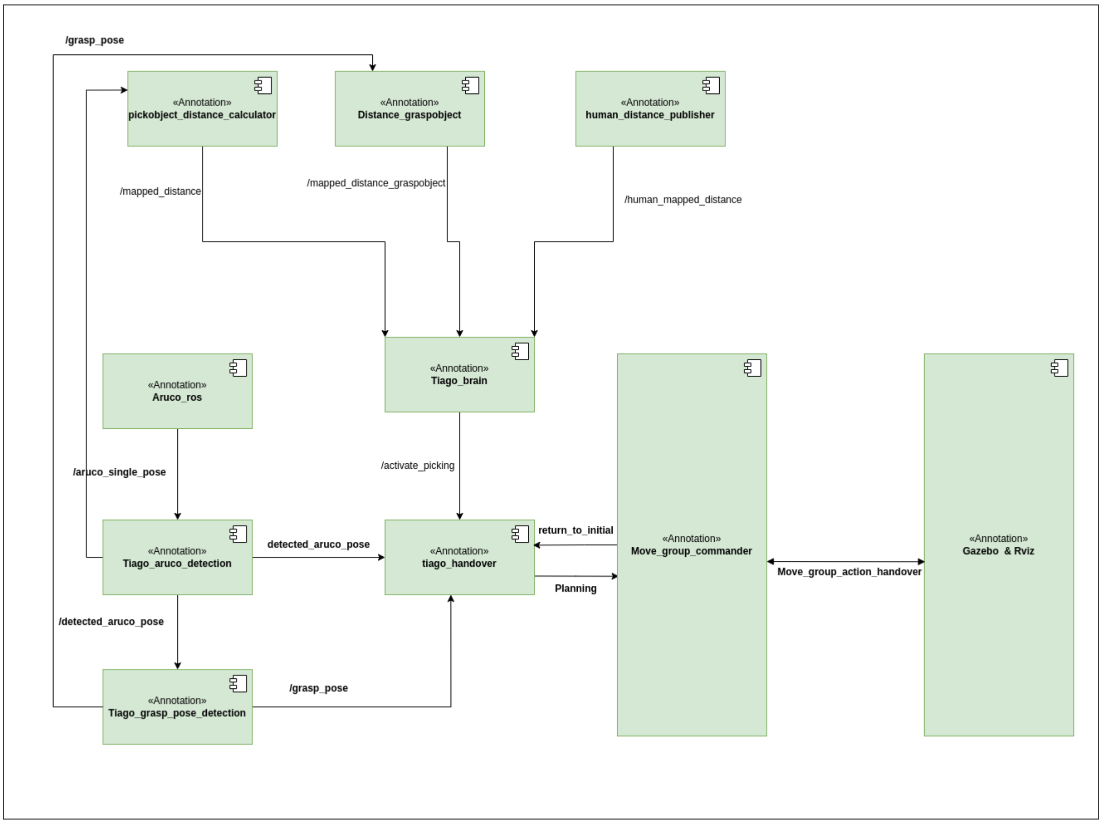


Figure 7.10: Multi-sensory-integration Decision making node(Component diagram)

7.2.1 Multi-sensory Modality Node working

In the following section, we are going to learn the three sensory nodes from the ros side that has been encoded in the 3 dimensions of the Angular model(Decision-making node), and using the Burst technique the decision is taken, And in the current section has the working state of the sensorial-modality-2 and 3 but in the single sensory architecture, the single-sensory modality 1 has been discussed.

7.2.1.1 Sensorial-modality2(`Distance_graspobject`)

This node describes the linear distance calculation of the detected object as the nature of the qubit in the decision-making node has the behavior to represent the uncertain data of the environment with fuzzy logic representation. The script initializes a ROS node named (`Distance_graspobject`), sets up a TransformListener, and subscribes to the topic `/grasp_pose`.

In the callback function, it transforms the position from the `xtion_rgb_frame`

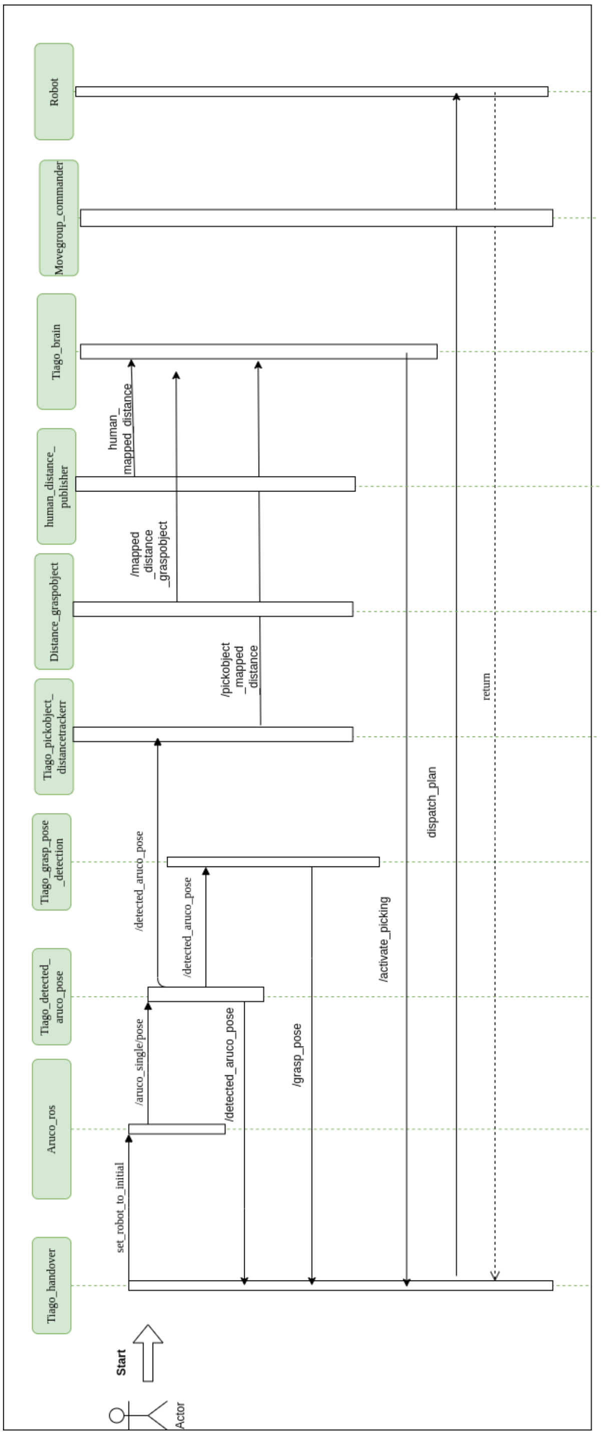


Figure 7.11: Multi-sensory Decision-making Sequence Diagram

to the robot's `base_footprint` frame and the received pose to the `base_footprint` frame using TF2.

Then, it calculates the Euclidean distance between the two points in 3D space using the distance formula:

$$\text{distance} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$

Where (x_1, y_1, z_1) are the coordinates of the `xtion_rgb_frame` position, and (x_2, y_2, z_2) are the coordinates of the received pose in the `base_footprint` frame.

The script sets the minimum distance (min_distance) and maximum distance (max_distance) values, which define the range for mapping the calculated distance.

To map the distance to a value between -1 and 1 with a center point of 0, it performs the following mapping:

$$\text{mapped\_value} = \frac{\text{distance} - \text{min\_distance}}{\text{max\_distance} - \text{min\_distance}} \times 2 - 1$$

This maps the distance between min_distance and max_distance to the range $[-1, 1]$, with 0 being the center point. It then takes the absolute value of the mapped value to ensure it remains positive and clamps it between 0 and 1.

Finally, the script publishes the fuzzy representation of the object mapping value on the topic `/mapped_distance_graspobject` and the original distance on the topic `/distance_graspobject`. The purpose of the script is to represent the knowledge of the robot based on its orientation.

7.2.1.2 Sensorial-modality3(`human_distance_publisher`)

The Node uses the typical shoulder width to calculate how far a human subject is from the Robot. It determines the distance in pixels between the shoulders using posture estimation and transforms that distance into a real-world measurement using a given width and the focal length of the camera. Following the mapping, the distance is reported as ROS topics titled `/human_distance` and `/human_mapped_distance` To calculate the distance from the camera to the person (in centimeters):

$$\text{distance} = \frac{\text{known\_width} \times \text{focal\_length}}{\text{pixel\_width}} \tag{7.1}$$

where:

distance is the distance from the camera to the person in centimeters,
known_width is the known width of the subject (e.g., average shoulder width)
focal_length is the focal length of the camera in pixels, and
pixel_width is the width of the shoulder landmarks in pixels.

To map the distance to a value between -1 and 1 with a center point of 0:

$$\text{mapped\_value} = \left(\frac{\text{distance} - \text{min\_distance}}{\text{max\_distance} - \text{min\_distance}}\right) \times 2 - 1 \tag{7.2}$$

$$\text{mapped\_value} = |\text{mapped\_value}| \tag{7.3}$$

$$\text{mapped\_value} = \min(1, \text{mapped\_value}) \tag{7.4}$$

where:

mapped_value is the mapped value between -1 and 1 with 0 as the center,
min_distance is the minimum distance in centimeters, and
max_distance is the maximum distance in centimeters.

based upon the above image reference 7.2.1.2, the calculation of linear distance

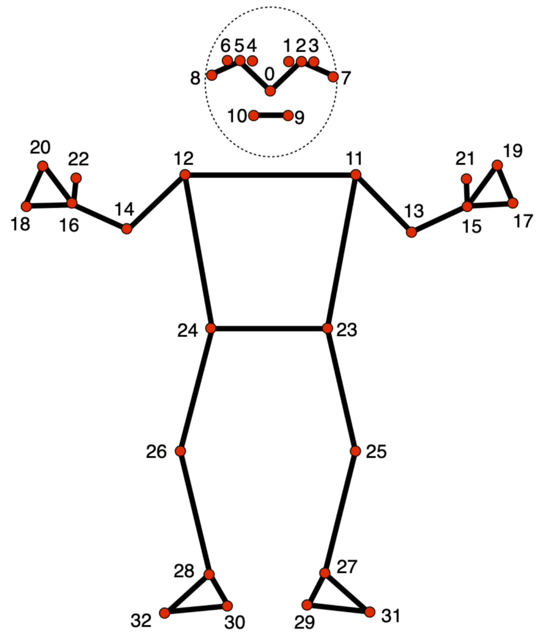


Figure 7.12: Mediapipe-pose landmarks

in the and pose estimation landmarks can be seen as the reference.

7.2.2 Quantum Amplitude Amplification for Fuzzy Value Identification

Before getting into the discussion about multi-sensory decision making lets us have the following working mathematical basis to have a clear understanding,how a single qubit system works in the multi-sensory encoding in the Hilbert state Step 1: Prepare the Initial State Start with a single qubit in the $|0\rangle$ state (the default state). You can represent this in the Dirac notation as $|\psi\rangle = |0\rangle$.

Step 2: Encode Fuzzy Values To encode the fuzzy values into the single qubit system, you can use rotations. For each fuzzy value, apply a rotation to the initial state:

$$\begin{aligned}&\text{For the fuzzy value } 0.4, \quad \text{apply a rotation of } \theta = 2\arccos\left(\sqrt{0.4}\right)\\ &\text{around the Y-axis:} \quad R_y(\theta)|\psi\rangle\\ &\text{For the fuzzy value } 0.7, \quad \text{apply a rotation of } \theta = 2\arccos\left(\sqrt{0.7}\right)\\ &\text{around the Y-axis:} \quad R_y(\theta)|\psi\rangle\\ &\text{For the fuzzy value } 0.8, \quad \text{apply a rotation of } \theta = 2\arccos\left(\sqrt{0.8}\right)\\ &\text{around the Y-axis:} \quad R_y(\theta)|\psi\rangle\end{aligned}$$

Step 3: Amplitude Amplification To amplify the amplitude of the state corresponding to the fuzzy value you are interested in (let's say, 0.7), you can perform the following steps:

- Apply the Hadamard gate (H) to the qubit: $\quad HR_y(\theta)|\psi\rangle$
- Define the oracle O that flips the sign of the state $|0\rangle$
- if it corresponds to the fuzzy value you want to identify: $\quad OHR_y(\theta)|\psi\rangle$
- Apply the Hadamard gate again: $\quad HOHR_y(\theta)|\psi\rangle$
- Apply the Grover diffusion operator D: $\quad D = 2|0\rangle\langle 0| - I$, where I is the identity operator.

- Then, apply D to the state: $DHOHR_y(\theta)|\psi\rangle$

Repeat the amplitude amplification process (D H O H) several times (typically $O(\sqrt{N})$ times, where N is the number of possible states) to maximize the amplitude of the state corresponding to the fuzzy value you are interested in (in this case, 0.7).

Step 4: Measurement Perform a single-shot measurement on the qubit. The state with the highest amplitude after the amplitude amplification process will have a higher probability of being measured, and thus, it will give you the decision (0 or 1) corresponding to the fuzzy value.

7.2.3 Multi-sensory Decision Node working

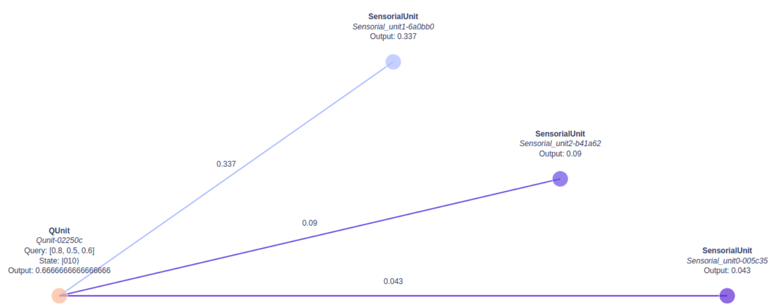


Figure 7.13: Multi-sensory Graph

The decision node where the following is developed with the Q-robot python package which has been described in the 5.3. The current description is about the Multi-sensory modality working where we have encoded the uncertain data of the detected aruco object from the previous node 7.1.2.1. The following information is encoded in the qubit Hilbert state based on the fuzzy knowledge representation. Once the encoding data is, the superposition measurement has been taken with the single-shot measurement and the decision has been made with the state representing the presence and absence of the object with qubit output. Once the decision has been made as we have used the boolean-based approach in the script True or false is published from the `Tiago_brain` node to the `Tiago_handover node`. And once after that, the robot gets triggered and does the following work once the decision is received. Let's have a look based on the quantum mechanic's calculation for understanding purposes. In this document, we will discuss a quantum-inspired decision-making scenario based on the provided Python script. To represent the encoding of fuzzy logic values in a single qubit system using a Bloch sphere plot, we can utilize the Bloch vector to illustrate the quantum state. The Bloch vector provides a geometrical representation of the quantum state on the Bloch sphere.

For each fuzzy value, we'll calculate the corresponding Bloch vector using the formula:

$$\text{Bloch vector} = \begin{bmatrix} 2\,\mathrm{Re}(\sqrt{p}) \\ 2\,\mathrm{Im}(\sqrt{p}) \\ 1-2p \end{bmatrix}$$

where p is the probability of measuring the state in the $|1\rangle$ state (the fuzzy value in this case). Once we have the Bloch vector, we can plot it on the Bloch sphere.

The fuzzy value encoding is represented as follows: If the mapped distance is between 0.1 and 0.3, the fuzzy value is 0.3: To encode the fuzzy logic values in a single qubit system and make a decision (true or false) based on the fuzzy values, you can use the amplitudes of the quantum state to represent the fuzzy probabilities. The general form of a single qubit state is:

$$|\psi\rangle = \sqrt{p}|1\rangle + \sqrt{1-p}|0\rangle$$

where p is the probability of measuring the state in the $|1\rangle$ state, and $(1-p)$ is the probability of measuring the state in the $|0\rangle$ state.

You can map the fuzzy logic values to the probabilities of obtaining the $|1\rangle$ state as follows:

For fuzzy value "0.3":

$$p = 0.3$$

$$|\psi\rangle = \sqrt{0.3}|1\rangle + \sqrt{0.7}|0\rangle$$

For fuzzy value "0.5":

$$p = 0.5$$

$$|\psi\rangle = \sqrt{0.5}|1\rangle + \sqrt{0.5}|0\rangle$$

For fuzzy value "0.7":

$$p = 0.7$$

$$|\psi\rangle = \sqrt{0.7}|1\rangle + \sqrt{0.3}|0\rangle$$

For fuzzy value "0.2":

$$p = 0.2$$

$$|\psi\rangle = \sqrt{0.2}|1\rangle + \sqrt{0.8}|0\rangle$$

For fuzzy value "0.4":

$$p = 0.4$$
$$|\psi\rangle = \sqrt{0.4}|1\rangle + \sqrt{0.6}|0\rangle$$

For fuzzy value "0.8":

$$p = 0.8$$
$$|\psi\rangle = \sqrt{0.8}|1\rangle + \sqrt{0.2}|0\rangle$$

To make a decision (true or false) based on the fuzzy value encoded in the quantum state, you need to perform a measurement on the qubit. The measurement will collapse the quantum state, and you will obtain either the outcome $|0\rangle$ or $|1\rangle$ with probabilities determined by the amplitudes in the state.

For example, if you measure the state $|\psi\rangle = \sqrt{0.3}|1\rangle + \sqrt{0.7}|0\rangle$, there is a 30% chance of getting the outcome $|1\rangle$ (representing fuzzy value "0.3") and a 70% chance of getting the outcome $|0\rangle$ (representing fuzzy value "0.7").

Similarly, for other states, you can interpret the measurement outcome as true (fuzzy value associated with $|1\rangle$) or false (fuzzy value associated with $|0\rangle$) based on the probabilities obtained from the measurement.

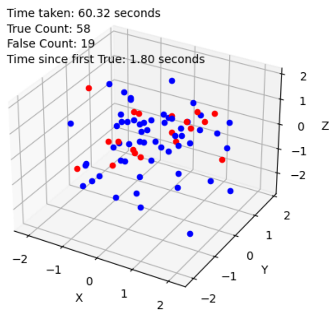
Keep in mind that due to the probabilistic nature of quantum measurement, you might need to repeat the measurement multiple times and analyze the statistics to make an informed decision based on fuzzy logic values.

The script measures the quantum unit output after processing the input fuzzy value using the AngularModel. The decision-making threshold is set to 0.5, which means that if the measured quantum unit output is less than or equal to 0.5, the script decides that the object is present (True), otherwise, it decides the object is not present (False).

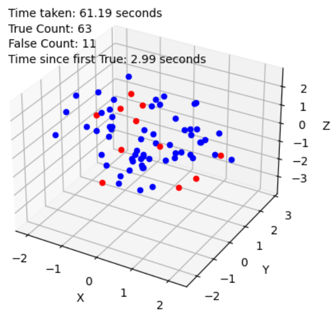
When the quantum unit output is less than or equal to 0.5, it means the measured fuzzy value is closer to "0.3" (which is the fuzzy value representing "Object is not present"). When the quantum unit output is greater than 0.5, it means the measured fuzzy value is closer to "0.7" (which is the fuzzy value representing "Object is present").

So, based on the measurement outcome, the script publishes either True (if the object is present) (object orientation is ok) (Human is coming to handover an object) or False (if the object is not present) (if the object orientation is not ok) (Human is not coming to handover an object) to the topic `/activate_picking`.

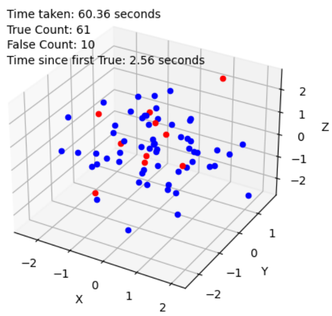
7.2.4 Result of the Multi-sensory decision node working in simulation data analysis



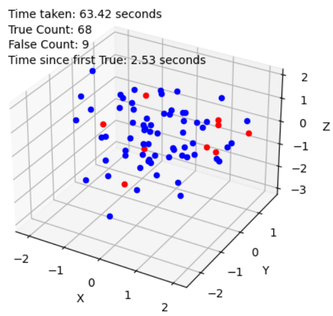
(a) (Multi-query[0.0])



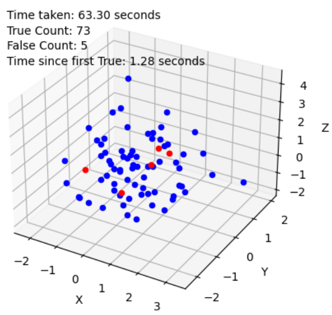
(b) (Multi-query[0.1])



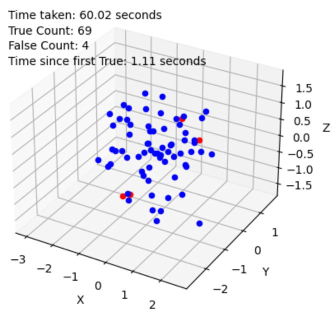
(c) (Multi-query[0.2])



(d) (Multi-query[0.3])

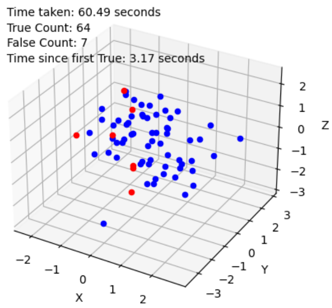


(e) (Multi-query[0.4])

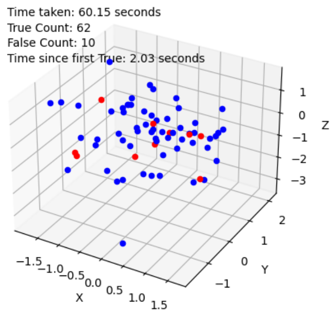


(f) (Multi-query[0.5])

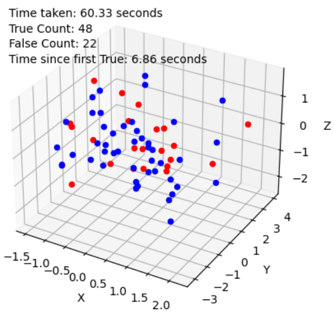
Figure 7.14: [Multi-sensory Qubit behavior based on the Query approach -1]



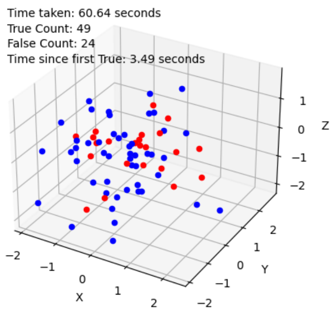
(a) (Multi-query[0.6])



(b) (Multi-query[0.7])



(c) (Multi-query[0.8])



(d) (Multi-query[0.9])

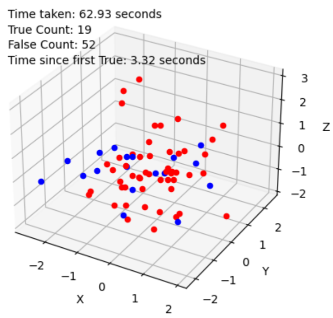


Figure 7.15: [Multi-sensory Qubit behavior based on the Query approach -2]

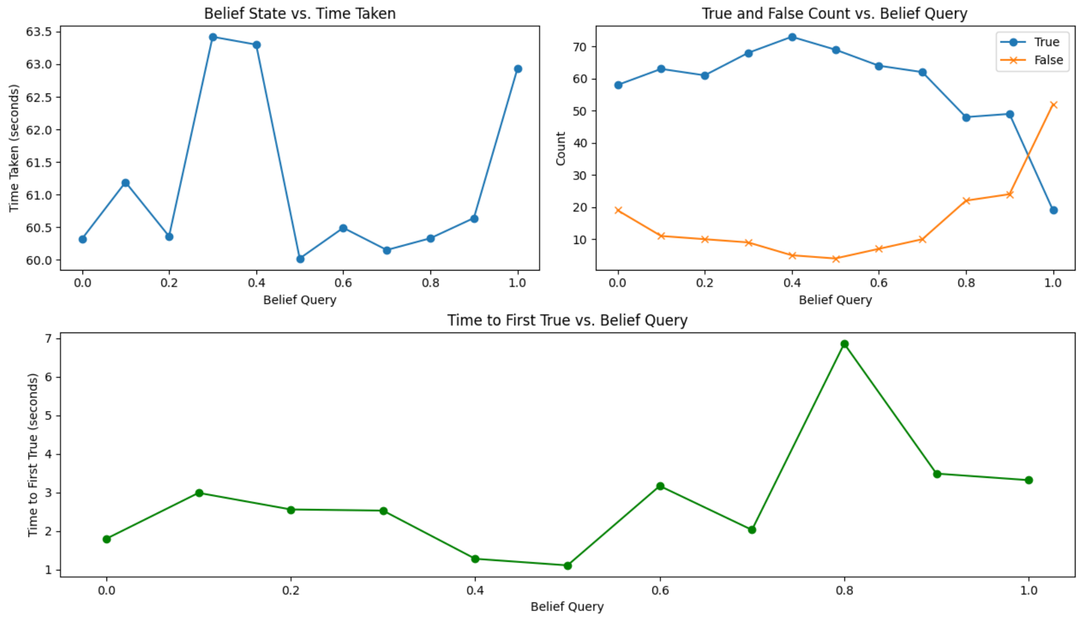


Figure 7.16: Robot belief state sim(Multi-sensory architecture)

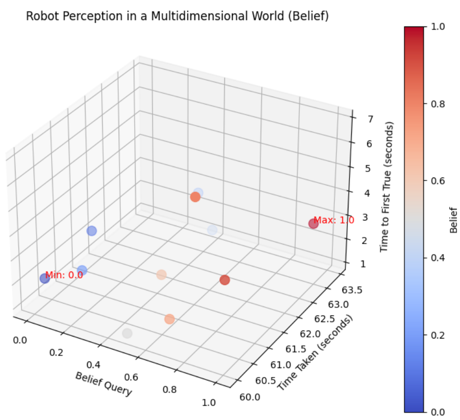


Figure 7.17: Robot belief state sim-3d world(Multi-sensory architecture)

Chapter 8

Experiments and results

8.1 Real-time robot Deployment analysis for Tiago ++ robot

After a detailed discussion about the thesis working from scratch from the initial chapters, currently, we are going to discuss the entire working implementation in real-time with the robot Tiago ++.At first 8.1.1 the real-time scenario of the Handover task. In our Thesis scenario, we have a scenario that is Human-robot interaction with the cognitive scenario focused on the Handover task. But the entire work has been discussed in the previous chapter with the simulation working. But the following work has been deployed in real-time.

8.1.1 Cognitive Architecture working-Table pick and handover scenario

In our real-time work, we have a handover scenario with the tiago++ robot and the human and multiple objects with the aruco marker placed samples have been tested but for the example, the scenario can refer to the 8.1 which describes the complete working scenario which has been focused in the thesis scenario.where a human place the object In front of the tiago robot camera range based on the classical approach the aruco pose detected and the object has been grasped and reached for the certain goal to handover.

(a) Scene-1

(b) Scene-2

(c) Scene-3

(d) Scene-4

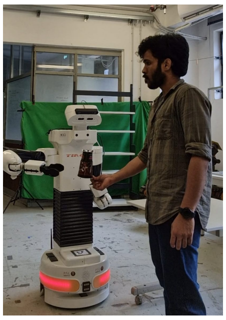
(e) Scene-5

(f) Scene-6

Figure 8.1: Traditional-Cognitive architecture for robot realtime

8.1.2 Cognitive Architecture working-realtime handover scenario

A person is holding an object with an ArUco identifier in this handover scenario. The ArUco marking on the object is recognized by the robot's vision system. After being identified, the robot checks the marker's ID to make sure it matches a recognized object. When validation is successful, the robot's mechanism is activated. It takes the item with care out of the person's hand. The robot then completes duties in accordance with the specifications of the item after the handover. The robot informs the human that the handover was successful. The robot can handle a variety of objects thanks to the person's ability to provide them with distinctive ArUco marks. The robot handles the objects with safety and accuracy throughout the operation. When a mysterious marking is found, the robot looks for further information from the



(a) Scene 1

(b) Scene 2

(c) Scene 3

Figure 8.2: Realtime-handover scenario

8.1.3 Quantum computing inspired Cognitive Architecture working-realtime

We have two quantum-inspired architectures: one is single-sensory and the other is multi-sensory Architecture. In real-time, it reflects the above scenario for the working of the handover scenario which we have discussed in the above sections, but the decision-making has been processed with the Qiskit-based approach, where both workings encode the robot's uncertain perceived data into the qubit Hilbert state with the fuzzy-based knowledge representation, and the outcome as the decision. And the robot gets triggered based on that and completes the scenario.

The scenario described involves two quantum-inspired architectures, one single-sensory and the other multi-sensory, both of which encode uncertain perceived data into qubit Hilbert states using fuzzy-based knowledge representation. The decision-making process is performed using the Qiskit-based approach, and the outcome of this decision determines the robot's actions in real time.

Let's break down the steps in this scenario:

1. Quantum-Inspired Architectures:
   - Single-Sensory Architecture: This design most likely processes information from a single sensory input, i.e., the linear distance between the robot and the observed item as determined by the robot's vision.
   - Multi-Sensory Architecture: This architecture is designed to handle data from multiple sensory inputs, combining information from different sensors to create a more comprehensive representation of the environment. The following is the linear distance between the robot and the detected object, the Robot and orientation of the object, and the robot and the person handover object
2. Uncertain Perceived Data and Fuzzy-Based Knowledge Representation:
   - Uncertain Perceived Data: In the real-world scenario that is handover task in our case, sensor data can often be noisy, imprecise, or uncertain. The quantum-inspired architectures in this scenario likely deal with such uncertain data from the sensors.7.2.1.2
   - Fuzzy-Based Knowledge Representation: Fuzzy logic is a mathematical approach that deals with uncertainty and imprecision. It's commonly used in AI and control systems to handle vague or ambiguous data. using that we are defining the uncertain data.

3. Encoding Perceived Data into Qubit Hilbert States: Quantum computing uses qubits as the fundamental unit of information as we are using q-robot package which is a single-qubit system. The quantum-inspired architectures encode the uncertain data from sensors into the quantum state of qubits using Hilbert space representation. This process allows for quantum parallelism and superposition, potentially enabling more efficient and powerful computations.

4. Qiskit-Based Decision-Making: Qiskit is an open-source quantum computing framework developed by IBM. It provides tools for working with quantum circuits, algorithms, and quantum simulators. In this scenario, the decision-making process uses Qiskit to perform computations and make decisions based on the encoded data in qubits.

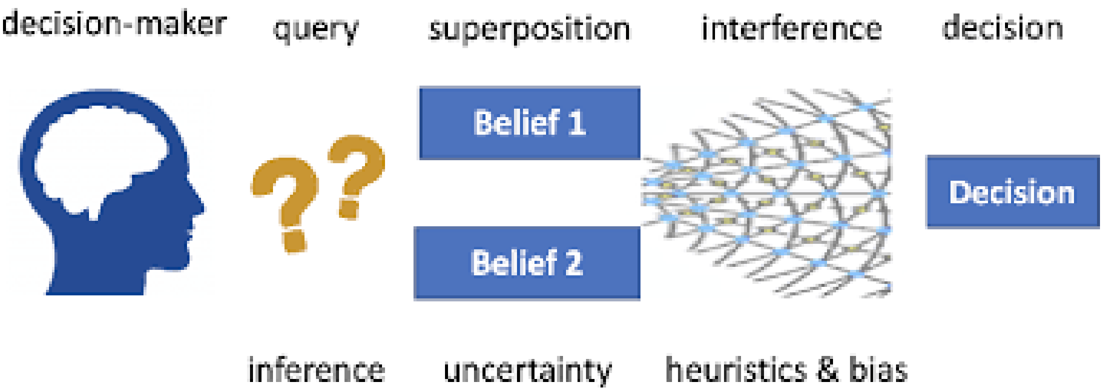


Figure 8.3: Decision-system

5. Real-Time Decision-Making and Action: Once the decision is made using the Qiskit-based approach, the robot takes appropriate actions based on the outcome. The robot triggers and gets the object from the human in the handover scenario.

Overall, this scenario combines quantum-inspired architectures with fuzzy logic and qubit-based decision-making to enable the robot to process uncertain data efficiently and make informed real-time decisions. It highlights the potential benefits of leveraging quantum computing principles in artificial intelligence and robotics applications.

Table 8.1: Comparison of Classical and Quantum-inspired Architecture

Criteria	**Classical Architecture**	**Quantum-Inspired Architecture**
Processing Paradigm:	The classical architecture relies on conventional algorithms for sensor data handling and decision-making.	The quantum-inspired architecture uses quantum-inspired data processing techniques to handle uncertain or noisy sensor data, leveraging quantum principles like superposition and parallelism for specific computations.
Handling Uncertainty	The classical approach may use traditional methods for uncertainty handling, potentially less efficient with complex and uncertain data.	Quantum-inspired techniques handle uncertain sensor data more efficiently using fuzzy logic and quantum superposition, improving decision-making with noisy or imprecise data.
Decision-Making	The classical decision-maker uses standard algorithms and logic, which may be limited in handling complex problems or large datasets.	The decision-maker, implemented with Qiskit, brings quantum computing capabilities, solving certain problems more efficiently using algorithms like Grover's search and quantum simulations.
Resource Requirements	The classical architecture is less resource-intensive, relying on standard computing infrastructure.	Quantum-inspired components require extra resources like quantum simulators, leading to higher computational and infrastructure demands.
Maturity and Robustness	classical architecture are mature and thoroughly tested due to their widespread use in diverse robotic applications.	Quantum-inspired approaches are relatively new and may be less mature and robust than established classical approaches.
Ease of Development and Debugging	Developing and debugging classical ROS components is easier with familiar programming paradigms and tools.	Quantum-inspired components have a steeper learning curve, and debugging quantum systems can be challenging due to the inherent complexity of quantum computing.
Scalability	Classical architectures scale readily with established computing technologies.	The scalability of quantum-inspired architectures could be limited by current quantum computing constraints.

Table 8.1 will give a comparison of the classical and quantum-inspired architecture by its feasibility based on the criteria of the Architecture developed in the complete thesis work. And the implementation of the Quantum-like perception model was analyzed by measuring the speed of the robot to the handover task based on multiple tests with a query-based approach from 0.0 to 1.0 with a timer analysis. And the following approach results in the development of next-gen robots. The following proposed architecture is to introduce the possibility of perception, cognition, and decision processing followed by quantifying the robot belief state using the uncertainty data from the robot environment.

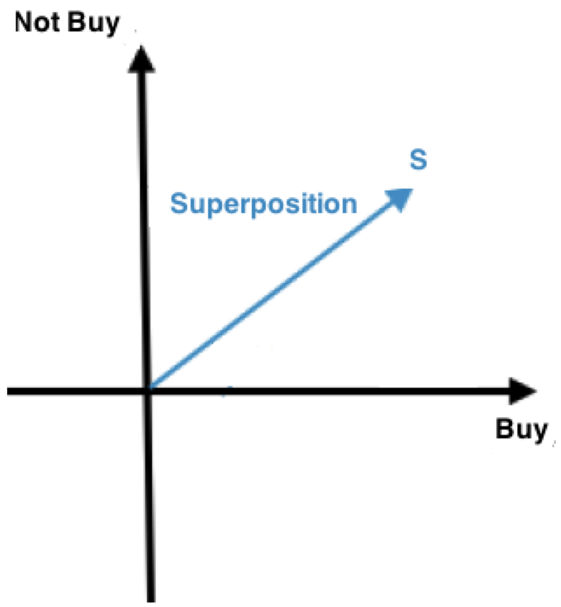
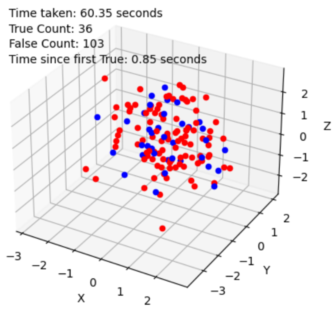


Figure 8.4: Decision-representation of a qubit

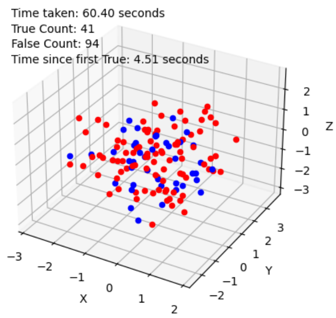
8.2 Result

The following implementation of the quantum-robot architecture in Tiago++ robot for a handover task results in improved perception, Adaptive decision-making, Efficient action planning, Enhanced human-robot interaction, and Modeling uncertainty. where the real-time analysis of the robot. First reference the image 8.7 for the single sensory architecture working with the data analysis for proving the robot belief state with a timer around 1min with multiple query approaches.

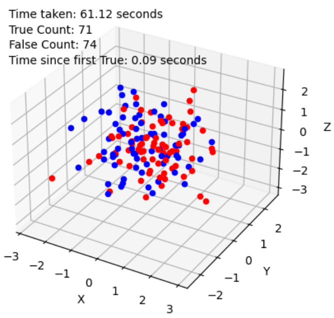
8.2.1 Sensorial unit 1 Robot-belief realtime



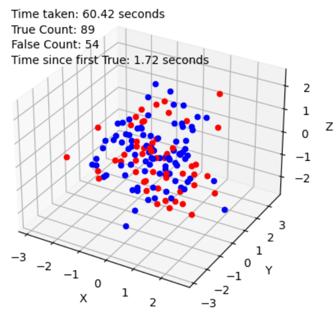
(a) (real-single-query[0.0])



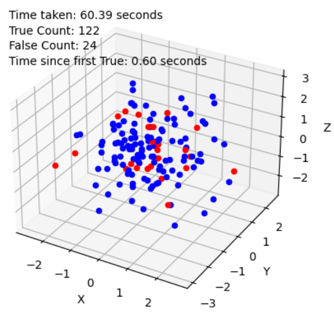
(b) (real-single-query[0.1])



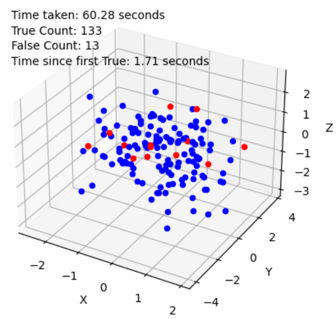
(c) (real-single-query[0.2])



(d) (real-single-query[0.3])

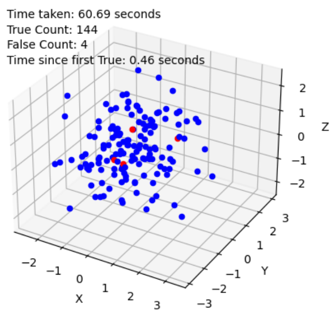


(e) (real-single-query[0.4])

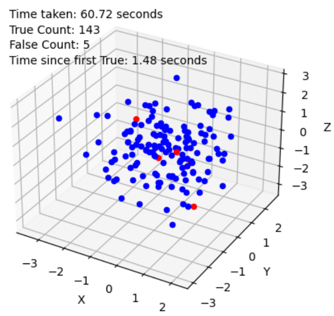


(f) (real-single-query[0.5])

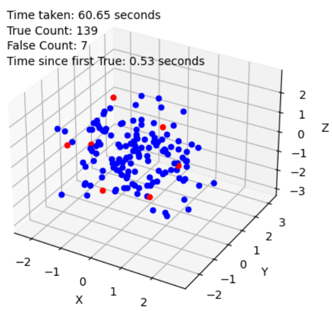
Figure 8.5: [R-1sensory Qubit behavior based on the Query approach -(part-A)]



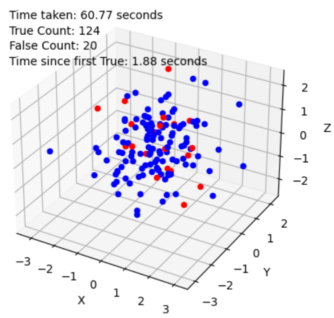
(a) (real-single-query[0.6])



(b) (real-single-query[0.7])



(c) (real-single-query[0.8])



(d) (real-single-query[0.9])

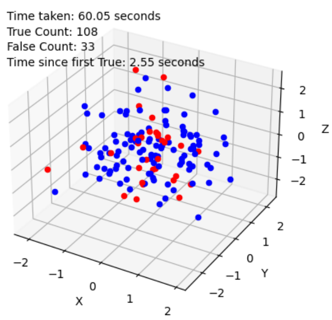


Figure 8.6: [R-1sensory Qubit behavior based on the Query approach -(part-B)]

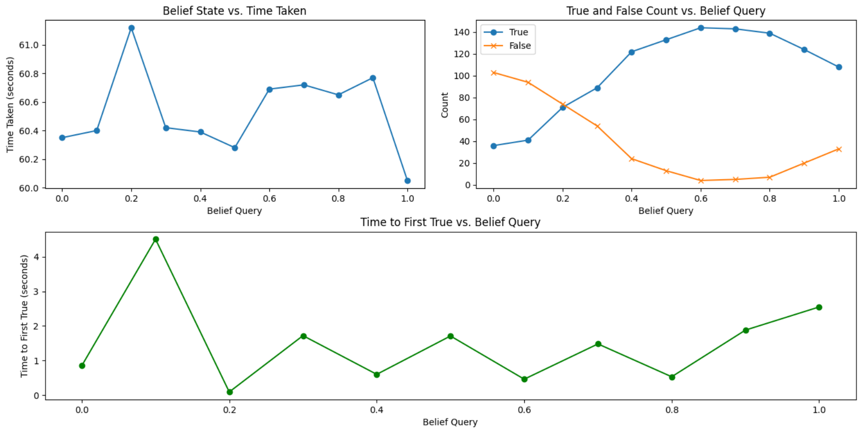


Figure 8.7: Sensorial unit 1 Robot-belief state

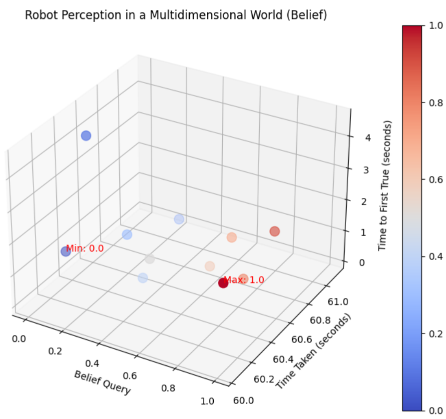


Figure 8.8: sensorial unit 1 Robot-belief state real-time

Chapter 9

Conclusion and Future Work

In summary, the thesis work results in the development of Cognitive architecture for robots inspired by Quantum computing [53], The main contribution of the work is to develop a traditional cognitive architecture and implement a quantum-like perception model to bring the outcome as Cognitive architecture for robots inspired by Quantum computing are the main contributions of this work[54]. And the following architecture has been deployed in the Tiago++ 5.1 robot at first using simulation in Gazebo and later with real-time scenarios including human interaction, the conventional Chapter(6) and quantum-inspired structures Chapter(7) were tested. These designs' testing and analysis produced encouraging results and insightful data for comparative analysis to have the robot's belief state as per the query operators and the following beliefs has been tested in the different scenario. There are many potential areas for development and enhancement in future work. The pick-and-place scenario with the Tiago++ robot is the focus of the thesis, but additional situations with more complicated settings and mobile manipulation tasks might benefit from the use of the quantum-like perception model.

Let's break down the future work listing because Multiple windows have been opened to develop a Cognitive Architecture for robots inspired by quantum computing with a hybrid approach:

- Real-time implementation Comparision test with the traditional and Quantum-inspired architecture working experiment in the handover task needs to be done.
- "I have developed a single qubit system that embeds two words, 'hi' and 'bye,' enabling decision-making. When the user inputs 'hi,' the qubit represents the state 1, and when the input is 'bye,' the qubit represents the state 0. This example demonstrates a quantum state preparation and mea-

surement approach.this is just for the example application i gave but the vision is like the below working "

- Encode the words "up," "down," "left," and "right" onto a qubit using specific numerical values as amplitudes.
- Perform a measurement on the qubit, obtaining a measurement outcome and associated probability amplitudes.
- Plot the measured outcome's corresponding numerical value on a scale. Compare the plotted value with predefined thresholds for each word to classify the requested command.
- Make a decision based on the comparison, assigning the appropriate command (e.g., "up," "down," "left," or "right") based on the threshold comparisons and the measured state.

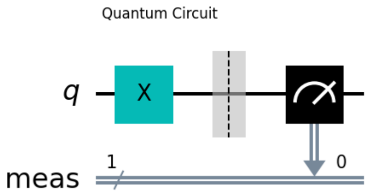


Figure 9.1: Quantum state preparation and measurement 1 approach(Encode="hi as 1")

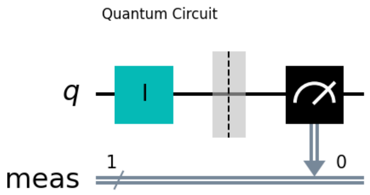


Figure 9.2: Quantum state preparation and measurement 2 approach(Encode="bye as 0")

- Need to study the possibility Quantum inspired reinforcement learning for trajectory planning in the robot [55] [56] [57] [58] [59] [60] [61] [31]. And my suggestion based on For handover tasks, quantum-like reinforcement learning uses quantum-inspired structures to express states, actions, and rewards. The decision-making system searches and exploits the state-action space to discover the best handover techniques by making use of quantum features like superposition and entanglement. The policy is updated depending on rewards obtained using a quantum-like reinforcement learning method, such as Quantum Reinforcement Learning (QRL). The system attempts to create seamless handover transitions, reduce lost objects, and make the best judgments possible through training and evaluation.

- As per the current thesis architecture design we have implemented the Quantum-like perception model with input from three dimensions(thesis work), but further work is needed to expand it to include five-dimensional modalities as the quantum-robot python package has the model capability that has the benchmark for 5-dimensional sensory input in a classical computer system.Need to improve the robot's perceptive skills. Advanced posture estimators [62]also need to replace in place of the existing Aruco-based method to enhance the robot's perception accuracy and capabilities and additionally to include the pre-trained data set that gives the robot knowledge advanced.

- Increasing the human-robot interaction perspective applicability [63] [64]

- A navigation architecture is required for Tiago++'s future development in order for it to be able to adapt to mobile manipulation activities and complex circumstances. The navigation-based pick and place tasks might benefit greatly from the integration of a shared decision-maker that resembles the human brain(not direct related just to have decision-making in the complex scenario), which would be a huge step toward developing more intelligent and adaptable robotic systems. During the initial study, an example scenario with a turtle bot has been developed and the following can be seen in the following [54][65] [66] [67] which would be the reference for future development ideas.

Overall, this study explores the potential of quantum-inspired architectures and implements the sense-plan-act paradigm to establish the foundation for the future generation of robots. We can open the door for more sophisticated and effective robotic systems in a variety of applications sectors to incorporate with the ongoing development and improvement of these models and architecture Quantum-like reinforcement learning, Quantum-like text classification etc which has the focus on the development of the current Architecture to the development of A hybrid Cognitive Architecture for robots inspired by quantum computing.

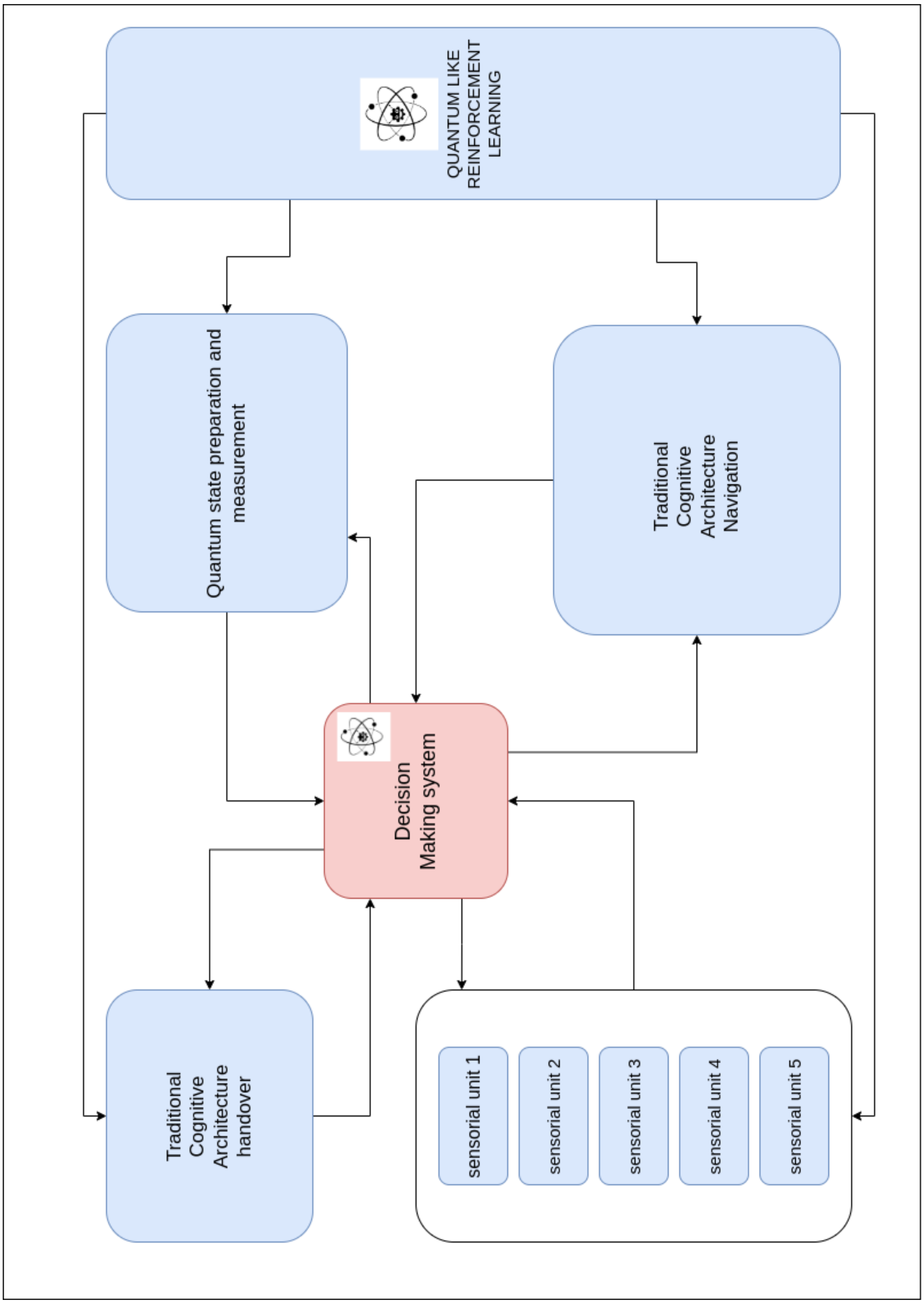


Figure 9.3: A hybrid Cognitive Architecture for Robots inspired by quantum computing

References

[1] P. Pylkkänen, "Weak vs. strong quantum cognition," in *Advances in Cognitive Neurodynamics (IV) Proceedings of the Fourth International Conference on Cognitive Neurodynamics-2013*, pp. 411–418, Springer, 2015. 1, 40

[2] D. Lanza, P. Solinas, and F. Mastrogiovanni, "A preliminary study for a quantum-like robot perception model," 2020. `https://arxiv.org/abs/2006.02771v1`. 2, 10, 35, 36, 37, 44, 51, 52

[3] D. Lanza, P. Solinas, and F. Mastrogiovanni, "Multi-sensory integration in a quantum-like robot perception model," 2020. `https://arxiv.org/abs/2006.16404`. 2, 3, 10, 36, 37, 52

[4] M. F. Pinto, L. M. Honorio, A. Melo, and A. L. Marcato, "A robotic cognitive architecture for slope and dam inspections," *Sensors*, vol. 20, no. 16, p. 4579, 2020. 4

[5] V. Cannella, A. Chella, and R. Pirrone, "A meta-cognitive architecture for planning in uncertain environments," *Biologically Inspired Cognitive Architectures*, vol. 5, pp. 1–9, 2013. 4

[6] C. Burghart, R. Mikut, R. Stiefelhagen, T. Asfour, H. Holzapfel, P. Steinhaus, and R. Dillmann, "A cognitive architecture for a humanoid robot: A first approach," in *5th IEEE-RAS International Conference on Humanoid Robots, 2005.*, pp. 357–362, IEEE, 2005. 4, 35

[7] J. E. Laird, K. R. Kinkade, S. Mohan, and J. Z. Xu, "Cognitive robotics using the soar cognitive architecture.," in *CogRob@ AAAI*, Citeseer, 2012. 4, 35

[8] D. P. Benjamin, D. M. Lyons, and D. W. Lonsdale, "Adapt: A cognitive architecture for robotics.," in *ICCM*, pp. 337–338, 2004. 4, 35

[9] P. Tandon, S. Lam, B. Shih, T. Mehta, A. Mitev, and Z. Ong, *Quantum Robotics: A Primer on Current Science and Future Perspectives.* Synthesis Lectures on Quantum Computing, Springer Cham, 2017. 8

[10] C. Petschnigg, M. Brandstötter, H. Pichler, M. Hofbaur, and B. Dieber, "Quantum computation in robotic science and applications," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 803–810, IEEE, 2019. 8

[11] A. Chella, S. Gaglio, M. Mannone, G. Pilato, V. Seidita, F. Vella, and S. Zammuto, "Quantum planning for swarm robotics," *Robotics and Autonomous Systems*, p. 104362, 2023. 8

[12] J. R. Busemeyer and Z. Wang, "What is quantum cognition, and how is it applied to psychology?," *Current Directions in Psychological Science*, vol. 24, no. 3, pp. 163–169, 2015. 8

[13] J. R. Busemeyer, P. Fakhari, and P. Kvam, "Neural implementation of operations used in quantum cognition," *Progress in biophysics and molecular biology*, vol. 130, pp. 53–60, 2017. 8

[14] E. M. Pothos and J. R. Busemeyer, "Quantum cognition," *Annual review of psychology*, vol. 73, pp. 749–778, 2022. 8

[15] D. Aerts, M. Czachor, and S. Sozzo, "Quantum interaction approach in cognition, artificial intelligence and robotics," *arXiv preprint arXiv:1104.3345*, 2011. 8

[16] C. Lebiere, F. Jentsch, and S. Ososky, "Cognitive models of decision making processes for human-robot interaction," in *Virtual Augmented and Mixed Reality. Designing and Developing Augmented and Virtual Environments: 5th International Conference, VAMR 2013, Held as Part of HCI International 2013, Las Vegas, NV, USA, July 21-26, 2013, Proceedings, Part I 5*, pp. 285–294, Springer, 2013. 10, 11

[17] D. Kalogeras, "The quantum theory in decision making," *Journal of Computations y Modelling*, vol. 3, no. 4, pp. 61–81, 2013. 10, 38

[18] J. M. Yearsley and J. R. Busemeyer, "Quantum cognition and decision theories: A tutorial," *Journal of Mathematical Psychology*, vol. 74, pp. 99–116, 2016. 10, 38

[19] T. Kovalenko, S. Vincent, V. Yukalov, and D. Sornette, "Calibration of quantum decision theory: aversion to large losses and predictability of probabilistic choices," *Journal of Physics: Complexity*, vol. 4, no. 1, p. 015009, 2023. 10, 38

[20] A. U. Igamberdiev and N. E. Shklovskiy-Kordi, "The quantum basis of spatiotemporality in perception and consciousness," *Progress in biophysics and molecular biology*, vol. 130, pp. 15–25, 2017. 10

[21] W. Hu *et al.*, "Empirical analysis of decision making of an ai agent on ibm's 5q quantum computer," *Natural Science*, vol. 10, no. 01, p. 45, 2018. 10

[22] P. Benioff, "Quantum robots and environments," *Physical Review A*, vol. 58, no. 2, p. 893, 1998. 11

[23] M. Huber-Liebl, R. Römer, G. Wirsching, I. Schmitt, M. Wolff, *et al.*, "Quantum-inspired cognitive agents," *Frontiers in Applied Mathematics and Statistics*, vol. 8, p. 909873, 2022. 11

[24] S. Pradhan, A. Padhi, and B. K. Behera, "Design and simulation of an autonomous quantum flying robot vehicle: an ibm quantum experience," *arXiv preprint arXiv:2206.00157*, 2022. 11

[25] S. Chatterjee and A. Das, "An ensemble algorithm using quantum evolutionary optimization of weighted type-ii fuzzy system and staged pegasos quantum support vector classifier with multi-criteria decision making system for diagnosis and grading of breast cancer," *Soft Computing*, vol. 27, no. 11, pp. 7147–7178, 2023. 11

[26] L. Rosendahl, A. S. Bizyaeva, and J. Cohen, "A novel quantum approach to the dynamics of decision making.," in *CogSci*, 2020. 11, 38

[27] S. Han and X. Liu, "An extension of multi-attribute group decision making method based on quantum-like bayesian network considering the interference of beliefs," *Information Fusion*, vol. 95, pp. 143–162, 2023. 11, 38

[28] F. Khoshnoud, M. B. Quadrelli, I. I. Esat, and D. Robinson, "Quantum cooperative robotics and autonomy," *arXiv preprint arXiv:2008.12230*, 2020. 11

[29] P. Benioff, "Some foundational aspects of quantum computers and quantum robots," *Superlattices and microstructures*, vol. 23, no. 3-4, pp. 407–417, 1998. 11

[30] E. Kagan and I. Ben-Gal, "Navigation of quantum-controlled mobile robots," *Recent advances in mobile robotics*, vol. 15, p. 311, 2011. 11

[31] D. Heimann, H. Hohenfeld, F. Wiebe, and F. Kirchner, "Quantum deep reinforcement learning for robot navigation tasks," *arXiv preprint arXiv:2202.12180*, 2022. 11, 89

[32] M. Mannone, V. Seidita, and A. Chella, “Categories, quantum computing, and swarm robotics: A case study,” *Mathematics*, vol. 10, no. 3, p. 372, 2022. 11

[33] V. G. Ivancevic, “Entangled swarm intelligence: Quantum computation for swarm robotics.,” *Mathematics in Engineering, Science & Aerospace (MESA)*, vol. 7, no. 3, 2016. 11

[34] S. Mahanti, S. Das, B. K. Behera, and P. K. Panigrahi, “Quantum robots can fly; play games: an ibm quantum experience,” *Quantum Information Processing*, vol. 18, pp. 1–10, 2019. 11

[35] A. Chella, S. Gaglio, G. Pilato, F. Vella, and S. Zammuto, “A quantum planner for robot motion,” *Mathematics*, vol. 10, no. 14, p. 2475, 2022. 11

[36] J. Clark, T. West, J. Zammit, X. Guo, L. Mason, and D. Russell, “Towards real time multi-robot routing using quantum computing technologies,” in *Proceedings of the International Conference on High Performance Computing in Asia-Pacific Region*, pp. 111–119, 2019. 11

[37] P. Atchade-Adelomou, G. Alonso-Linaje, J. Albo-Canals, and D. Casado-Fauli, “qrobot: A quantum computing approach in mobile robot order picking and batching problem solver optimization,” *Algorithms*, vol. 14, no. 7, p. 194, 2021. 11

[38] A. Y. Kitaev, A. Shen, and M. N. Vyalyi, *Classical and quantum computation.* No. 47, American Mathematical Soc., 2002. 14

[39] D. Aerts and S. Sozzo, “Quantum interference in cognition: Structural aspects of the brain,” *arXiv preprint arXiv:1204.4914*, 2012. 19

[40] E. Manousakis, “Quantum formalism to describe binocular rivalry,” *arXiv preprint arXiv:0709.4516*, 2007. `https://doi.org/10.48550/arXiv.0709.4516`. 30, 35

[41] Y.-P. Gunji, K. Sonoda, and V. Basios, “Quantum cognition based on an ambiguous representation derived from a rough set approximation,” *Biosystems*, vol. 141, pp. 55–66, 2016. 31

[42] E. Conte, A. Y. Khrennikov, O. Todarello, A. Federici, L. Mendolicchio, and J. P. Zbilut, “Mental states follow quantum mechanics during perception and cognition of ambiguous figures,” *Open Systems & Information Dynamics*, vol. 16, no. 01, pp. 85–100, 2009. `https://doi.org/10.1142/S1230161209000074`. 32, 35

[43] V. M. Peri and D. Simon, "Fuzzy logic control for an autonomous robot," in *NAFIPS 2005-2005 Annual Meeting of the North American Fuzzy Information Processing Society*, pp. 337–342, IEEE, 2005. 34

[44] I. Konukseven, B. Kaftanoglu, and T. Balkan, "Multisensor controlled robotic tracking and automatic pick and place," in *Proceedings of the 1997 IEEE/RSJ International Conference on Intelligent Robot and Systems. Innovative Robotics for Real-World Applications. IROS'97*, vol. 3, pp. 1356–1362, IEEE, 1997. 34

[45] L. A. Zadeh, "A summary and update of "fuzzy logic"," in *2010 IEEE International Conference on Granular Computing*, pp. 42–44, IEEE, 2010. 34

[46] L. A. Zadeh, "Knowledge representation in fuzzy logic," *An introduction to fuzzy logic applications in intelligent systems*, pp. 1–25, 1992. 34

[47] Z. Toffano and F. Dubois, "Quantum eigenlogic observables applied to the study of fuzzy behaviour of braitenberg vehicle quantum robots," *Kybernetes*, 2019. 34

[48] X. Li, Z.-y. Liu, V. Zhang, and B. Xu, "Study on robot coalition problem based on novel quantum-inspired ant colony algorithm," 2012. 37

[49] Y. Hu and C. K. Loo, "A generalized quantum-inspired decision making model for intelligent agent," *The Scientific World Journal*, vol. 2014, 2014. 37

[50] M. Mannone, V. Seidita, and A. Chella, "Modeling and designing a robotic swarm: A quantum computing approach," *Swarm and Evolutionary Computation*, vol. 79, p. 101297, 2023. 37

[51] Q. Zhang, *Quantum inspired concepts in decision making.* Missouri University of Science and Technology, 2021. 37

[52] G. G. Rigatos and S. G. Tzafestas, "Parallelization of a fuzzy control algorithm using quantum computation," *IEEE Transactions on Fuzzy Systems*, vol. 10, no. 4, pp. 451–460, 2002. 60

[53] C. Moreira, L. Fell, S. Dehdashti, P. Bruza, and A. Wichert, "Towards a quantum-like cognitive architecture for decision-making," *arXiv preprint arXiv:1905.05176*, 2019. 88

[54] Y. Guru, "Q-robot thesis introductory repository." `https://github.com/yeshwanthguru/q-robot-Thesis-Introductory`, (2023). 88, 90

[55] Y. Li, A. H. Aghvami, and D. Dong, “Intelligent trajectory planning in uav-mounted wireless networks: A quantum-inspired reinforcement learning perspective,” *IEEE Wireless Communications Letters*, vol. 10, no. 9, pp. 1994–1998, 2021. 89

[56] J. Li, R. Zhang, and Y. Yang, “Multi-auv autonomous task planning based on the scroll time domain quantum bee colony optimization algorithm in uncertain environment,” *PloS one*, vol. 12, no. 11, p. e0188291, 2017. 89

[57] D. Dong, C. Chen, and Z. Chen, “Quantum reinforcement learning,” in *International Conference on Natural Computation*, pp. 686–689, Springer, 2005. 89

[58] D. Dong, C. Chen, H. Li, and T.-J. Tarn, “Quantum reinforcement learning,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 38, no. 5, pp. 1207–1220, 2008. 89

[59] S. Wu, S. Jin, D. Wen, and X. Wang, “Quantum reinforcement learning in continuous action space,” *arXiv preprint arXiv:2012.10711*, 2020. 89

[60] F. A. Cárdenas-López, L. Lamata, J. C. Retamal, and E. Solano, “Multiqubit and multilevel quantum reinforcement learning with quantum technologies,” *PloS one*, vol. 13, no. 7, p. e0200455, 2018. 89

[61] D. Dong, C. Chen, J. Chu, and T.-J. Tarn, “Robust quantum-inspired reinforcement learning for robot navigation,” *IEEE/ASME transactions on mechatronics*, vol. 17, no. 1, pp. 86–97, 2010. 89

[62] H. Talbi, M. Batouche, and A. Draa, “A quantum-inspired evolutionary algorithm for multiobjective image segmentation,” *International Journal of Mathematical, Physical and Engineering Sciences*, vol. 1, no. 2, pp. 109–114, 2007. 90

[63] N. Masuyama, C. K. Loo, and N. Kubota, “Quantum-inspired bidirectional associative memory for human–robot communication,” *International Journal of Humanoid Robotics*, vol. 11, no. 02, p. 1450006, 2014. 90

[64] F. Yan, A. M. Iliyasu, and K. Hirota, “Conceptual framework for quantum affective computing and its use in fusion of multi-robot emotions,” *Electronics*, vol. 10, no. 2, p. 100, 2021. 90

[65] D. Dong, C. Chen, J. Chu, and T.-J. Tarn, “Robust quantum-inspired reinforcement learning for robot navigation,” *IEEE/ASME Transactions on Mechatronics*, vol. 17, no. 1, pp. 86–97, 2012. 90

[66] K.-H. Han and J.-H. Kim, "Quantum-inspired evolutionary algorithm for a class of combinatorial optimization," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 6, pp. 580–593, 2002. 90

[67] N. Mishra, R. S. Chandra, B. K. Behera, and P. K. Panigrahi, "Automation of quantum braitenberg vehicles using finite automata: Moore machines," *Quantum Information Processing*, vol. 19, pp. 1–12, 2020. 90

[68] P. D. Bruza, Z. Wang, and J. R. Busemeyer, "Quantum cognition: a new theoretical approach to psychology," *Trends in cognitive sciences*, vol. 19, no. 7, pp. 383–393, 2015.

[69] C. Chen, P. Yang, X. Zhou, and D. Dong, "A quantum-inspired q-learning algorithm for indoor robot navigation," in *2008 IEEE International Conference on Networking, Sensing and Control*, pp. 1599–1603, 2008.

[70] P. Tandon, S. Lam, B. Shih, T. Mehta, A. Mitev, and Z. Ong, "Quantum agent models," in *Quantum Robotics: A Primer on Current Science and Future Perspectives*, pp. 33–46, Springer, 2017.

[71] S. Lipovetsky, "Quantum-like data modeling in applied sciences," *Stats*, vol. 6, no. 1, pp. 345–353, 2023.

[72] A. Khrennikov, "Quantum-like cognition and rationality: biological and artificial intelligence systems," in *Quantum Computing in the Arts and Humanities: An Introduction to Core Concepts, Theory and Applications*, pp. 153–178, Springer, 2022.

[73] A. Meghdadi, M.-R. Akbarzadeh-T., and K. Javidan, "A quantum-like model for predicting human decisions in the entangled social systems," *IEEE Transactions on Cybernetics*, vol. 52, no. 7, pp. 5778–5788, 2022.

[74] P. Benioff, "Quantum robots and quantum computers," *arXiv preprint quant-ph/9706012*, 1997.

[75] T. A. Metzler, L. M. Lewis, and L. C. Pope, "Could robots become authentic companions in nursing care?," *Nursing Philosophy*, vol. 17, no. 1, pp. 36–48, 2016.

[76] R. A. F. Cunha, N. Sharma, Z. Toffano, and F. Dubois, "Fuzzy logic behavior of quantum-controlled braitenberg vehicle agents," in *Quantum Interaction: 11th International Conference, QI 2018, Nice, France, September 3–5, 2018, Revised Selected Papers 11*, pp. 111–122, Springer, 2019.

[77] A. Aly, S. Griffiths, and F. Stramandinoli, "Metrics and benchmarks in human-robot interaction: Recent advances in cognitive robotics," *Cognitive Systems Research*, vol. 43, pp. 313 – 323, 2017.

[78] N. Zioui, Y. Mahmoudi, A. Mahmoudi, M. Tadjine, and S. Bentouba, "A novel quantum-computing-based quaternions model for a robotic arm position," *Int. J. Comput. Intellig. Control*, vol. 12, no. 2, pp. 1–4, 2021.

[79] J.-A. Li, D. Dong, Z. Wei, Y. Liu, Y. Pan, F. Nori, and X. Zhang, "Quantum reinforcement learning during human decision-making," *Nature human behaviour*, vol. 4, no. 3, pp. 294–307, 2020.

[80] N. Meyer, C. Ufrecht, M. Periyasamy, D. D. Scherer, A. Plinge, and C. Mutschler, "A survey on quantum reinforcement learning," *arXiv preprint arXiv:2211.03464*, 2022.