Student Conference Proceedings
Vol 1, No 1, 2025, Article ID 1968, 4 Pages

Student Conference Lübeck

Proceedings Article

Path Planning for Medical Robots – Approaches and Experimental Evaluation –

Tim Stichnoth [a,*] · Dennis Kundrat [b,c,*] · Daniel Reichard [d]

[a] Student of Robotics and Autonomous Systems, Universität zu Lübeck, Lübeck, Germany
[b] Institute for Electrical Engineering in Medicine, Universität zu Lübeck, Lübeck, Germany
[c] Fraunhofer Research Institution for Individualized and Cell-Based Medical Engineering IMTE, Lübeck, Germany
[d] Carl Zeiss AG, ZEISS Innovation Hub @ KIT, Eggenstein-Leopoldshafen, Germany
[*] Corresponding author, email: tim.stichnoth@student.uni-luebeck.de; dennis.kundrat@imte.fraunhofer.de

Received 03 February 2025; Accepted 25 August 2025; Published online 16 September 2025

Abstract

Path planning is a fundamental challenge in medical robotics, demanding precision and safety in complex environments. This paper reviews traditional and AI-based path planning approaches, focusing on their applicability in medical settings. A simulation-based experimental framework was developed, incorporating a UR5e robotic arm and NVIDIA Isaac Sim. To assess the framework's capabilities, a path planning experiment was conducted comparing the RRT algorithm and the cuRobo motion planner by NVIDIA, with cuRobo achieving a 35.6 % improvement over the RRT algorithm. While the experiment was limited to a reduced scenario, the results illustrate the potential of the setup to evaluate key metrics such as computational efficiency, safety margins, and path optimality. The study highlights the strengths of the experimental framework and its components as a foundation for future, more complex investigations into path planning in medical environments.

I. Introduction

Path planning in medical robotics requires high precision and adaptability in dynamic environments. The complexity of human anatomy, factors like tissue deformation or organ movement, often result in a cluttered environment, making it especially challenging. With medical robotics becoming increasingly integrated into clinical settings, developing efficient path planning strategies is crucial for enhancing surgical precision and reducing patient risks [1]. Traditional methods often struggle in such dynamic settings, while AI-based approaches offer promising solutions through learning and adaptation [2]. However, their real-time safety and reliability remain underexplored. This paper reviews common path planning methods, both traditional and AI-based, and presents an experimental simulation framework to assess their performance for real-world medical applications.

II. Methods and materials

Path planning involves determining a feasible trajectory from a start to a goal pose of a robot's end effector while avoiding obstacles. The concept of configuration space is central, describing all possible robot states, which in turn are represented by a vector $q \in \mathbb{R}^n$ of joint angle values for a robot with n joints. This section reviews key path planning approaches in robotics, focusing on both traditional and AI-based methods, and introduces an experimental simulation framework used for their evaluation.

II.I. Traditional Path Planning

Traditional path planning methods have been widely used in robotic path planning. This section provides an overview of two key approaches.

II.I.1. Rapidly-exploring Random Tree (RRT)

The RRT algorithm is a path planning algorithm that utilizes a random sampling technique to explore the configuration space of a robot. The algorithm incrementally constructs a tree from the initial configuration, expanding towards the goal until either the target is reached or a predefined exploration threshold is met [3].

As shown in Fig. 1 the algorithm proceeds as follows: The process begins by selecting the initial configuration, q_{start}, and then a random configuration, q_{rand}, is sampled from the configuration space. The nearest tree node, q_{near}, is identified, and a new node, q_{new}, is generated along the path from q_{near} to q_{rand} using a fixed step size δ. If q_{new} does not result in a collision, it is added to the tree. The distance between q_{new} and q_{goal} is then evaluated. If q_{new} lies within the defined domain, surrounding the goal, a valid path from the start to the goal is obtained. If not, the procedure is repeated until the goal is reached or the maximum number of samples is exceeded [3].

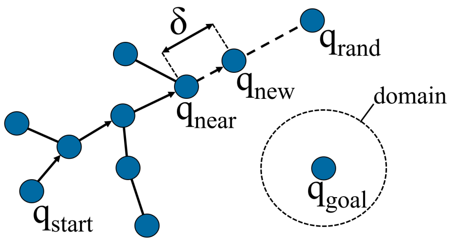

Figure 1: The RRT expansion method. The RRT algorithm incrementally builds a tree by sampling random configurations and extending towards the goal configuration. Adapted from [3].

II.I.2. Velocity Potential Field (VPF)

The optimization-based VPF-algorithm represents the robot's environment using attractive and repulsive potential fields. The target is modeled as an attractive potential, which generates a velocity that moves the robot's end effector toward the goal. Obstacles are modeled as repulsive potentials, which generate velocities that push the robot away [1]. The attractive potential U_{att} is given by:

$$U_{\text{att}}(q) = \frac{1}{2}\zeta\rho_g^2\,, \tag{1}$$

where q is the current configuration of the robot, ρ_g is the Cartesian distance between the end effector and the target, and ζ is the attractive potential gain. The repulsive potential U_{rep} is defined as:

$$U_{\text{rep}}(q) = \begin{cases} \frac{1}{2}k\left(\frac{1}{\rho_b} - \frac{1}{\rho_0}\right)^2 & \text{if } \rho_b \leq \rho_0 \\ 0 & \text{if } \rho_b > \rho_0 \end{cases}\,, \tag{2}$$

where ρ_b is the minimum distance between the robot and the obstacle, ρ_0 the threshold beyond which no repulsive force is applied and k the repulsive potential gain [1]. The attractive and repulsive velocity, induced by the potential fields are then determined by taking the negative gradients of the potentials and converted from Cartesian to joint space using the robot's Jacobian afterwards [1].

II.II. AI-Based Path Planning

Unlike traditional methods, AI-based path planning approaches leverage machine learning to enable robots to adapt to complex, dynamic environments and improve their decision-making capabilities over time. Two exemplary approaches are presented in this section.

II.II.1. Q-Learning (QL)

Q-learning, as proposed in [2], is a model-free reinforcement learning algorithm, where an agent (e.g., a robot arm) learns an optimal policy by interacting with its environment. The first step in the QL process involves discretizing the robot's configuration space, based on a predefined angular resolution of the robot's joints. The robot can then perform actions, which include rotating each joint, or keeping it stationary. The agent receives a reward or penalty based on the consequences of its actions. Rewards are assigned for progressing toward the goal, while penalties are given for undesirable outcomes, such as collisions with obstacles or unsafe movements [2].

The Q-learning algorithm focuses on iteratively refining the action-value function, represented by Q-values, which estimate the expected future reward for taking a particular action in a given state. During training, the Q-values are updated by iteratively adjusting the value of a state-action pair based on the observed reward and the maximum expected future reward [2].

Through this iterative learning process, the robot gradually improves its decision-making ability, converging towards an optimal policy that minimizes the risk of collisions and efficiently guides the robot arm towards the target. Q-learning allows the robot to adapt to dynamic environments by continually refining its policy based on feedback from the environment [2].

II.II.2. NVIDIA cuRobo

The cuRobo motion planner by NVIDIA utilizes numerous advanced AI-based methods. A central aspect of its approach is the use of a GPU-enhanced optimization method, designed to evaluate multiple solutions in parallel. By incorporating a parallel noisy line search, it quickly assesses and selects effective step directions, significantly accelerating the convergence process for complex planning tasks [4]. To further enhance the optimization, cuRobo combines gradient-based methods with particle-based optimization. This hybrid strategy

10.18416/SCP.2025.1968

starts by sampling multiple trajectory candidates, refining them using a weighted evaluation of costs, and transitioning to gradient-based optimization for precise local improvements. This ensures thorough exploration of the solution space and leads to robust outcomes [4]. Another innovation lies in a differentiable framework for continuous collision checking, which uses signed distance calculations to measure proximity to obstacles and compute optimization gradients. This method integrates seamlessly with different environment models, such as geometric shapes, mesh-based representations, and voxel grids. These AI-driven techniques enable cuRobo to achieve efficient, high-speed motion planning while addressing complex constraints [4].

II.III. Experimental Evaluation Framework

In this section an experimental evaluation framework for analyzing and comparing the performance and applicability of path planning approaches, like the examples presented earlier, is introduced.

II.III.1. UR5e Robot Arm

The UR5e robot arm, manufactured by Universal Robots, is a 6-DOF articulated manipulator designed for a range of precise tasks. It features a working radius of 850 mm and a maximum payload capacity of 5 kg. The arm offers high repeatability of $\pm$ 0.03 mm, which is crucial for applications requiring fine control, such as medical procedures. With a maximum speed of 1 $\frac{\mathrm{m}}{\mathrm{s}}$ and integrated torque sensing, the UR5e is capable of dynamic and responsive movements. Its compact and lightweight design allows for seamless integration into various medical environments [5].

II.III.2. NVIDIA Isaac Sim

NVIDIA Isaac Sim is a comprehensive simulation platform that provides realistic physics simulations, enabling precise interaction between robots and collision objects. The platform supports dynamic obstacle detection and real-time feedback, making it ideal for evaluating path planning algorithms in complex environments. Additionally, Isaac Sim integrates sensor simulations like depth cameras and lidar to enhance collision detection accuracy [6].

III. Results and discussion

In this section the results of an exemplary path planning experiment using both the basic RRT algorithm and the cuRobo motion planner explained above are presented and discussed using an evaluation metric.

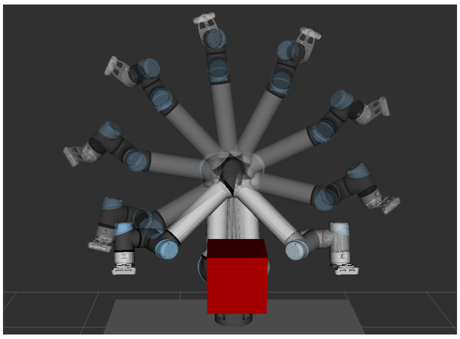

Figure 2: Path Planning Experiment. The RRT algorithm is used to determine a valid path to move the robot's end effector from the initial pose (right) to the goal pose (left). The calculated path is visualized by the transparent robot configurations in between.

III.I. Path Planning Experiment

The experimental evaluation framework, implemented using ROS2 and MoveIt2, is used to conduct an exemplary simulated path planning experiment utilizing the RRT algorithm from the Open Motion Planning Library and the cuRobo motion planner, both in their default configurations. The robot model, consisting of the UR5e robot itself and a depth camera attached to it as an end effector, is supposed to execute a movement from a given initial pose to a goal pose while making sure that collisions with both a modeled bottom plate and a basic collision object which is visualized as a red box in Fig. 2 are prevented. Both poses of the robot's end effector are defined relative to the world coordinate system, placed at the coordinate origin, and as a tuple of seven elements $(x, y, z, q_x, q_y, q_z, q_w)$, where the first three describe the position in meters and the last four are part of a quaternion, used to represent the orientation. The initial pose is defined as (0.40, 0.92, 1.27, -0.04, 0.63, 0.04, 0.78) and the goal pose as (0.40, 0.12, 1.27, -0.04, 0.63, 0.04, 0.78). The robot's base coordinate system is placed at (0.2, 0.5, 1, 0, 0, -0.71, 0.71).

III.II. Evaluation Metric

For evaluating the performance of path planning techniques in a dynamic medical environment a weighted sum of the following three key parameters can be considered:

- **Real-Time Responsiveness (R)**: Time required for the algorithm to plan the robot's path. Faster planning results in a better score, crucial for real-time medical applications.
- **Safety Margin (S)**: The distance maintained between the robot and obstacles. Larger safety margins are essential for preventing collisions and ensuring patient safety.

10.18416/SCP.2025.1968

- **Path Optimality (O)**: The efficiency of the path in terms of absolute joint motion. Paths with less absolute joint motion are preferred, minimizing robot motion and maximizing efficiency.

A path the planner needs more than $\hat{R} = 2\,\mathrm{s}$ to calculate is assumed to be not applicable for a medical application. The robot should also keep a safety margin of $\hat{S} = 0.01\,\mathrm{m}$ at all points of the calculated path. For the absolute joint motion one full rotation for each of the six joints is considered to be the worst-case. This would result in $\hat{O} = 37.7\,\mathrm{rad}$. If the calculated path leads to a violation of these limits, it is assumed to be invalid. For a valid path the overall metric M can be calculated as:

$$M = w_1 \frac{R}{\hat{R}} + w_2 \frac{\hat{S}}{S} + w_3 \frac{O}{\hat{O}}, \tag{3}$$

where w_1, w_2, w_3 are the weights for each parameter, which are all set to $\frac{1}{3}$. A score closer to zero indicates better performance in the context of medical robot navigation. Table 1 shows the resulting values for the planning time (R), the safety margin (S), the absolute joint motion (O) and the overall evaluation metric (M) for conducting the described path planning experiment both with the RRT planning alogrithm and the cuRobo motion planner.

Table 1: Path planning results

Planner	R	S	O	M
RRT	0.17 s	0.067 m	13.13 rad	0.194
cuRobo	0.12 s	0.067 m	8.23 rad	0.125

For both planners the minimal distance between the robot and the considered collision objects is 0.067 m and reached in the initial configuration. It is not undercut at any other point of the path. In terms of the remaining parameters, planning time and absolute joint motion, the cuRobo motion planner outperforms the RRT planner which also results in a better score M.

IV. Conclusion

This study has reviewed both traditional and AI-based path planning methods in the context of medical robotics. Traditional approaches such as RRT and VPF offer robust frameworks for structured environments but often lack adaptability to dynamic and complex medical scenarios. In contrast, AI-based methods, such as Q-Learning and the cuRobo motion planner, demonstrate significant potential for real-time responsiveness and optimization in such environments, although challenges related to their integration, robustness, and reliability in dynamic medical scenarios, beyond the scope of the simulation framework, should not be overlooked.

The experimental results underscore the importance of evaluating path planning algorithms against real-world constraints like safety margins, computational efficiency, and path optimality. The proposed metric provides a structured way to assess these methods and compare their applicability for medical environments, with the cuRobo motion planner showing a 35.6 % improvement over the RRT algorithm. It outperformed RRT in terms of both planning time and path optimality, demonstrating its potential for faster and more efficient motion planning in medical applications. Considering additional parameters, such as robustness to uncertainty, which ensures reliable performance under sensor noise, and adaptability to dynamic obstacles, which evaluates navigation around moving obstacles, would further improve the proposed evaluation metric, making the results more robust and meaningful for medical applications.

Future work should also focus on integrating multimodal sensor data and refining AI models to improve adaptability and precision. These advancements will bridge the gap between simulation and clinical use, enabling the evaluation of these models on real-world systems and ensuring the development of safer and more efficient robotic systems in healthcare.

Acknowledgments

The work has been carried out at ZEISS Innovation Hub @KIT, Eggenstein-Leopoldshafen and supervised by Prof. P. Rostalski, Institute for Electrical Engineering in Medicine, Universität zu Lübeck.

Author's statement

Conflict of interest: Authors state no conflict of interest.

References

[1] X. Xia, T. Li, S. Sang, Y. Cheng, H. Ma, Q. Zhang, and K. Yang. Path planning for obstacle avoidance of robot arm based on improved potential field method. *Sensors*, 23(7), 2023, doi:10.3390/s23073754.

[2] M. Ji, L. Zhang, and S. Wang, A path planning approach based on q-learning for robot arm, in *2019 3rd International Conference on Robotics and Automation Sciences (ICRAS)*, 15–19, 2019. doi:10.1109/ICRAS.2019.8809005.

[3] Y. Liu and G. Zuo, Improved rrt path planning algorithm for humanoid robotic arm, in *2020 Chinese Control And Decision Conference (CCDC)*, 397–402, 2020. doi:10.1109/CCDC49329.2020.9164659.

[4] B. Sundaralingam, S. K. S. Hari, A. Fishman, C. Garrett, K. V. Wyk, V. Blukis, A. Millane, H. Oleynikova, A. Handa, F. Ramos, N. Ratliff, and D. Fox, Curobo: Parallelized collision-free minimum-jerk robot motion generation, 2023. arXiv: 2310.17274 [cs.RO].

[5] Universal Robots, General Catalogue, 11_2023_General-Catalogue_iREX_A4_PRINT.indd, Accessed: 2024-12-05, 2023.

[6] NVIDIA Corporation, Isaac Sim, Accessed: 2024-12-05, 2023. URL: https://developer.nvidia.com/isaac-sim.