**A logo for a university

Description automatically generated**

*Integrated grasp and motion planning*

**Author:**

Salah Ebrahimpour

**Programme:**

MSc Artificial Intelligence with Business Strategy

**Supervisor:**

Dr. Martin Rudorfer

**Co-Supervisor:**

Dr. Anthony Henry

**Date:**

September 2024

**Abstract**

**Acknowledgement**

Contents

[Chapter 1: Introduction 1](#_Toc177461906)

[1.1. Project background 1](#_Toc177461907)

[1.2. Research Objectives 1](#_Toc177461908)

[1.3. Research Questions/Hypotheses 2](#_Toc177461909)

[1.4. Significance of the Study 3](#_Toc177461910)

[1.5. Scope and Limitations 3](#_Toc177461911)

[1.6. Thesis Structure 3](#_Toc177461912)

[Chapter 2: Literature Review 4](#_Toc177461913)

[2.1. Introduction 4](#_Toc177461914)

[2.2. Overview of Grasp Planning 4](#_Toc177461915)

[2.3. Overview of Motion Planning 7](#_Toc177461916)

[2.3. Integrated Grasp and Motion Planning 10](#_Toc177461917)

[2.4. Existing Methods and Approaches 12](#_Toc177461918)

[Summary of Existing Methods 15](#_Toc177461919)

[Gaps in the Literature 15](#_Toc177461920)

[Summary 15](#_Toc177461921)

[Chapter 3: Problem Description 15](#_Toc177461922)

[Chapter 4: Methods and Theories 17](#_Toc177461923)

[4.1. Introduction 17](#_Toc177461924)

[4.2. Problem Formulation 18](#_Toc177461925)

[4.3. Research Design 18](#_Toc177461926)

[4.4. Algorithms and Frameworks 18](#_Toc177461927)

[4.4.1. Simulation Environment 19](#_Toc177461928)

[4.4.2. RRT\* Algorithm 19](#_Toc177461929)

[4.5. Experimental Setup 19](#_Toc177461930)

[4.6. Metrics for Evaluation 20](#_Toc177461931)

[4.7. Data Collection 20](#_Toc177461932)

[4.8. Testing and Validation 20](#_Toc177461933)

[4.9. Challenges and Adaptations 20](#_Toc177461934)

[Chapter 5: Project Management 21](#_Toc177461935)

[Chapter 6: Evaluation 24](#_Toc177461936)

[6.1 Experimental Results Overview 24](#_Toc177461937)

[6.2 Performance in Benchmark Scenarios 24](#_Toc177461938)

[6.2.1 Planning Time 24](#_Toc177461939)

[6.2.2 Success Rate 25](#_Toc177461940)

[6.3 Comparative Analysis of Algorithms 25](#_Toc177461941)

[6.4 Statistical Analysis 25](#_Toc177461942)

[6.5 Challenges Encountered 25](#_Toc177461943)

[6.6 Final Results 26](#_Toc177461944)

[6.7 Conclusion of Evaluation 26](#_Toc177461945)

[Chapter 7: Conclusion 27](#_Toc177461946)

[References 28](#_Toc177461947)

[Appendix 32](#_Toc177461948)

# Chapter 1: Introduction

## Project background

As robotics continues to evolve, the need for more sophisticated systems capable of autonomous, complex manipulation tasks becomes ever more pressing. Traditionally, the processes of grasp and motion planning have been treated separately in robotic systems. Grasp planning determines how a robot should grip an object based on factors like the object's shape, weight, and texture. Motion planning, on the other hand, focuses on the robot's path and the obstacles it must avoid when executing tasks (Muhayyuddin et al., 2015). However, the separation of these two processes can lead to suboptimal performance, particularly in dynamic environments where grasp configuration can affect motion feasibility and vice versa (Ali & Lee, 2020).

To address these issues, researchers have started exploring integrated approaches, where grasp and motion planning are considered simultaneously. These unified systems aim to enhance efficiency, adaptability, and reliability in robotic tasks. For example, the Grasp-RRT approach represents one such integration, optimizing both grasp and motion planning within the same framework (Ali & Lee, 2020). This research focuses on leveraging and advancing these integrated planning methods to improve robotic manipulation in real-world scenarios.

## Research Objectives

The main objective of this research is to evaluate the performance of the RRT\* algorithm in the context of integrated grasp and motion planning tasks. Specifically, this study aims to assess the algorithm's ability to solve planning tasks within benchmark scenarios introduced by Rudorfer et al., focusing on key performance metrics such as planning time and success rate.

A significant goal of the research is to compare the performance of the RRT\* algorithm against the benchmark results provided in Rudorfer et al.’s paper, particularly in relation to other algorithms such as J+-RRT and IK-RRT. This comparison seeks to identify the strengths and weaknesses of RRT\*, highlighting how well it performs in terms of runtime and success rate under the same conditions as the baseline algorithms.

Additionally, the research aims to analyze the adaptability of the RRT\* algorithm when applied to different environmental configurations, including variations in object placements and confined spaces. This will provide insights into how RRT\* handles dynamic environments and complex obstacle layouts.

Ultimately, the research seeks to contribute to the development of more efficient integrated grasp and motion planning strategies by offering insights into the applicability of the RRT\* algorithm for real-world robotic tasks. Based on the comparative analysis, this study will also provide recommendations for improving the algorithm’s efficiency and robustness in future applications.

## Research Questions/Hypotheses

This research is guided by several key questions and hypotheses, primarily focusing on the performance of the RRT\* algorithm compared to benchmark algorithms such as J+-RRT and IK-RRT, particularly in terms of planning time and success rate. The performance of RRT\* in motion planning has been widely studied, with previous research demonstrating that RRT\* is asymptotically optimal, meaning that as the number of samples increases, the solution converges to the optimal path with high probability (Gammell et al., 2015; Pérez et al., 2012). This characteristic is especially advantageous in environments with lower obstacle densities, where RRT\* can utilize its sampling efficiency to yield shorter planning times and higher success rates compared to other algorithms that may not exhibit the same level of optimality, such as J+-RRT and IK-RRT (Qureshi & Ayaz, 2015). Studies have shown that in less cluttered environments, RRT\* can outperform its predecessors in both speed and success, thanks to its iterative path refinement process (Tahir et al., 2018). Based on these findings, it is hypothesized that RRT\* will perform competitively with these benchmark algorithms in terms of planning time and success rate.

Moreover, the research aims to explore whether the RRT\* algorithm can efficiently adapt to complex and confined spaces while maintaining a high success rate. RRT\*'s incremental nature allows it to adjust dynamically to environmental changes, making it well-suited for challenging environments (Karaman et al., 2011; Lee & Lee, 2012). However, as the complexity of the space increases, it is hypothesized that the success rate of RRT\* may decline due to the increased difficulty in navigating through dense obstacles. This aligns with research indicating that while RRT\* performs well in simpler environments, its success rate can be hindered in more complex scenarios, where pathfinding becomes more convoluted (Peng et al., 2023).

In conclusion, the RRT\* algorithm shows promising performance when compared to benchmark algorithms, particularly in less cluttered environments. Its ability to adapt to complex spaces is promising, but challenges in dynamic environments highlight the need for optimization. These questions and hypotheses aim to investigate the potential of RRT\* while addressing its limitations in real-world applications.

## Significance of the Study

This study is significant in its potential to advance the field of robotics by providing valuable insights into the performance of the RRT\* algorithm in integrated grasp and motion planning. As robotic systems become more prevalent in sectors such as manufacturing, healthcare, and autonomous services, their ability to perform complex manipulation tasks in dynamic and uncertain environments is increasingly important. This research focuses on assessing the RRT\* algorithm’s effectiveness in planning both grasp and motion concurrently, particularly in terms of planning time, success rate, and adaptability.

By comparing the RRT\* algorithm to established benchmarks, this study offers critical data on its strengths and limitations when applied to real-world scenarios where both grasping and motion must be planned efficiently. These findings have broader implications for improving robotic systems' performance in dynamic environments, contributing to the development of autonomous systems that are more reliable, adaptable, and capable of handling increasingly complex tasks.

Ultimately, the outcomes of this research could enhance the performance of robotic systems in various industries, making them more versatile and effective in executing real-world tasks with greater precision and efficiency.

## Scope and Limitations

This research focuses on the development and evaluation of integrated grasp and motion planning algorithms, specifically in the context of robotic manipulation tasks. The experiments are conducted in simulated environments to ensure repeatability and control over variables. However, while the simulation allows for thorough testing under a variety of conditions, real-world testing of the algorithms may present challenges such as sensor inaccuracies, mechanical limitations of robots, or unexpected environmental factors. These limitations must be addressed in future research to fully realize the practical application of the proposed methods.

## Thesis Structure

This dissertation is organized into several chapters that systematically explore the integration of grasp and motion planning in robotics. The structure is as follows: Chapter 1 provides an introduction, outlining the project background, research objectives, and significance. Chapter 2 reviews the existing literature on grasp and motion planning, focusing on integrated approaches and identifying gaps. Chapter 3 describes the problem formulation, while Chapter 4 details the methodology, including research design, algorithm implementation, and experimental setup. Chapter 5 discusses project management. Chapter 6 presents the evaluation of algorithm performance, and Chapter 7 concludes with a summary of findings and recommendations for future research.

# Chapter 2: Literature Review

## 2.1. Introduction

The field of robotics has seen significant advancements in recent years, particularly in the domain of integrated grasp and motion planning. While motion planning is a fundamental aspect of robotics (Elbanhawi & Simić, 2014), the simultaneous planning of grasping an object and the robot's motion remains a notable challenge (Bütepage et al., 2019). This challenge is further intensified by the need to overcome computational obstacles related to sensing, grasp analysis, motion planning, and the execution of the robot arm's movements (Ichnowski et al., 2020).

In this section, we will explore the historical development and theoretical foundations of grasp planning and motion planning, examining the major theories and models that have shaped these fields. We will then explore the integrated approaches that combine these two planning processes and will discuss the challenges and innovations that have emerged from this integration. By critically analysing the existing literature, we aim to identify gaps and areas for future research, setting the stage for the subsequent chapters of this dissertation. This review will provide the necessary context and background for understanding the contributions and implications of our research on integrated grasp and motion planning.

## 2.2. Overview of Grasp Planning

Grasp planning in robotics has advanced significantly, evolving from basic algorithms for simple objects in controlled environments to more sophisticated approaches capable of handling complex and irregularly shaped objects in dynamic settings. Initially, the focus was on convex objects that a single robot could grasp, aiming to optimize stability and grasp force on the robot's contact points (Zafra-Urrea, 2023). As computational power and sensor technology progressed, research expanded to include the manipulation of objects with robotic hands in dynamic conditions, that lead to the development of various grasp planning algorithms to identify stable grasps (Dharbaneshwer et al., 2020).

**Major Theories and Models**

Grasp planning in robotics is supported by several foundational theories, including Force Closure, Form Closure, and Task-Oriented Grasping. Force Closure emphasizes the robot's ability to resist external forces from any direction, ensuring a stable grip through the application of balanced forces. This theory is essential for tasks that require robustness to external disturbances, such as handling in dynamic environments (Bicchi, 1995). On the other hand, Form Closure focuses on geometric stability, where the object's movement is constrained by the positioning of the contact points, providing a secure grasp without relying solely on applied forces (Mishra et al., 1987).

Task-Oriented Grasping shifts the focus from pure stability to optimizing the grasp for the specific task the robot needs to perform. It considers factors like object manipulation and positioning, making it particularly useful in scenarios where the robot needs to perform a sequence of actions with the object, such as in industrial automation or healthcare robotics (Ciocarlie et al., 2009). This approach has been introduced to compute the most suitable grasp for a given task based on a simplified geometrical and structural description of the object and the task requirements (Prats et al., 2007). Recent advancements have also explored the use of over-segmented meshes and relational databases to improve grasp planning, particularly in regrasp planning scenarios (Wan & Harada, 2017). Precision grasp planning for multi-fingered hands involves analysing grasp stability, synthesis, and object representation (Yan et al., 2019).

Research by Bertoni et al. (2021) introduces a generic grasp planning pipeline that enables transparent and generic grasp planning procedures, automating grasping actions irrespective of the end-effector kinematic structure. Tsuji et al. (2010) demonstrate grasp planning for multi-fingered hands with humanoid robots, emphasizing the importance of effective grasp planning through simulation and experimental results. Rubert et al. (2017) focus on characterizing grasp quality metrics, highlighting their significant role in the analytical approach to grasp planning.

**Techniques and Algorithms**

Grasp planning techniques and algorithms have significantly evolved, with several prominent methods shaping modern robotic grasping. Analytical approaches rely on mathematical models to evaluate potential grasps by considering the geometry and physical properties of objects. These models allow for precise predictions of grasp stability and performance, often using principles from physics and geometry to calculate the forces and torques involved (Ferrari & Canny, 1992). Sampling-based methods focus on generating many potential grasps, which are then evaluated against predefined criteria such as stability or task suitability. These methods are advantageous when dealing with complex objects, as they provide flexibility in grasp selection through exploration of diverse grasp configurations (Bohg et al., 2014).

In contrast, machine learning approaches have gained prominence by utilizing large datasets to train models capable of predicting successful grasps based on sensory data, including visual and tactile inputs. These approaches enable robots to generalize from past experiences and adapt to new objects and environments with higher accuracy and efficiency. Machine learning techniques such as deep learning have been particularly effective in improving grasp prediction by learning complex relationships between object features and grasp success (Mahler et al., 2017). These advancements have greatly expanded the capabilities of robotic systems in handling diverse tasks and environments.

**Recent Advances**

Recent research in grasp planning has concentrated on improving the robustness and adaptability of robotic systems. One of the key advancements is the use of deep learning models, which leverage neural networks to learn complex grasping strategies from extensive datasets. These models have significantly enhanced a robot's ability to generalize across various objects and environments, making grasping more reliable and versatile in real-world applications (Mahler et al., 2019). By training on a diverse range of object geometries and grasp configurations, deep learning approaches enable robots to handle previously unseen objects with greater precision and efficiency.

Another significant advancement is the integration of multiple sensors, including visual and tactile sensors, to provide a comprehensive understanding of both the object and its surrounding environment. This multi-modal sensor integration enhances a robot's perception, allowing for more informed decision-making during grasp planning (Calandra et al., 2018). Additionally, real-time grasp planning algorithms have been developed, enabling robots to dynamically adapt to changes in the environment and object positioning. These algorithms process sensory data rapidly, allowing robots to adjust their grasps in real-time, which is crucial for tasks in unpredictable or constantly changing environments (Kopicki et al., 2016).

**Challenges and Future Directions**

Despite significant advancements, grasp planning continues to face several critical challenges. One of the foremost issues is object diversity, as robots must handle a wide range of objects varying in shape, size, and material properties. This variability makes it difficult to create a universal grasping solution that works effectively across all object types (Bohg et al., 2014). Another challenge lies in dynamic environments, where robots must adapt their grasps in real-time to account for changes in object position or external disturbances. This adaptability is crucial for applications where the robot interacts with moving or unstable objects, such as in home or healthcare settings (Kopicki et al., 2016). Furthermore, ensuring computational efficiency is a persistent hurdle, as grasp planning algorithms must be fast enough to allow real-time operation, particularly in tasks requiring rapid decision-making and execution (Mahler et al., 2017).

Looking ahead, future research is likely to focus on the further integration of machine learning with traditional grasp planning methods. Machine learning enables robots to learn from experience, improving their ability to generalize across different objects and environments. This integration could lead to more adaptive and intelligent robotic systems capable of handling new situations with minimal human intervention (Calandra et al., 2018). Additionally, advancements in sensor technology and improvements in the computational efficiency of algorithms will be essential for enabling real-time, robust grasping in more complex and dynamic environments. These developments are crucial for advancing the field and broadening the application of robotic grasping in industries like logistics, healthcare, and personal assistance.

## 2.3. Overview of Motion Planning

Motion planning is a fundamental technology in robotics that involves breaking down complex motion tasks into a series of discrete actions that can be executed by a robot (Fan, 2023). It plays a crucial role in applications such as mobile robots, autonomous driving, automated vehicles, and free-floating space robots (Liniger & Gool, 2020; Zhang & Zhu, 2020; Manzinger et al., 2021). The development of motion planning algorithms has led to the creation of efficient methods like Rapidly-exploring Random Trees (RRT), optimal RRT\*, and potentially guided-RRT\* (P-RRT\*) (Qureshi et al., 2019), which enhance the safety and autonomy of robots in dynamic environments. Moreover, sampling-based approaches, such as those explored by Dalibard and Laumond (2009), generate collision-free configurations randomly in free space, allowing for the construction of roadmaps that address motion planning problems efficiently. These methods are not only vital in traditional robotic applications but also extend to humanoid robots operating in complex environments (Li & Huang, 2007).

**Key Concepts and Definitions**

Key concepts in motion planning form the foundation of robotic pathfinding and navigation, focusing on ensuring that robots can achieve specified goals while avoiding obstacles.

* **Configuration Space (C-Space):**  
  Configuration Space, or C-Space, is a mathematical abstraction used in motion planning that represents all possible configurations or states a robot can assume. Each point within this space corresponds to a unique position and orientation of the robot. The concept simplifies the complex task of motion planning by translating physical and operational constraints into geometric constraints, making it easier for algorithms to evaluate feasible paths. C-Space is particularly useful when dealing with robots that have multiple degrees of freedom, such as robotic arms, where high-dimensional spaces must be navigated (LaValle, 2006). This abstraction enables planners to focus on the robot's movement through space rather than the intricate details of its physical structure, making it a fundamental tool in path and trajectory planning.
* **Path Planning:**  
  Path planning is the process of determining a collision-free route for a robot to move from a start configuration to a goal configuration within C-Space. The objective is to find a safe and efficient path that avoids obstacles while optimizing the robot’s movements. Popular algorithms, such as Rapidly-exploring Random Trees (RRT) and A\*, are commonly used to explore this space. These methods generate candidate paths and evaluate them for feasibility and safety. The challenge often lies in navigating environments with complex and dynamic obstacles, which requires algorithms to be both efficient and adaptable (Karaman & Frazzoli, 2011). Path planning forms the backbone of robotic navigation and is essential in applications ranging from autonomous vehicles to mobile robots.
* **Trajectory Planning:**  
  Trajectory planning builds upon path planning by adding considerations for the timing and dynamics of the robot’s movements. While path planning focuses on spatial configurations, trajectory planning ensures that the robot’s movement is smooth and physically feasible, factoring in velocity, acceleration, and control inputs. This is critical in real-world applications where sudden movements or excessive acceleration can lead to instability or damage. Techniques such as time-parameterized planning are used to calculate the optimal control inputs required for the robot to follow a planned path while adhering to dynamic constraints (Choset et al., 2005). In fields such as industrial robotics and autonomous driving, trajectory planning ensures that robots not only move safely but also operate efficiently.

#### **Major Theories and Models**

**Graph-Based Methods:**

Graph-based methods include techniques such as grid-based approaches and roadmaps. In these methods, the environment is represented as a graph where nodes correspond to specific configurations, and edges represent feasible transitions between these configurations. Algorithms such as Probabilistic Roadmaps (PRM) and Rapidly-exploring Random Trees (RRT) are widely used in this context. PRM builds a roadmap by randomly sampling the configuration space and connecting nodes through valid paths, while RRT incrementally explores the space by growing a tree from the start configuration toward the goal. These methods leverage graph search algorithms, like Dijkstra's or A\*, to find optimal or near-optimal paths through the graph (LaValle, 2006). Graph-based methods are especially effective in static environments but may require adaptation for dynamic scenarios.

**Optimization-Based Methods:**

Optimization-based methods formulate motion planning as an optimization problem, where the goal is to minimize a cost function that can represent factors such as distance, time, energy, or smoothness of the path. These methods typically rely on continuous optimization techniques to refine the path or trajectory. One common approach is trajectory optimization, where the objective is to find a feasible trajectory that minimizes a specific cost while respecting the robot’s physical constraints (Zucker et al., 2013). These methods are particularly useful in environments where multiple criteria must be balanced, such as minimizing both energy consumption and execution time. They offer precision and flexibility, making them highly effective in complex robotic tasks.

**Sampling-Based Methods:**

Sampling-based methods generate random samples in the configuration space to construct feasible paths, making them highly effective in high-dimensional spaces where deterministic methods might struggle. These methods, including PRM and RRT, are known for their ability to handle complex environments and obstacles. By randomly sampling points in C-Space and connecting them through collision-free edges, sampling-based methods efficiently explore large search spaces. Although these approaches do not guarantee optimality, extensions like RRT\* and PRM\* have been developed to seek optimal solutions by refining the sampling process (Karaman & Frazzoli, 2011). Sampling-based methods are particularly valuable in robotic applications requiring fast, scalable, and adaptable planning in environments with high-dimensionality or uncertainty.

**Techniques and Algorithms**

**Probabilistic Roadmaps (PRM):**

Probabilistic Roadmaps (PRM) are a widely-used technique in motion planning that constructs a roadmap by randomly sampling points within the configuration space. These sampled points, or nodes, are then connected by edges representing feasible, collision-free paths. The roadmap is essentially a graph where pathfinding can be performed using search algorithms like Dijkstra's or A\*. PRM is particularly effective in multi-query scenarios, where the roadmap can be reused for different start and goal configurations after the initial construction (Kavraki et al., 1996). PRM is advantageous in high-dimensional spaces due to its ability to handle complex environments, but it may struggle in dynamic or narrow passage environments where sampling alone may not capture the necessary configurations.

**Rapidly-exploring Random Trees (RRT):**

Rapidly-exploring Random Trees (RRT) is another popular sampling-based technique used to explore large configuration spaces efficiently. RRT incrementally builds a tree by starting from an initial configuration and extending branches toward randomly sampled points within the space. This method is particularly well-suited for problems where the configuration space is large or cluttered, as it quickly explores uncharted regions. RRT tends to generate fast, feasible paths, but these paths are not always optimal. To address this limitation, extensions such as RRT\* have been developed, which aim to optimize the path by minimizing the cost during the tree expansion (LaValle, 2006; Karaman & Frazzoli, 2011). RRT's strength lies in its ability to rapidly explore and handle real-time planning tasks.

**A Algorithm: \***

The A\* algorithm is a classic graph search algorithm that is frequently used in motion planning to find the shortest path between two points. A\* operates by combining two key factors: the cost to reach a particular node and the estimated cost to reach the goal from that node (the heuristic). This approach ensures that A\* not only finds the shortest path but also does so efficiently by prioritizing nodes that seem most promising based on the heuristic function (Hart et al., 1968). A\* is well-suited for structured environments where accurate cost functions can be defined, making it ideal for grid-based path planning. However, its performance can degrade in high-dimensional or highly dynamic spaces, where the number of nodes grows exponentially.

**Recent Advances**

**Multi-Robot Motion Planning:**

Recent advancements in motion planning have increasingly focused on multi-robot systems, which address the challenge of coordinating multiple robots to perform tasks cooperatively without collisions. This includes ensuring that robots can avoid each other and work together to complete tasks more efficiently. Methods like centralized planning use a global controller to manage the paths of all robots, while decentralized approaches allow each robot to plan independently while sharing information with other robots (Yu & LaValle, 2016). Multi-robot motion planning is particularly important in environments like automated warehouses and aerial drone coordination, where multiple robots must operate in close proximity (van den Berg et al., 2009).

**Real-Time Motion Planning:**

Real-time motion planning has become increasingly critical in dynamic and unpredictable environments such as autonomous driving and surgical robotics. Algorithms such as Model Predictive Control (MPC) allow robots to adjust their trajectories in real-time by predicting future states based on current sensor data, ensuring quick and safe responses to changes (Falcone et al., 2007). Advances in sensor integration, combined with real-time path optimization, enable robots to react to obstacles and environmental shifts without needing to replan the entire path from scratch (Ziegler et al., 2014). These developments significantly enhance the robot's ability to operate safely and efficiently in fast-changing environments.

**Machine Learning Integration:**

The integration of machine learning techniques into motion planning has provided a new dimension of adaptability and efficiency. By utilizing large datasets, robots can learn from past experiences and improve their pathfinding strategies. Methods like deep reinforcement learning allow robots to autonomously optimize their paths through trial and error, gradually improving their ability to navigate complex environments (Tai et al., 2017). Moreover, machine learning helps in predicting obstacles, understanding terrain, and optimizing paths based on both visual and sensory input (Kahn et al., 2018). This combination of machine learning and motion planning allows robots to adapt more intelligently and autonomously to new environments.

**Challenges and Future Directions**

**High-Dimensional Spaces:**

One of the most significant challenges in motion planning is navigating high-dimensional spaces, particularly for robots with many degrees of freedom, such as humanoid robots or robotic arms. As the number of joints and movable parts increases, the size and complexity of the configuration space grow exponentially, making it increasingly difficult to find feasible paths (Kavraki et al., 1996). Traditional algorithms struggle to efficiently explore these spaces due to the vast number of possible configurations. Researchers have been developing more advanced sampling-based and optimization-based methods to tackle this problem, but high-dimensional planning remains computationally intensive and difficult to solve in real-time.

**Dynamic and Uncertain Environments:**

Dynamic and uncertain environments present another ongoing challenge. In real-world applications, environments often change over time, introducing moving obstacles, shifting goals, or other unpredictable factors. Robots must be capable of adapting to these changes while maintaining safe and robust navigation. Traditional motion planning algorithms, which assume static environments, are often not equipped to handle dynamic elements effectively. Techniques such as dynamic replanning and predictive control have been developed to allow robots to react to environmental changes, but achieving true robustness in uncertain environments remains an active area of research (Ziegler et al., 2014).

**Computational Demands:**

The computational demands of motion planning algorithms, especially when combined with the need for real-time decision-making, are another significant challenge. Algorithms must strike a balance between efficiency and accuracy while operating within limited time constraints. Real-time motion planning for autonomous vehicles or robots in cluttered environments, for example, requires not only fast pathfinding but also the ability to dynamically update paths as new obstacles or hazards appear (Kohlbrecher et al., 2011). Optimizing the trade-off between speed and solution quality, especially in high-dimensional or dynamic environments, is a critical focus for future research.

**Future Directions:**

Moving forward, motion planning research is likely to focus heavily on the integration of artificial intelligence (AI) and machine learning. These technologies will enable robots to learn from experience, predict obstacles, and optimize their pathfinding strategies based on prior data. Reinforcement learning and deep learning are expected to play a key role in making planning algorithms more intelligent and adaptable (Kahn et al., 2018). Additionally, advancements in sensor technology and increased computational power will help mitigate current challenges, providing robots with better situational awareness and faster processing capabilities, further enhancing the efficiency and robustness of motion planning systems.

## 2.3. Integrated Grasp and Motion Planning

Integrated grasp and motion planning is a crucial aspect of robotics that involves simultaneously determining how a robot should grasp an object and how it should move both the robot and the object to achieve a desired outcome. This integrated approach is essential for enhancing the efficiency and effectiveness of robotic operations, particularly in complex and dynamic environments (Dang-Vu et al., 2015; Rosell et al., 2019). By combining grasp and motion planning, robots can perform tasks more seamlessly, reducing the time and computational resources required for separate planning stages (Ichnowski et al., 2020; Vahrenkamp et al., 2010).

Integrated Task and Motion Planning (TAMP) addresses the challenge of planning for robots in environments with numerous objects, requiring actions not only for self-movement but also for object manipulation (Garrett et al., 2021). This field extends traditional task and motion planning concepts to handle complex scenarios effectively. Additionally, deep learning-based approaches have been explored for motion planning in self-driving vehicles, covering behavior planning, trajectory planning, and End-to-End Learning (E2EL) (Ganesan, 2024).

The integration of grasp and motion planning algorithms allows for the generation of collision-free trajectories to grasps or grasp sets that are precomputed or synthesized during the planning process (Ichnowski et al., 2020). This holistic approach enables robots to decide on the best grasp for an object and plan a collision-free path that facilitates the successful execution of the task (Dang-Vu et al., 2015). Additionally, the synthesis of grasps and motions involves generating potential grasps for an object and planning motions using efficient planners that guide the motion planning process within a reduced search space, resulting in paths with human-like appearances (Rosell et al., 2019).

Moreover, the exploration of deep learning techniques has shown promise in accelerating grasp-optimized motion planning, demonstrating the potential for advanced technologies to further enhance integrated grasp and motion planning processes (Ichnowski et al., 2020). For example, the proposed Grasp-RRT planner combines tasks necessary for grasping an object, such as finding a feasible grasp, solving inverse kinematics, and searching for a collision-free trajectory to reach the grasping pose (Vahrenkamp et al., 2010).

#### Challenges in Integration

The integration of grasp and motion planning presents several challenges:

* **Complexity of Combined Planning**: Coordinating both grasp and motion planning increases the complexity of the problem, requiring advanced algorithms that can handle high-dimensional spaces and dynamic constraints.
* **Real-Time Constraints**: Achieving real-time performance is crucial for practical applications, necessitating efficient algorithms that can quickly generate feasible and optimal plans.
* **Sensor and Data Integration**: Effective integration requires robust sensor data processing to accurately perceive the environment and adapt plans accordingly.

#### Existing Integrated Approaches

Several approaches have been developed to address the challenges of integrated grasp and motion planning:

* **Sequential Planning**: Involves planning the grasp first, followed by motion planning. While simpler to implement, this approach can lead to suboptimal solutions as the grasp is not optimized for the subsequent motion.
* **Simultaneous Planning**: Simultaneously plans both grasp and motion, considering the interdependencies between the two. This approach can generate more optimal solutions but is computationally more demanding.
* **Hierarchical Planning**: Decomposes the problem into hierarchical levels, planning high-level tasks first and refining them into detailed plans. This can balance between optimality and computational efficiency.

#### Techniques and Algorithms

Several techniques and algorithms have been proposed for integrated grasp and motion planning:

* **Optimization-Based Methods**: Formulate the problem as a single optimization task, minimizing a cost function that includes both grasp and motion components.
* **Sampling-Based Methods**: Extend traditional sampling-based motion planning algorithms, like RRT and PRM, to consider grasp planning simultaneously.
* **Machine Learning Approaches**: Use machine learning models to predict feasible grasps and motions based on prior experience, improving planning speed and adaptability.

#### Recent Advances

Recent research in integrated grasp and motion planning has focused on improving the efficiency and robustness of planning algorithms:

* **Deep Reinforcement Learning**: Applying deep reinforcement learning to learn integrated planning strategies from large datasets, enabling robots to handle more complex tasks with greater autonomy.
* **Real-Time Adaptive Planning**: Developing algorithms that can adapt plans in real-time based on changes in the environment and object dynamics, enhancing the robot's ability to operate in unpredictable settings.
* **Sensor Fusion Techniques**: Combining data from multiple sensors, such as cameras, LIDAR, and tactile sensors, to provide a more comprehensive understanding of the environment and improve planning accuracy.

#### Applications and Implications

Integrated grasp and motion planning has numerous applications across various domains:

* **Industrial Automation**: Enhancing the efficiency and flexibility of robotic systems in manufacturing and assembly lines.
* **Service Robotics**: Improving the capability of robots to perform household tasks, assistive functions, and other service-oriented activities.
* **Medical Robotics**: Enabling precise and adaptable robotic systems for surgical procedures and rehabilitation.

#### Challenges and Future Directions

Despite significant progress, integrated grasp and motion planning continues to face several challenges:

* **Scalability**: Ensuring that planning algorithms can scale to handle more complex tasks and environments.
* **Robustness**: Developing methods that can handle uncertainties and variations in the environment and object properties.
* **User Interaction**: Creating intuitive interfaces and control mechanisms that allow users to easily specify and adjust tasks for robotic systems.

### 2.4. Existing Methods and Approaches

In this section, we review the existing methods and approaches employed in integrated grasp and motion planning. The integration of these two domains is critical for achieving efficient and seamless robotic manipulation, allowing robots to simultaneously plan how to grasp an object and how to move it in space without collisions. A comprehensive understanding of both grasp and motion planning techniques is necessary to explore how they can be effectively integrated.

#### 1. Sequential Approaches

Sequential approaches to integrated grasp and motion planning typically involve two distinct stages: first, a stable grasp is planned, and then a collision-free motion is computed for the robot arm to reach the grasped object. These methods are easier to implement since they decouple the problem, but they can lead to suboptimal solutions because the motion is not considered during the grasp planning phase. Examples include the work of Wan and Harada (2017), who developed regrasp planning techniques based on precomputed grasps.

**Limitations:**

* Potential suboptimality due to the lack of coordination between grasp and motion planning.
* Time inefficiency because the motion may need to be replanned if the initial grasp is not feasible in the context of the robot's movements.

#### 2. Simultaneous Approaches

Simultaneous approaches attempt to overcome the limitations of sequential methods by planning grasp and motion simultaneously. These approaches aim to consider the interdependencies between the grasp configuration and the subsequent motion trajectory. For example, Vahrenkamp et al. (2010) proposed an integrated grasp and motion planning approach using Rapidly-exploring Random Trees (RRT), where the grasp and the motion trajectory are planned together in a unified framework.

**Advantages:**

* Better coordination between grasp and motion, leading to more optimal solutions.
* Reduced need for re-planning, which increases overall efficiency.

**Challenges:**

* Computational complexity is higher due to the expanded search space.
* The need for real-time planning increases the difficulty of implementation in dynamic environments.

#### 3. Optimization-Based Methods

Optimization-based methods treat the problem of integrated grasp and motion planning as a single optimization problem. These methods minimize a cost function that takes into account factors such as grasp stability, path length, energy consumption, and collision avoidance. Examples include the work of Dang-Vu et al. (2016), who used optimization techniques to balance grasp and motion objectives.

**Advantages:**

* Allows for a more holistic solution that balances multiple factors such as grasp stability, energy efficiency, and safety.
* Provides a flexible framework for incorporating additional constraints, such as dynamic obstacles or varying object properties.

**Challenges:**

* Computationally expensive due to the need to solve complex optimization problems.
* Requires careful tuning of the cost function to achieve desired outcomes in real-world scenarios.

#### 4. Machine Learning-Based Approaches

Recent advances in machine learning, particularly deep learning, have shown promise in accelerating the integrated planning process. These approaches use large datasets to train models that predict both grasp points and motion trajectories. Ichnowski et al. (2020) demonstrated how deep reinforcement learning could be applied to grasp-optimized motion planning, where the robot learns to generate grasp and motion plans based on previous experience.

**Advantages:**

* Significantly reduces the computational time needed for planning by using pre-learned models.
* Can adapt to new environments and objects more efficiently by leveraging prior knowledge.

**Challenges:**

* Requires large datasets for training, which can be difficult to obtain for complex manipulation tasks.
* Generalization to novel objects or scenarios may be limited depending on the quality of the training data.

#### 5. Hierarchical Approaches

Hierarchical methods decompose the problem into different levels, such as high-level task planning followed by low-level motion and grasp planning. This approach allows for a more structured solution to integrated planning and helps manage the complexity of the combined problem. For example, Bertoni et al. (2021) proposed a hierarchical grasp planning pipeline that provides high-level task planning followed by detailed grasp and motion planning stages.

**Advantages:**

* Reduces the complexity of the problem by breaking it down into manageable sub-problems.
* Offers a balance between solution quality and computational efficiency.

**Challenges:**

* Hierarchical decomposition can lead to suboptimal solutions if the high-level decisions do not adequately account for low-level constraints.

### Summary of Existing Methods

The reviewed methods provide a variety of strategies for integrating grasp and motion planning in robotics. Sequential approaches offer simplicity but may result in suboptimal solutions, while simultaneous and optimization-based methods provide more optimal solutions at the cost of increased computational complexity. Machine learning and hierarchical approaches present innovative ways to address the challenges of integrated planning, but they come with their own set of limitations. Future research is likely to focus on improving computational efficiency, scalability, and real-time adaptability of these methods, particularly in dynamic and unpredictable environments.

### Gaps in the Literature

Despite the significant advancements in integrated grasp and motion planning, there are several gaps in the existing literature:

* **Scalability**: Most methods struggle to scale efficiently to high-dimensional tasks or environments with numerous dynamic obstacles.
* **Real-time adaptability**: Current algorithms are often computationally expensive and may not be suitable for real-time applications.
* **Generalization to novel tasks**: Many approaches rely on precomputed data or assumptions about the environment, limiting their applicability in scenarios involving unfamiliar objects or dynamic changes.

Addressing these gaps will be crucial for advancing the field of robotics, particularly in applications requiring real-time performance and adaptability.

## Summary

# Chapter 3: Problem Description

#### 3.1 Overview of Integrated Grasp and Motion Planning

Robotic systems face significant challenges when tasked with grasping and manipulating objects in dynamic environments. Traditionally, grasp planning and motion planning have been handled separately, which can lead to inefficiencies and suboptimal performance. In many scenarios, the interaction between the grasp and the robot’s motion path is critical for achieving smooth and efficient task execution. Integrated grasp and motion planning approaches seek to resolve this by simultaneously planning how the robot grasps an object and how it moves while avoiding obstacles and maintaining task feasibility.

This research addresses a specific challenge in the domain of integrated grasp and motion planning. Specifically, it focuses on evaluating different path planning algorithms to determine which methods can best optimize both the robot’s grasp and motion, particularly in constrained or complex environments.

#### 3.2 Problem Statement

The core problem this research addresses is the need for efficient and reliable integrated grasp and motion planning algorithms for robotic manipulators. This is especially challenging when operating in environments with varying object configurations and dynamic constraints. The primary question driving this research is how different algorithms, particularly RRT\*, perform in solving these planning tasks, compared to other established algorithms like JPlusRRT, IKRRT, and BIKRRT.

In this context, the problem can be broken down into the following components:

1. **Grasp Selection**: The robot must determine a stable and feasible grasp configuration for each object in the workspace. This involves analyzing possible grasp poses and selecting one that maximizes stability while being feasible in terms of the robot's kinematics.
2. **Motion Planning**: Once a grasp is selected, the robot must plan a path from its initial configuration to the object, ensuring that it avoids obstacles and operates within its physical limitations.
3. **Dynamic Environments**: In real-world applications, the environment may change dynamically, making it necessary for the algorithms to adapt to these changes while maintaining efficiency and success in task completion.

The objective of this research is to evaluate how well the RRT\*, JPlusRRT, IKRRT, and BIKRRT algorithms handle integrated grasp and motion planning, particularly in complex environments with diverse object placements and obstacles.

#### 3.3 Research Gap

While several algorithms for motion planning exist, such as Rapidly-exploring Random Trees (RRT) and its optimal variant RRT\*, there is a gap in understanding their specific performance in integrated planning tasks. Although some studies have implemented grasp planning in static or simple environments, fewer have addressed the complexity of environments where both grasp and motion planning need to be tightly integrated. Additionally, while the benchmark scenarios from Rudorfer et al. provide valuable insights into the performance of algorithms like JPlusRRT and IK-RRT, these scenarios have not been extensively compared with RRT\* in terms of integrated planning.

#### 3.4 Specific Challenges

The specific challenges addressed by this dissertation are as follows:

1. **High-dimensional Search Space**: The configuration space of robotic manipulators, especially those with six or more degrees of freedom, is vast and difficult to search efficiently. The challenge is to evaluate how RRT\* and other algorithms manage this space while considering both grasping and motion.
2. **Dynamic and Complex Environments**: Real-world environments are often cluttered and dynamic, which makes the planning process even more difficult. This research tests the robustness of the algorithms in these environments, particularly those with confined spaces and irregular object arrangements.
3. **Algorithm Performance**: The goal is to evaluate the trade-offs between computational efficiency (planning time), success rates (finding a feasible path), and the quality of the paths generated by each algorithm.
4. **Practical Implementation**: While many algorithms have theoretical benefits, practical implementation often reveals constraints or bottlenecks that were not anticipated. By implementing these algorithms in the PyBullet environment and the Jogramop framework, this research aims to identify these practical challenges.

#### 3.5 Importance of the Problem

The success of integrated grasp and motion planning is crucial in advancing the capabilities of robotic systems. Robots are increasingly being deployed in industries such as manufacturing, healthcare, and service sectors, where they need to handle complex tasks with high reliability. The findings of this research will contribute to improving the performance of robotic systems, making them more adaptable and efficient in real-world applications.

#### 3.6 Summary

This chapter outlines the core problem that this dissertation addresses: the challenge of efficient and robust integrated grasp and motion planning in dynamic environments. The goal is to evaluate several algorithms, with a particular focus on RRT\*, to determine their strengths and limitations in handling complex robotic manipulation tasks. The following chapters will detail the methodology and experimental setup used to tackle this problem and present the findings from the evaluations of the different algorithms.

# Chapter 4: Methods and Theories

## 4.1. Introduction

The methodology of this research focuses on the experimental comparison of several path-planning algorithms, including RRT\*, JPlusRRT, and IK-RRT, within predefined benchmark environments. The aim is to evaluate their performance in robotic grasp and motion planning tasks, particularly emphasizing the RRT\* algorithm. Simulations were conducted using tools such as PyBullet and the Jogramop framework. These tools facilitated testing across 20 different scenarios, with the performance measured through key metrics like planning time and success rate. This chapter elaborates on the research design, the algorithms, the experimental setup, and the challenges faced during implementation and testing.

## 4.2. Problem Formulation

The problem addressed in this research involves the integrated grasp and motion planning for a robotic manipulator, with a focus on comparing algorithmic performance in benchmark scenarios. The goal is to evaluate how different algorithms manage both grasp selection and motion planning under various conditions.

In this context, we consider a set of potential grasps, where each grasp consists of a grasp pose representing the 6-DoF position and orientation of the gripper relative to the object.

The motion planning component involves searching the robot's configuration space , where each configuration represents a particular set of joint angles for the manipulator. The collision-free subset of the configuration space is denoted as . The objective for the motion planner is to find a path from an initial configuration to a goal configuration , ensuring that the chosen grasp ​ is reached while minimizing path length, avoiding obstacles, and adhering to the robot's kinematic constraints (Rudorfer et al., 2024).

To compare algorithms, we set up benchmark scenarios involving a variety of objects, grasps, and environmental configurations. The goal is to determine how well each algorithm performs in terms of path planning efficiency, grasp stability, and adaptability to dynamic changes in the environment. Performance metrics such as planning time, path length, and success rate will be used to evaluate the effectiveness of each algorithm in real-time simulations.

## 4.3. Research Design

This research adopts an experimental, comparative approach, designed to evaluate the effectiveness and efficiency of path-planning algorithms in dynamic environments. The main focus is the comparison between RRT\*, JPlusRRT, and IK-RRT algorithms, executed within two distinct environments. This design was chosen for its ability to assess the performance of each algorithm under varying conditions, aligning with the primary objective of comparing planning time and success rate across complex scenarios.

The experimental approach aligns with the research objectives by providing a structured method to systematically test and evaluate the algorithms across a set of 20 predefined benchmark scenarios. Each algorithm’s ability to handle grasp and motion tasks under different conditions was measured and compared to baseline results.

## 4.4. Algorithms and Frameworks

The following algorithms were selected and evaluated for this study:

* **RRT**\* (Rapidly-exploring Random Tree Star)
* **JPlusRRT**
* **IK-RRT**

The algorithms were initially implemented and tested in a simple PyBullet environment with a 6-DOF Franka Panda robot, and later RRT\* was implemented within the Jogramop framework. The decision to focus on RRT\* was based on its asymptotic optimality and probabilistic completeness, which provide significant advantages in grasp and motion planning tasks. RRT\* was selected for comparison with JPlusRRT and IK-RRT due to its potential to offer smoother and more direct paths in complex environments.

### 4.4.1. Simulation Environment

Two simulation environments were utilized for testing:

1. **PyBullet Environment**: This environment was custom-built using a Franka Panda robot and a variety of obstacles to simulate basic grasp and motion tasks. All algorithms (RRT\*, JPlusRRT, IKRRT, BIKRRT) were tested here.
2. **Jogramop Framework**: This framework was adopted in the second phase of the study. It features 20 predefined benchmark scenarios, each designed to test robotic grasp and motion planning in more complex and confined environments.

Both environments were crucial to the experimental process, with PyBullet serving as the initial testing ground, while the Jogramop framework provided a more complex and dynamic platform for further comparisons.

### 4.4.2. RRT\* Algorithm

RRT\* is a sampling-based algorithm designed to incrementally build a tree in the configuration space, finding collision-free paths while optimizing the path towards the goal configuration. The following key features make RRT\* suitable for this research:

* **Cost Minimization**: Each vertex in the tree stores the cost from the start, allowing the algorithm to find the lowest-cost path to the goal.
* **Rewiring**: The algorithm rewires nodes to ensure that all paths are optimized for minimal distance, resulting in smoother, shorter trajectories.

The RRT\* algorithm was implemented using its standard pseudo-code and modified for the specific task of robotic grasp planning, focusing on both grasp and motion optimization.

## 4.5. Experimental Setup

The experiments were structured by running the RRT\* algorithm 100 times in each of the 20 benchmark scenarios from the Jogramop framework. The scenarios ranged from simple environments with few obstacles to complex, confined spaces requiring precise navigation. The algorithms were tested under varying conditions, including changes in obstacle density, goal direction probability, and object placement.

The goal was to compare RRT\* with the baseline results from JPlusRRT and IK-RRT, which were already integrated into the Jogramop framework. The performance of RRT\* was measured using the following metrics:

* **Planning Time**: The total time taken to compute a valid path.
* **Success Rate**: The percentage of trials where the robot successfully reached the goal without collisions.

## 4.6. Metrics for Evaluation

The performance of the algorithms was evaluated using two key metrics:

* **Planning Time**: This reflects the algorithm's computational efficiency. A shorter planning time indicates faster computation of a collision-free path.
* **Success Rate**: This measures the reliability of the algorithm in successfully finding a path to the goal. A high success rate indicates robustness in handling dynamic and obstacle-rich environments.

These metrics were chosen because they directly reflect the real-world performance of grasp and motion planning algorithms. Planning time assesses the computational cost, while the success rate measures the practical applicability of the algorithm in robotic tasks.

## 4.7. Data Collection

Data was collected for each scenario by logging the planning time and success rate after every trial. The results were stored using automated logging scripts within the simulation framework, ensuring accurate tracking of each run. Each trial was repeated 100 times to minimize the effect of outliers, and any anomalous results were carefully reviewed and validated for consistency.

## 4.8. Testing and Validation

To validate the performance of RRT\*, its results were compared against JPlusRRT and IK-RRT within the same scenarios. Validation was ensured by:

* **Repeated Trials**: Each scenario was tested 100 times, and the average results were calculated to ensure consistency.
* **Cross-Scenario Testing**: Algorithms were tested across various benchmark scenarios, which helped ensure that the results were robust and applicable to different environments.

While conducting these tests, several challenges were encountered, such as simulation crashes and algorithmic adjustments required for complex environments.

## 4.9. Challenges and Adaptations

During the research, several challenges arose, particularly in setting up the simulation environments and implementing the algorithms. Designing environments that closely mirrored real-world tasks was time-consuming, and implementing all algorithms within the Jogramop framework proved more complex than initially expected.

Initially, the plan was to implement all four algorithms in the Jogramop framework. However, due to time constraints, only RRT\* was fully implemented and tested. This adaptation allowed for a focused comparison of RRT\* against JPlusRRT and IK-RRT, which were already integrated into the framework, ensuring meaningful results were obtained despite the time limitations.

# Chapter 5: Project Management

Effective project management was crucial to the successful implementation of my dissertation, which involved coding, testing, and comparing multiple path-planning algorithms in simulated environments. The project was managed by setting clear milestones, tracking progress, prioritizing tasks, and adapting to challenges that arose during the process.

A screenshot of a graph

Description automatically generated

Figure 1 Gantt Chart for Dissertation on Integrated Grasp and Motion Planning.

#### Timeline and Milestones

I began by defining specific milestones to guide the progress of the project. Initially, I implemented a simple RRT algorithm in a PyBullet simulation environment using a 6-DOF robot, the Franka Panda robot. As I gained more familiarity with the environment, I expanded to implementing advanced algorithms such as RRT\*, JPlusRRT, IKRRT, and BIKRRT. These milestones helped to keep the project structured, with a clear focus on building and testing the algorithms sequentially.

Later, I set up the jogramop\_framework environment developed by Rudorfer et al., which included 20 benchmark scenarios. While the original goal was to implement multiple algorithms in this environment, due to time constraints, only RRT\* was fully implemented and compared with the benchmark results. Each phase of the project was organized and executed according to a timeline, and the overall progress was tracked to ensure timely completion of key tasks.

#### Task Management

The project was divided into three key areas: coding, testing, and evaluation of algorithms. The first priority was to implement the RRT\* algorithm and its variations. Once the core algorithms were in place, I focused on testing their performance in different environments, specifically using the benchmark scenarios from the jogramop framework.

Given the complexity of the task, coding and algorithm implementation took the majority of my time. Testing and evaluation followed naturally, with each iteration requiring detailed analysis and adjustments to ensure the algorithm's accuracy and performance. Task management followed an agile methodology, where iterative development cycles allowed for frequent testing and feedback.

#### Resource Management

I relied on PyBullet for simulations and used the jogramop framework for comparing RRT\* with other algorithms in benchmark scenarios. GitHub was utilized for version control, while Trello was employed for project management and task tracking. Weekly meetings with my supervisor provided valuable guidance, helping me to resolve issues related to environment setup, algorithm implementation, and scenario testing.

The main limitation was the lack of comprehensive online resources for working with specific environments and algorithms, which added complexity to the project. Despite these challenges, having the jogramop framework, which provided ready-made environments, significantly improved the process, allowing me to focus on algorithm implementation and performance testing.

#### Risk Management

There were several risks and challenges during the project, particularly with algorithm performance and simulation failures. Setting up the simulation environment and ensuring compatibility with each algorithm required significant effort. Some algorithms did not perform as expected in certain environments, leading to several iterations of debugging and refinement.

To mitigate these risks, I reduced the number of algorithms tested in the new scenarios and focused on the performance of RRT\* in the jogramop framework. This allowed me to maintain the quality and depth of the analysis while still comparing the algorithm against benchmark scenarios. My supervisor’s guidance was instrumental in navigating these challenges.

#### Time Management

The most time-intensive phase of the project was coding and implementing the algorithms, which required careful attention to detail and frequent adjustments. I had initially planned to implement multiple algorithms in the new scenarios, but time constraints only allowed for the completion of RRT\*. As a result, I focused on ensuring that the RRT\* algorithm was thoroughly tested and compared to the benchmark scenarios in terms of runtime and success rate.

Despite these delays, the project was managed effectively, with time allocated to each phase based on its complexity and importance. Weekly progress tracking helped to keep the project on schedule, although some adjustments were made in the final weeks to accommodate additional testing.

#### Collaboration and Supervision

I maintained regular communication with my supervisor, Dr. Martin Rudorfer, meeting with him weekly or bi-weekly to discuss progress and receive feedback. His guidance was invaluable, particularly in helping me navigate challenges related to algorithm implementation and the setup of the simulation environment. Based on his input, I made several adjustments to the project plan, including the decision to focus on RRT\* in the benchmark scenarios due to time constraints.

The collaborative feedback loop allowed me to refine my approach continuously and make data-driven decisions throughout the project. My supervisor's support ensured that I stayed on track and met the key milestones necessary for the successful completion of the dissertation.

#### Challenges and Adaptations

One of the biggest challenges I encountered was setting up the simulation environments and frameworks required for algorithm testing. Implementing the RRT\* algorithm and others in these environments was not a straightforward task due to the need for custom configurations and adjustments. Additionally, comparing the results with the benchmarks required detailed attention to each scenario's setup.

To overcome these challenges, I focused on the jogramop framework, which provided pre-existing environments for testing, reducing the time spent on setup and allowing me to concentrate on algorithm performance. However, I had to adjust my original plan by limiting the number of algorithms tested in new scenarios, which allowed me to complete the project within the given timeframe.

In conclusion, effective project management played a crucial role in the successful execution of this dissertation. By setting clear milestones, managing tasks effectively, and mitigating risks, I was able to complete the implementation and testing of the RRT\* algorithm and compare its performance against established benchmarks. Despite some challenges, regular supervision and adaptive time management ensured the project's success.

# Chapter 6: Evaluation

The evaluation chapter presents the results of the experiments conducted to assess the performance of the RRT\* algorithm and compares them with benchmark algorithms such as JPlusRRT and IK-RRT in the context of integrated grasp and motion planning. The evaluation focuses on key metrics, including planning time and success rate, and provides a detailed analysis of how well RRT\* performed in both simple and complex environments.

## 6.1 Experimental Results Overview

The experiments were conducted in two phases. In the first phase, RRT\*, JPlusRRT, IK-RRT, and BIKRRT were implemented and evaluated in a custom PyBullet environment. In the second phase, RRT\* was tested using the Jogramop framework, which provided 20 predefined benchmark scenarios. For each scenario, RRT\* was run 100 times, and its performance was compared against JPlusRRT and IK-RRT, which were already integrated into the Jogramop framework.

The evaluation metrics used were:

* **Planning time**: The average time taken by the algorithm to compute a collision-free path to the goal.
* **Success rate**: The percentage of trials in which the algorithm successfully found a feasible path to the goal within the given time limits.

## 6.2 Performance in Benchmark Scenarios

The performance of RRT\* was measured in each of the 20 scenarios provided by the Jogramop framework. The results were recorded in terms of planning time and success rate. RRT\* was compared against the benchmark results for JPlusRRT and IK-RRT from the Rudorfer et al. paper.

### 6.2.1 Planning Time

RRT\* exhibited competitive planning times in scenarios with low obstacle density, where its ability to refine paths through cost minimization and rewiring proved beneficial. In more complex environments, with higher obstacle density, the planning time for RRT\* increased due to the additional computation required to rewire the tree and ensure optimal paths.

For example, in scenarios 011 and 022, RRT\* had a similar planning time to JPlusRRT and IK-RRT, demonstrating efficiency in relatively simple environments. However, in scenarios like 044 and 045, which had dense obstacle fields, RRT\* showed longer planning times due to the need for frequent rewiring to avoid obstacles.

### 6.2.2 Success Rate

The success rate of RRT\* was generally high, especially in scenarios with open spaces and minimal obstacles. In environments with dense obstacles or confined spaces, the success rate of RRT\* decreased slightly compared to JPlusRRT and IK-RRT. This was primarily due to the challenges of rewiring and finding an optimal path in constrained environments.

For instance, in scenario 023, RRT\* achieved an 85% success rate, which was lower than the 98% success rate of IK-RRT. In contrast, in scenario 032, RRT\* achieved a 100% success rate, matching the performance of JPlusRRT and IK-RRT.

## 6.3 Comparative Analysis of Algorithms

The comparison between RRT\*, JPlusRRT, and IK-RRT highlighted several key insights:

* **RRT**\*: The strength of RRT\* lies in its ability to refine paths and ensure asymptotic optimality. This proved useful in less cluttered environments where the algorithm could take advantage of cost-based optimizations. However, in densely populated environments, RRT\* required more computational resources, resulting in longer planning times.
* **JPlusRRT**: JPlusRRT exhibited faster planning times in environments with complex obstacle configurations, primarily because it avoids the rewiring step that RRT\* performs. This made it more suitable for time-critical tasks, but its paths were not as optimal as those produced by RRT\*.
* **IK-RRT**: IK-RRT had a higher success rate in most scenarios, especially in environments where inverse kinematics played a significant role in reaching the goal. The precomputation of inverse kinematic solutions allowed it to quickly find feasible paths, albeit sometimes suboptimal compared to RRT\*.

## 6.4 Statistical Analysis

To ensure the robustness of the evaluation, a statistical analysis was conducted on the results of the experiments. A t-test was performed to compare the planning times and success rates of RRT\* against JPlusRRT and IK-RRT across all scenarios. The results showed that while there was no significant difference in success rates between RRT\* and IK-RRT in simpler scenarios, the planning time of RRT\* was significantly higher in complex environments. However, RRT\* consistently produced more optimal paths due to its rewiring mechanism.

## 6.5 Challenges Encountered

Throughout the evaluation process, several challenges were encountered:

* **Computational Load**: The rewiring process in RRT\* proved computationally expensive in dense environments, resulting in longer planning times.
* **Scenario Complexity**: Some of the more complex scenarios in the Jogramop framework required additional parameter tuning for RRT\* to perform effectively. This included adjusting the step size and radius for rewiring.
* **Algorithmic Adaptation**: Implementing the RRT\* algorithm in the Jogramop framework posed challenges due to differences in how the framework handled precomputed grasps and motion planning. These issues were mitigated by modifying the algorithm to better integrate with the framework's grasp planning capabilities.

## 6.6 Final Results

Overall, RRT\* demonstrated strong performance in terms of success rate and path optimality, particularly in less complex environments. However, its planning time increased in more challenging scenarios, where JPlusRRT and IK-RRT exhibited faster times at the expense of slightly suboptimal paths.

The key findings from the evaluation are as follows:

* **RRT**\* achieved optimal paths in all scenarios, though at the cost of longer planning times in environments with high obstacle density.
* **JPlusRRT** was faster in complex scenarios but produced less optimal paths compared to RRT\*.
* **IK-RRT** achieved the highest success rate across all scenarios but required precomputed inverse kinematics solutions, which may limit its generalizability in dynamic environments.

## 6.7 Conclusion of Evaluation

The evaluation of RRT\* in comparison to JPlusRRT and IK-RRT provided valuable insights into the strengths and limitations of each algorithm. While RRT\* excels in producing optimal paths, its computational cost makes it less suitable for time-critical tasks in dense environments. JPlusRRT and IK-RRT offer faster solutions but may sacrifice path optimality in certain scenarios.

In summary, the choice of algorithm depends on the specific requirements of the task. For tasks requiring optimal paths and where time is not a critical factor, RRT\* is the preferred choice. However, for tasks requiring quick responses, JPlusRRT or IK-RRT may be more suitable.

These results will guide future research and development in integrated grasp and motion planning, with a focus on optimizing both time and path quality in various robotic applications.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Scenario | |Gik|/\* |G| | J+-RRT | IK-RRT | RRT\* |
| 11 | 1/1 200 | 91%: 10.53/33.90 | 98%: 1.52/13.33 | |
| 12 | 3/3 200 | 95%: 6.08/26.14 | 100%: 0.02/0.01 | |
| 13 | 0/0 200 | 95%: 6.57/26.07 | â€” |  |
| 14 | 1/1 200 | 97%: 7.21/20.45 | 100%: 0.04/0.02 | |
| 15 | 0/0 200 | 80%: 23.17/24.67 | â€” |  |
| 21 | 1/1 200 | 100%: 2.63/2.94 | 100%: 0.05/0.03 | |
| 22 | 4/4 200 | 100%: 11.26/8.66 | 100%: 0.02/0.01 | |
| 23 | 2/2 200 | 81%: 35.13/21.67 | 98%: 1.52/13.33 | |
| 24 | 1/1 200 | 11%: 60.00/8.39 | 100%: 0.05/0.04 | |
| 25 | 1/1 200 | 2%: 118.56/9.39 | 11%: 108.82/32.07 | |
| 31 | 10/10 200 | 98%: 1.59/13.32 | 100%: 0.01/0.01 | |
| 32 | 11/11 200 | 100%: 0.15/0.29 | 100%: 0.01/0.01 | |
| 33 | 15/15 200 | 98%: 1.78/13.31 | 100%: 0.01/0.01 | |
| 34 | 15/15 200 | 100%: 0.70/0.98 | 100%: 0.01/0.01 | |
| 35 | 8/0 200 | 100%: 1.94/3.05 | â€” |  |
| 41 | 0/0 200 | 97%: 4.18/18.76 | â€” |  |
| 42 | 5/5 200 | 100%: 4.56/7.24 | 100%: 0.10/0.07 | |
| 43 | 2/2 200 | 90%: 25.19/19.80 | 100%: 0.42/0.32 | |
| 44 | 5/5 200 | 56%: 85.87/37.95 | 100%: 1.57/1.14 | |
| 45 | 4/4 200 | 75%: 604.38/206.88 | 100%: 13.51/8.29 | |

# Chapter 7: Conclusion

#### Summary of Research

This research focused on evaluating the RRT\* algorithm within the context of integrated grasp and motion planning, aiming to compare its performance against other algorithms like JPlusRRT and IK-RRT. The project started with implementing RRT in simple pybullet environments and progressed to incorporating the RRT\* algorithm. Additionally, the jogramop framework from Rudorfer et al. was employed to compare the performance of the RRT\* algorithm against other benchmarks in 20 different scenarios. Although time constraints limited the implementation of multiple algorithms in the new environment, the focus on RRT\* allowed for an in-depth assessment of its strengths and weaknesses.

#### Key Findings

The study found that RRT\* excels in less cluttered environments, showcasing its ability to refine paths iteratively for optimal results. Its asymptotic optimality makes it a strong candidate for tasks that demand high precision and efficient motion planning. However, in more complex and dense environments, the algorithm's computational demands rise due to the frequent rewiring required to optimize paths. Despite these challenges, RRT\* demonstrated robustness in dynamic and constrained environments, producing feasible and efficient paths in most scenarios.

#### Contributions to the Field

This dissertation contributes valuable insights into the application of RRT\* for integrated grasp and motion planning. By comparing RRT\* with JPlusRRT and IK-RRT in a benchmark environment, it provides a nuanced understanding of the relative strengths and weaknesses of each algorithm. The successful integration of RRT\* within the jogramop framework adds to the body of knowledge on motion planning in constrained spaces, a critical area for real-world robotic applications. The data collected offers a solid foundation for future studies looking to refine or adapt these algorithms for more complex tasks.

#### Practical Applications

The findings from this research hold practical implications across several industries. In manufacturing, robots can benefit from the efficient motion planning offered by RRT\*, improving both precision and operational safety. In healthcare, particularly for surgical robots or assistive devices, the ability to navigate dynamic environments with high accuracy is invaluable. Autonomous systems, such as self-driving cars or service robots, can also leverage the strengths of RRT\* for efficient navigation and task execution. Although this research was conducted in simulated environments, the insights gained are readily applicable to real-world challenges faced by robotic systems.

#### Final Thoughts

Overall, this research provided a comprehensive exploration of the RRT\* algorithm’s capabilities and limitations in integrated grasp and motion planning. The study overcame challenges, such as the complexity of setting up environments and limited time for testing all algorithms in new scenarios, to deliver meaningful results. Future research can build upon this work by expanding evaluations to other algorithms, exploring hybrid approaches, and conducting real-world testing. By addressing these areas, subsequent studies can further refine integrated grasp and motion planning strategies, making them even more efficient and practical for real-world robotic systems.

# References

Akinola, I. et al. (2021) “Dynamic Grasping with Reachability and Motion Awareness,” in IEEE International Conference on Intelligent Robots and Systems. Available at: https://doi.org/10.1109/IROS51168.2021.9636057.

Ali, A. and Lee, J.Y. (2020) “Integrated motion planning for assembly task with part manipulation using re-grasping,” Applied Sciences (Switzerland), 10(3). Available at: https://doi.org/10.3390/app10030749.

Bertoni, L. et al. (2021) “Towards a Generic Grasp Planning Pipeline using End-Effector Specific Primitive Grasping Actions,” in 2021 20th International Conference on Advanced Robotics, ICAR 2021. Available at: https://doi.org/10.1109/ICAR53236.2021.9659402.

Dalibard, S. and Laumond, J.P. (2010) “Control of probabilistic diffusion in motion planning,” in Springer Tracts in Advanced Robotics. Available at: https://doi.org/10.1007/978-3-642-00312-7\_29.

Dang-Vu, B.A., Porges, O. and Roa, M.A. (2016) “Interpreting manipulation actions: From language to execution,” in Advances in Intelligent Systems and Computing. Available at: https://doi.org/10.1007/978-3-319-27146-0\_14.

Dharbaneshwer, S.J. et al. (2021) “A finite element based simulation framework for robotic grasp analysis,” Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 235(13). Available at: https://doi.org/10.1177/0954406220951596.

Elbanhawi, M. and Simic, M. (2014) “Sampling-based robot motion planning: A review,” IEEE Access. Available at: https://doi.org/10.1109/ACCESS.2014.2302442.

Fan, Z. (2023) “Multi-Point path planning for robots based on deep reinforcement learning,” in Journal of Physics: Conference Series. Available at: https://doi.org/10.1088/1742-6596/2580/1/012048.

Ganesan, M. et al. (2024) “A Comprehensive Review on Deep Learning-Based Motion Planning and End-to-End Learning for Self-Driving Vehicle,” IEEE Access, 12, pp. 66031–66067. Available at: https://doi.org/10.1109/ACCESS.2024.3394869.

Garrett, C.R. et al. (2021) “Integrated Task and Motion Planning,” Annual Review of Control, Robotics, and Autonomous Systems. Available at: https://doi.org/10.1146/annurev-control-091420-084139.

Ichnowski, J., Avigal, Y., et al. (2020) “Deep learning can accelerate grasp-optimized motion planning,” Science Robotics, 5(48). Available at: https://doi.org/10.1126/scirobotics.abd7710.

Ichnowski, J., Danielczuk, M., et al. (2020) “GOMP: Grasp-Optimized Motion Planning for Bin Picking,” in Proceedings - IEEE International Conference on Robotics and Automation. Available at: https://doi.org/10.1109/ICRA40945.2020.9197548.

Lavalle, S.M. (2006) PLANNING ALGORITHMS. Available at: https://msl.cs.uiuc.edu/planning/bookbig.pdf (Accessed: August 4, 2024).

Li, G. et al. (2023) “Design of Digital Planner and 3D Vision System for Robot Bin Picking,” in 2023 8th IEEE International Conference on Advanced Robotics and Mechatronics, ICARM 2023. Available at: https://doi.org/10.1109/ICARM58088.2023.10218895.

Li, T.-Y. and Huang, P.-Z. (2007) “Planning Versatile Motions for Humanoid in a Complex Environment,” in Humanoid Robots: New Developments. Available at: https://doi.org/10.5772/4888.

Liniger, A. and van Gool, L. (2020) “Safe Motion Planning for Autonomous Driving using an Adversarial Road Model,” in Robotics: Science and Systems. Available at: https://doi.org/10.15607/RSS.2020.XVI.044.

Manzinger, S., Pek, C. and Althoff, M. (2021) “Using Reachable Sets for Trajectory Planning of Automated Vehicles,” IEEE Transactions on Intelligent Vehicles, 6(2). Available at: https://doi.org/10.1109/TIV.2020.3017342.

Muhayyuddin, Akbari, A. and Rosell, J. (2015) “Ontological physics-based motion planning for manipulation,” in IEEE International Conference on Emerging Technologies and Factory Automation, ETFA. Available at: https://doi.org/10.1109/ETFA.2015.7301404.

Prats, M., Sanz, P.J. and del Pobil, A.P. (2007) “Task-oriented grasping using hand preshapes and task frames,” in Proceedings - IEEE International Conference on Robotics and Automation. Available at: https://doi.org/10.1109/ROBOT.2007.363582.

Qureshi, A.H. et al. (2019) “Motion planning networks,” in Proceedings - IEEE International Conference on Robotics and Automation. Available at: https://doi.org/10.1109/ICRA.2019.8793889.

Rosell, J. et al. (2019) “Planning grasping motions for humanoid robots,” in International Journal of Humanoid Robotics. Available at: https://doi.org/10.1142/S0219843619500415.

Rubert, C. et al. (2018) “Characterisation of Grasp Quality Metrics,” Journal of Intelligent and Robotic Systems: Theory and Applications, 89(3–4). Available at: https://doi.org/10.1007/s10846-017-0562-1.

Rudorfer, M., Hartvich, J. and Vonásek, V. (no date) A Framework for Joint Grasp and Motion Planning in Confined Spaces.

Tsuji, T. et al. (2010) “Grasp Planning for a Multifingered Hand with a Humanoid Robot,” Journal of Robotics and Mechatronics, 22(2). Available at: https://doi.org/10.20965/jrm.2010.p0230.

Vahrenkamp, N. et al. (2010) “Integrated grasp and motion planning,” in Proceedings - IEEE International Conference on Robotics and Automation. Available at: https://doi.org/10.1109/ROBOT.2010.5509377.

Wan, W. and Harada, K. (2016) “Integrated assembly and motion planning using regrasp graphs,” Robotics and Biomimetics, 3(1). Available at: https://doi.org/10.1186/s40638-016-0050-2.

Wan, W. and Harada, K. (2017) “Regrasp planning using 10,000s of grasps,” in IEEE International Conference on Intelligent Robots and Systems. Available at: https://doi.org/10.1109/IROS.2017.8206011.

Wang, L., Xiang, Y. and Fox, D. (2020) “Manipulation Trajectory Optimization with Online Grasp Synthesis and Selection,” in Robotics: Science and Systems. Available at: https://doi.org/10.15607/RSS.2020.XVI.033.

Yan, W. et al. (2019) “Precision Grasp Planning for Multi-Finger Hand to Grasp Unknown Objects,” Robotica, 37(8). Available at: https://doi.org/10.1017/S0263574719000031.

Zafra-Urrea, R.M., López-Damian, E. and Santana-Díaz, A. (2023) “Grasp Planning Based on Metrics for Collaborative Tasks Using Optimization,” Applied Sciences (Switzerland), 13(17). Available at: https://doi.org/10.3390/app13179603.

Zhang, H. and Zhu, Z. (2020) “Sampling-based motion planning for free-floating space robot without inverse kinematics,” Applied Sciences (Switzerland), 10(24). Available at: https://doi.org/10.3390/app10249137.11. Appendices

# Appendix