# Integrated Grasp and motion planning

# Abstract

# Acknowledgements

Table of Contents

[Integrated Grasp and motion planning 1](#_Toc173411730)

[Abstract 1](#_Toc173411731)

[Acknowledgements 1](#_Toc173411732)

[List of Figures 4](#_Toc173411733)

[List of Tables 4](#_Toc173411734)

[List of Abbreviations 4](#_Toc173411735)

[Introduction 4](#_Toc173411736)

[Project background 4](#_Toc173411737)

[Problem Statement 4](#_Toc173411738)

[Research Objectives 4](#_Toc173411739)

[Research Questions/Hypotheses 4](#_Toc173411740)

[Significance of the Study 4](#_Toc173411741)

[Scope and Limitations 4](#_Toc173411742)

[Thesis Structure 4](#_Toc173411743)

[Literature Review 4](#_Toc173411744)

[Introduction 4](#_Toc173411745)

[Overview of Grasp Planning 4](#_Toc173411746)

[Overview of Motion Planning 4](#_Toc173411747)

[Integrated Grasp and Motion Planning 4](#_Toc173411748)

[Existing Methods and Approaches 4](#_Toc173411749)

[Gaps in the Literature 4](#_Toc173411750)

[Summary 4](#_Toc173411751)

[Methodology 5](#_Toc173411752)

[1. Introduction 5](#_Toc173411753)

[Research Design 5](#_Toc173411754)

[2. Experimental Setup 5](#_Toc173411755)

[2.1. Simulation Environment 5](#_Toc173411756)

[2.2. Robot and Environment Configuration 5](#_Toc173411757)

[3. Algorithms Implementation 5](#_Toc173411758)

[3.1. Description of Algorithms 5](#_Toc173411759)

[RRT\*: 5](#_Toc173411760)

[3.2. Integration with the Robot 6](#_Toc173411761)

[4. Evaluation Metrics 6](#_Toc173411762)

[4.1. Path Length 6](#_Toc173411763)

[4.2. Planning Time 7](#_Toc173411764)

[4.3. Success Rate 7](#_Toc173411765)

[5. Experimental Procedure 7](#_Toc173411766)

[5.1. Repeated Trials 7](#_Toc173411767)

[5.2. Varying Conditions 7](#_Toc173411768)

[5.3. Data Collection 7](#_Toc173411769)

[6. Data Analysis 7](#_Toc173411770)

[6.1. Statistical Analysis 7](#_Toc173411771)

[6.2. Visualization 8](#_Toc173411772)

[7. Summary 8](#_Toc173411773)

[11. Integrated Grasp and Motion Planning Approach 8](#_Toc173411774)

[13. Discussion 8](#_Toc173411775)

[14. Conclusion 8](#_Toc173411776)

[15. References 9](#_Toc173411777)

[16. Appendices 9](#_Toc173411778)

# List of Figures

# List of Tables

# List of Abbreviations

# Introduction

## Project background

## Problem Statement

## Research Objectives

## Research Questions/Hypotheses

## Significance of the Study

## Scope and Limitations

## Thesis Structure

# Literature Review

## 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).

This literature review aims to provide a comprehensive overview of the current state of knowledge in this domain, and highlights key concepts, methodologies, and findings from previous studies. Integrated grasp and motion planning involves the simultaneous consideration of how a robot should grasp an object and how it should move both the robot and the object to achieve a desired outcome. This dual planning approach is essential for enhancing the efficiency and effectiveness of robotic operations in complex environments.

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 analyzing 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.

## 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).

Task-oriented grasp planning algorithms have 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 analyzing 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.

These references collectively emphasize the critical role of grasp planning in robotics and will showcase advancements in methodologies and technologies to optimize grasp points, stability, and quality metrics for efficient and effective robotic interactions with objects.

**Major Theories and Models**

Several theories and models underpin grasp planning:

* **Force Closure**: Ensures that the robot’s grip can resist external forces from any direction, and can provide a stable grasp.
* **Form Closure**: Achieves stability by constraining the object’s movement through the geometric arrangement of contact points.
* **Task-Oriented Grasping**: Focuses on optimizing the grasp for the specific task the robot is intended to perform, considering factors like object manipulation and placement.

**Techniques and Algorithms**

Grasp planning techniques and algorithms have evolved significantly. Some of the prominent methods include:

* **Analytical Approaches**: Use mathematical models to predict and evaluate potential grasps based on object geometry and physical properties.
* **Sampling-Based Methods**: Generate a large number of potential grasps and evaluate them based on predefined criteria to select the best option.
* **Machine Learning Approaches**: Leverage large datasets and learning algorithms to train models that can predict successful grasps based on visual and tactile data.

**Recent Advances**

Recent research in grasp planning has focused on enhancing the robustness and adaptability of robotic grasps. Key advancements include:

* **Deep Learning Models**: These models use neural networks to learn complex grasping strategies from vast amounts of data, improving the robot's ability to generalize across different objects.
* **Sensor Integration**: Combining data from multiple sensors, such as cameras and tactile sensors, to provide a more comprehensive understanding of the object and its environment.
* **Real-Time Grasp Planning**: Developing algorithms that can operate in real-time, allowing robots to adapt to changes in the environment and object positioning dynamically.

**Challenges and Future Directions**

Despite significant progress, grasp planning remains a challenging problem due to:

* **Object Diversity**: The wide variety of objects in terms of shape, size, and material makes it difficult to develop a one-size-fits-all grasping solution.
* **Dynamic Environments**: Robots must be able to adapt their grasps in real-time to changes in the environment or object position.
* **Computational Efficiency**: Ensuring that grasp planning algorithms can operate quickly enough for real-time applications remains a significant hurdle.

Future research is likely to focus on further integrating machine learning approaches with traditional methods, enhancing the robot's ability to learn from experience and adapt to new situations. Additionally, improving sensor technology and computational efficiency will be critical for advancing the field of grasp planning.

## Overview of Motion Planning

Motion planning is a crucial technology in robotics that involves breaking down complex motion tasks into a series of discrete actions that can be executed (Fan, 2023). It plays a vital role in various 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). These algorithms aim to enhance the safety and autonomy of robots in their operational environments.

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).

Various paradigms have been investigated in motion planning, such as the sampling approach, which involves generating collision-free configurations randomly in free space to create a roadmap (Dalibard & Laumond, 2009). This approach has been successful in addressing motion planning problems efficiently. Moreover, motion planning is not limited to traditional robotics but extends to applications like humanoid robots in complex environments (Li & Huang, 2007).

#### Key Concepts and Definitions

Motion planning involves determining a feasible path or sequence of movements that a robot must follow to achieve a specific goal while avoiding obstacles. Key concepts include:

* **Configuration Space (C-Space)**: A representation of all possible positions and orientations of the robot. Each point in this space corresponds to a unique state of the robot.
* **Path Planning**: The process of finding a collision-free path from a start configuration to a goal configuration within the configuration space.
* **Trajectory Planning**: Extends path planning by considering the timing and dynamics of the robot’s movements, ensuring smooth and feasible trajectories.

#### Major Theories and Models

Several theories and models are central to motion planning:

* **Graph-Based Methods**: These include grid-based approaches and roadmaps, such as Probabilistic Roadmaps (PRM) and Rapidly-exploring Random Trees (RRT), which represent the environment as a graph and use search algorithms to find paths.
* **Optimization-Based Methods**: These methods formulate motion planning as an optimization problem, where the goal is to minimize a cost function that could represent distance, time, energy, or other criteria.
* **Sampling-Based Methods**: These approaches generate random samples in the configuration space to construct feasible paths, providing solutions where deterministic methods might struggle.

#### Techniques and Algorithms

Motion planning has seen numerous techniques and algorithms developed over the years, including:

* **Probabilistic Roadmaps (PRM)**: Constructs a roadmap by randomly sampling the configuration space and connecting these samples to form a graph. Paths are then found by searching this graph.
* **Rapidly-exploring Random Trees (RRT)**: Builds a tree by incrementally expanding random samples from the configuration space, aiming to rapidly explore large spaces.
* A Algorithm\*: A graph search algorithm that finds the shortest path by combining the cost to reach a node and the estimated cost to reach the goal.

#### Recent Advances

Recent advancements in motion planning focus on improving efficiency, robustness, and adaptability. Key developments include:

* **Multi-Robot Motion Planning**: Addressing the complexities of coordinating multiple robots, including collision avoidance and cooperative task execution.
* **Real-Time Motion Planning**: Developing algorithms capable of operating in real-time, enabling robots to adapt their paths dynamically to changes in the environment.
* **Machine Learning Integration**: Utilizing machine learning to predict and optimize paths based on prior experience and data, enhancing the robot's ability to navigate complex environments.

#### Challenges and Future Directions

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

* **High-Dimensional Spaces**: Robots with many degrees of freedom, such as humanoid robots, present significant challenges due to the complexity of their configuration spaces.
* **Dynamic and Uncertain Environments**: Ensuring robust and safe navigation in environments that change over time or have uncertain elements is a major ongoing challenge.
* **Computational Demands**: Developing algorithms that are both efficient and capable of handling complex planning tasks in real-time remains a critical focus.

Future research in motion planning is likely to emphasize further integration of AI and machine learning, enabling more intelligent and adaptable planning strategies. Advances in sensor technology and computational power will also play a crucial role in overcoming current limitations and enhancing the capabilities of motion planning systems.

## 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).

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).

## Existing Methods and Approaches

## Gaps in the Literature

## Summary

# Methodology

## 1. Introduction

This section describes the methodology used to compare four path planning algorithms—RRT\*, IKRRT, IKRRTOptimized, and JPlusRRT—in terms of path length, planning time, and success rate. The comparison aims to identify the strengths and weaknesses of each algorithm in different robotic path planning scenarios.

## Research Design

## 2. Experimental Setup

### 2.1. Simulation Environment

### 2.2. Robot and Environment Configuration

The robot used in the experiments is a 6-DOF manipulator with a gripper.

## 3. Algorithms Implementation

## 3.1. Description of Algorithms

### RRT\*:

The RRT\* (Rapidly exploring Random Tree Star) algorithm is an incremental sampling-based method used in motion planning, which is designed to ensure both probabilistic completeness and asymptotic optimality. This algorithm incrementally builds a tree of feasible trajectories rooted at the initial state, exploring the Configuration space in a manner that tends towards the goal configuration while optimizing the path. The following details highlight the key features and process of the RRT\* algorithm as discussed in the referenced paper by Karaman and Frazzoli (Karaman and Frazzoli, 2011).

#### Key Features of RRT\*

1. **Asymptotic Optimality**: Unlike its predecessor, RRT, the RRT\* algorithm is asymptotically optimal. This means that as the number of samples approaches infinity, the solution path converges to the optimal path.
2. **Probabilistic Completeness**: RRT\* retains the property of probabilistic completeness, ensuring that it will find a solution if one exists given sufficient time and samples.
3. **Incremental and Anytime Nature**: RRT\* is an incremental algorithm, meaning it can provide a feasible solution quickly and then improve the solution over time as more samples are added.

#### Algorithm Steps

1. **Initialization**: The algorithm starts with an initial state and initializes an empty tree.
2. **Sampling**: At each iteration, a random sample from the state space is generated.
3. **Nearest Neighbor Search**: The nearest vertex in the tree to the random sample is identified.
4. **Steering**: A new node is generated by moving from the nearest neighbor towards the random sample, constrained by a step size (η).
5. **Collision Checking**: The new node is added to the tree only if the path from the nearest neighbor to the new node is collision-free.
6. **Cost Calculation and Rewiring**:
   * **Cost Calculation**: For each new node added, the algorithm calculates the cost from the start node to this new node.
   * **Rewiring**: The algorithm attempts to connect the new node to nearby nodes (within a radius). If the new connection offers a lower cost path to these nearby nodes, the tree is rewired to reflect this improved path.
7. **Goal Checking**: The algorithm checks if the newly added node brings the tree closer to the goal and potentially connects it.

#### Important Parameters

* **γ (Gamma)**: A scaling factor used to determine the radius for rewiring connections. It ensures that the rewiring process considers a sufficient number of nearby nodes.
* **η (Eta)**: The maximum step size that the algorithm can move towards the random sample in one iteration.

IKRRT: A variant of RRT that incorporates inverse kinematics to handle high-dimensional configuration spaces.

IKRRTOptimized: An enhanced version of IKRRT with additional optimization techniques for better path quality.

JPlusRRT: An innovative RRT variant that combines Jacobian-based steering with RRT principles for improved performance.

## 3.2. Integration with the Robot

Each algorithm was adapted to work with the 6-DOF manipulator. Modifications were made to ensure compatibility with the robot’s kinematics and dynamics, and to handle the specific constraints of the simulation environment.

## 4. Evaluation Metrics

## 4.1. Path Length

Path length is measured as the sum of the Euclidean distances between consecutive waypoints in the path. It is a crucial metric as shorter paths are generally more efficient.

## 4.2. Planning Time

Planning time is recorded as the time taken from the start of the algorithm until a valid path is found. Timing functions in Python (e.g., time.time()) were used to measure this metric.

## 4.3. Success Rate

Success rate is calculated as the percentage of trials in which an algorithm successfully finds a valid path within a predefined time limit (e.g., 60 seconds).

# 5. Experimental Procedure

## 5.1. Repeated Trials

Each algorithm was tested in 1000 trials to ensure statistical significance. Each trial involves finding a path from a given start to a goal position with varying obstacle configurations.

## 5.2. Varying Conditions

Experiments were conducted under varying conditions, including different start and goal positions and different numbers of obstacles in different environments. This helps in understanding the robustness and adaptability of each algorithm.

## 5.3. Data Collection

Data on path length, planning time, and success rate were collected during each trial. Automated scripts were used to log these metrics for subsequent analysis.

# 6. Data Analysis

## 6.1. Statistical Analysis

The collected data were analyzed using statistical methods such as mean, standard deviation, and ANOVA to compare the performance of the algorithms. These methods help in identifying significant differences in performance metrics.

## 6.2. Visualization

Results were visualized using bar charts and line graphs to compare the path length, planning time, and success rate of each algorithm. These visualizations provide a clear and concise way to present the findings.

## 6.3. Summary

The methodology outlined above ensures a comprehensive and systematic comparison of the four path planning algorithms. By evaluating path length, planning time, and success rate under varied conditions, the study aims to provide insights into the efficiency and effectiveness of each algorithm in robotic path planning.

# 7. Integrated Grasp and Motion Planning Approach

* Introduction
* Theoretical Framework
* Algorithm Design
  + Grasp Planning Algorithm
  + Motion Planning Algorithm
  + Integration Strategy
* Implementation Details
  + Software Implementation
  + Hardware Implementation
* Summary

# 8. Discussion

* Introduction
* Interpretation of Results
* Comparison with Existing Methods
* Implications of Findings
* Limitations of the Study
* Recommendations for Future Research
* Summary

# 9. Conclusion

* Summary of Research
* Key Findings
* Contributions to the Field
* Practical Applications
* Final Thoughts

# 10. References

# 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, 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.

# 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.

# 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. (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.

# 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