Benchmarking of robot arm motion planning in cluttered environments

Shibao Yang
Dept. Computer Science
University of York
York, United Kingdom
shibao.yang@york.ac.uk

Pengcheng Liu
Dept. Computer Science
University of York
York, United Kingdom
pengcheng.liu@york.ac.uk

Nick Pears
Dept. Computer Science
University of York
York, United Kingdom
nick.pears@york.ac.uk

2023 28th International Conference on Automation and Computing (ICAC) | 979-8-3503-3585-9/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICAC57885.2023.10275283

***Abstract*—Motion planning is essential for robotic automation across various industries. However, generalizing research outcomes has been challenging due to the narrow focus of previous work on a specific robot arm system. Here, we take a broader approach by exploring the combinations of three popular robot arm systems, three levels of clutterness in the environment, and twelve popular motion planners. To conduct the necessary performance analysis, we employ Motionbenchmaker tool and introduce a sensitivity metric. Our approach is structured and accessible, enabling the identification of the best-performing planner-robotic arm combinations. We find that the LBKPIECE, RRTConnect, and BKPIECE planners with Franka and UR5 offers the best balance of effectiveness and robustness. More generally, our results help researchers and practitioners make informed decisions when selecting robotic arms and motion planners, for use in environments with different degrees of clutterness.**

***Index Terms*—Motion planning, Benchmarking, Motionbenchmaker, Robotic arms**

I. Introduction

Motion planning is essential for industrial robots to ensure precise movements and avoid obstacles [1]. However, challenges persist, including limited generalizability across robotic arm systems and insufficient examination of cluttered environments [2]–[4]. Several benchmarking studies have been conducted to evaluate and compare the performance of motion planning algorithms and robotic arm systems. For instance, a study [5] focused on the optimization and evaluation of motion planning algorithms in various scenarios, proposing a motion planning pipeline connecting the Open Motion Planning Library (OMPL) with optimized CHOMP or STOMP algorithms. Also [6] performed benchmarking tests on a 7-DOF robotic arm with various controllers to evaluate their accuracy, control efficiency, jitter, and robustness. While these studies provide valuable insights into motion planning algorithm performance, there are still gaps that need to be addressed. [7] introduced the Motionbenchmaker tool to generate and benchmark motion planning datasets. Another study [8] presented an extensible infrastructure for the analysis and visualization of motion planning algorithms. While these studies provide valuable insights into motion planning algorithm performance, various gaps still need to be addressed.

One of the major gaps in existing studies is the limited generalizability of the results to other robotic arm systems, as many studies focused on one robotic arm. Moreover, there is a lack of comprehensive investigation into the impacts of the working environment on motion planning, especially in the context of cluttered environments. The influence of the environment's properties, such as the size of the working space and the dimensions of obstacles, on motion planning performance, remains unexplored. Furthermore, there is a need for a standardized framework that enables the systematic comparison and evaluation of motion planners and robotic arm systems in various environments.

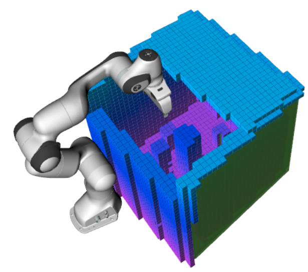


Fig. 1: An illustration of motion planning for Scenario 2. The end-effector of the Franka robot arm is inside the box, moving from one of the objects to the top of the other object without obstacle collisions in the process.

To address this gap, we conduct benchmarking studies that compare the performance of twelve OMPL [9] motion planners used with three different robotic arms: Franka [10], UR5 [11], and Kuka [12], see (TABLE I), in three environments of different levels of clutter. The Motionbenchmaker tool [7] is utilized to facilitate the benchmarking process, providing a unified platform for performing the evaluation of different motion planners and robotic arms. Our experiments investigate the performance of three robotic arms to determine their suitability for motion planning tasks in cluttered environments. The motion planners are tested in three distinct cluttered environments with varying levels of complexity: simple, moderate, and difficult, based on the benchmarks proposed by [13]. These environments will present unique features and require different planning strategies. The performance of the motion planners and robotic arms will be evaluated using the following metrics, as suggested by [14]: time efficiency, success rate, and sensitivity to the *range* parameter.

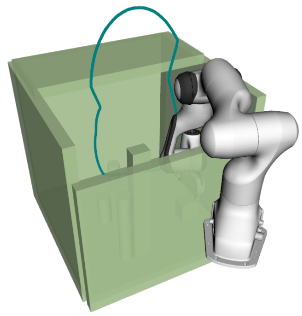


Fig. 2: In Scenario 2, the Franka robot arm's motion planning resulted in a trajectory depicted by a blue line.

TABLE I: Features of the robotic arm

Robotic arm	Feature	Application
Franka	7 DOFs, real-time motion planning, compliance control, advanced sensing capabilities, scalability	Assembly complex mechanical parts in manufacturing, inspections and measurements in research, surgical procedures in healthcare
UR5	6 DOFs, user-friendly interface, safe operation, repeatability, flexibility integration	Picking and placing goods, testing and evaluating new robotic algorithms, assisting to patients
Kuka	6 DOFs, precision and accuracy, high speed and performance, safe operation, customization, integration	It can be used in industrial production to automate the process of placing goods or products onto pallets

Contributions are: (1) exploration and analysis of the influence of the working environment properties on motion planning for robotic arms, with a focus on the size of the working space and the height of obstacles, (2) the evaluation of motion planning methods using three key metrics: time efficiency, success rate, and parameter sensitivity, (3) developing a cost function that can score motion planners across different scenarios, and recommend the appropriate planner based on the specific task requirements, and (4) a comparison of the performance of three robotic arms (Franka, UR5, and Kuka) in various cluttered environments, providing insights into the most efficient and robust planner-arm combinations. Our results enable researchers and practitioners to make informed decisions when selecting robotic arms and motion planners for their specific applications, ultimately improving the efficiency and robustness of robotic systems in complex environments.

In the following section, we review related work on motion planning in benchmarking. Section 3 presents the variations in scenarios and queries, as well as the metrics used for benchmarking. In Section 4, we describe the experimental setup and methodology for the benchmarking. A final section draws conclusions and discusses the implications of the findings.

II. Related works

Chamzas et al. [7] introduced MotionBenchMaker, a tool for generating diverse datasets with various robotic arms for benchmarking. They assessed planners using planning time and best cost but only tested three planners, limiting the analysis. In this paper, we address this limitation by analyzing twelve motion planners across three robotic arm systems and different levels of environmental clutter.

The RRT [15] motion planner uses randomized search and a tree-like structure in the configuration space to find a path between an initial and a goal configuration. RRTConnect [15] is a variant of RRT that generates two trees, one starting from the initial configuration and the other from the goal configuration, and connects them to find a path. RRTstar [15] is another RRT variant that employs cost-based rewiring to dynamically adjust the tree structure and find a lower-cost path. TRRT [16] is a version of RRT that uses adaptive sampling to adjust the distribution of random samples based on the current state of the tree and the progress toward the goal. EST [17] employs randomized search and space expansion principles, where the tree size is dynamically adjusted to focus the search in areas of the configuration space more likely to contain a path to the goal. SBL [18], KPIECE [19], BKPIECE [19], and LBKPIECE [19] are variants of single-query motion planning algorithms that use a graph-based structure in the configuration space to find a path. STRIDE [20] is designed for use in dynamic environments, where the environment changes over time, and FMT (Fast Marching Tree) [21] is a fast marching algorithm that uses a hierarchical tree structure to find a path.

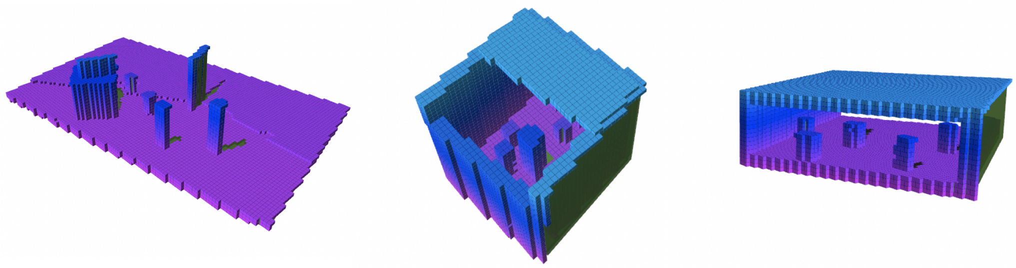
III. Software

We employ the Robot Operating System (ROS) [22] framework, the MoveIt [23] library for planning and executing robotic arm movements, and the Motionbenchmaker [7] repository for benchmarking motion planning algorithms. ROS is an open-source platform for building robot applications, while MoveIt provides a unified interface for performing complex tasks, such as grasping objects and navigating in three-dimensional space. The Motionbenchmaker repository offers a standardized and modular interface for benchmarking motion planning algorithms, supporting a wide range of robotic platforms and planning algorithms for performance evaluation and comparison.

IV. Problem formulation

A. Variation definitions

Motionbenchmaker can generate diverse scenes by introducing random variations to a nominal scene's object poses, both globally and locally. These perturbations are controlled by parameters specified in a configuration file and follow a Gaussian distribution for the probability of the random variable that perturbs the nominal positions of the objects. It can also define start and goal manipulation queries as pose offsets, creating a variety of motion planning problems. By combining scene sampling with problems, motion planning algorithms can be evaluated across a wide range of environments and scenarios under varying conditions influenced by Gaussian-based variations.



(a) Scenario 1 (b) Scenario 2 (c) Scenario 3

Fig. 3: Three scenarios setting: (a) represents a flat surface on which obstacles of different shapes are placed, the simplest scene without space constraints; (b) is a semi-closed box, which compresses a lot of space compared to (a); (c) is similar to a drawer, where the working space is significantly limited compared to (a) and (b).

B. Scenarios

In the first scenario (Fig.3.a), cluttered environments without spatial constraints feature obstacles on a single plane. A scene generation module is utilized to add random noise to the pose of collision objects relative to the global frame in a standard scene, as described in [24]. This study focuses on motion planning, not object grasping, with the robotic arm reaching an object.

The second scenario (Fig.3.b) reduces workspace with a semi-closed box, adding spatial constraints. The robotic arm must plan movements within the restricted space and reach an object, simulating motion planning in a more confined space compared to the previous scenario.

The third scenario (Fig.3.c) resembles a living room drawer with a more restricted workspace and obstacles that cannot be bypassed directly from above. The robotic arm must enter the drawer and reach an object, simulating motion planning in an even more confined space than before.

C. Metrics for selection

Various metrics are used based on time efficiency and success rate. Time efficiency is defined by the mean time taken by motion planners to compute feasible paths, while success rate assesses the percentage of successful path planning attempts. The robustness of the motion planner is evaluated by analyzing the impact of varying the parameter *range* on computation time. This study serves as a foundation for further investigations into refining parameters. In motion planning, the *range* parameter represents a finite interval or a set of discrete values, such as the maximum length of motion segments in tree-based algorithms. Larger *range* values can decrease the number of samples required but increase the complexity of collision checking, while smaller values may simplify these processes, albeit at the expense of slower planning.

D. Planners' score

To evaluate the planners' performance in each scenario and across all scenario-arm combinations, we calculate the average time efficiency (T_{avg}) for each planner in every scenario. Then we normalize the time efficiency scores using the subsequent formula:

$$N_t = 1 - \frac{T_{avg} - \min(T_{avg})}{\max(T_{avg}) - \min(T_{avg})} \quad (1)$$

Here, N_t represents the normalized time efficiency scores, which are inverted and scaled to a range of 0 to 1, with 1 signifying optimal performance.

Apply a weighted average function to aggregate the normalized time efficiency scores (N_t) from each scenario, accounting for the varying difficulty levels of the scenarios with weights w_1, w_2, and w_3. The essential function for computing the weighted average scores (W_{avg}) is presented as follows:

$$W_{avg} = w_1 * N_{t_1} + w_2 * N_{t_2} + w_3 * N_{t_3} \quad (2)$$

In this equation, N_{t_1}, N_{t_2}, and N_{t_3} denote the normalized time efficiency scores for each planner within their respective scenarios. The planners are subsequently ranked according to their weighted average scores.

V. Experiments

We conducted experiments using Motionbenchmaker tool to create scenarios for benchmarking motion planners in conjunction with robotic arms. These scenarios and queries were generated through C++ and Python scripts. For environmental configuration benchmarking, we used single-query planners from OMPL, selecting the most popular motion planners for comparison during parameter sensitivity benchmarking. Each test in Section 1 consisted of 100 runs with a planned timeout of 30 seconds. Planners' scores depend on assigned weights (1/6, 1/3, 1/2) for Scenarios 1, 2, and 3. For Section 2, each parameter *range* trial included 100 runs with a 30-second scheduled timeout, where a 0 mean time indicated all experiments failed. The mean time units in the experiments were seconds, and the success rate was expressed as a percentage. We conducted the experiment on a computer equipped with an Intel i9-11900 processor and an NVIDIA 3050.

A. Environment configuration benchmarking

The performance metrics derived from TABLES II, III, and IV reveal a striking distinction among the assessed planners in their ability to operate under diverse scenarios. Four planners

TABLE II: Scenario 1

Planner name	Franka		UR5		Kuka	
	Mean time (s)	Success rate (%)	Mean time (s)	Success rate (%)	Mean time (s)	Success rate (%)
RRT	1.504	89	4.793	59	1.622	65
RRTConnect	0.014	100	0.069	100	0.070	100
RRTstar	30.003	75	30.040	47	30.030	60
TRRT	1.288	66	5.150	37	1.216	70
EST	0.073	100	1.670	100	0.696	100
SBL	0.042	100	0.354	100	0.397	100
KPIECE	0.060	100	1.771	100	0.617	100
BKPIECE	0.063	100	0.793	100	1.542	100
LBKPIECE	0.066	100	1.507	100	1.579	100
PDST	0.059	99	3.351	97	0.685	100
STRIDE	0.095	100	1.762	100	0.794	100
FMT	0.615	100	5.643	100	22.058	99

TABLE III: Scenario 2

Planner name	Franka		UR5		Kuka	
	Mean time (s)	Success rate (%)	Mean time (s)	Success rate (%)	Mean time (s)	Success rate (%)
RRT	2.196	9	7.935	43	0.048	47
RRTConnect	0.408	100	1.047	100	3.616	99
RRTstar	30.001	4	30.058	31	30.057	52
TRRT	0.592	30	2.343	26	0.069	48
EST	2.239	98	5.228	97	2.086	53
SBL	0.529	100	1.505	99	5.762	100
KPIECE	2.521	94	6.349	92	4.157	57
BKPIECE	0.975	100	3.883	98	9.398	85
LBKPIECE	0.235	100	2.351	99	4.916	100
PDST	1.154	100	6.587	76	2.119	53
STRIDE	1.967	97	6.256	87	2.727	60
FMT	1.989	98	6.240	100	8.476	66

TABLE IV: Scenario 3

Planner name	Franka		UR5		Kuka	
	Mean time (s)	Success rate (%)	Mean time (s)	Success rate (%)	Mean time (s)	Success rate (%)
RRT	2.219	63	2.063	65	5.462	2
RRTConnect	0.092	100	0.330	100	7.351	79
RRTstar	30.002	50	30.023	55	30.005	1
TRRT	3.987	23	5.089	40		0
EST	2.419	95	4.255	88	15.719	5
SBL	0.156	100	0.488	100	6.100	91
KPIECE	1.084	84	4.039	85	15.166	6
BKPIECE	0.571	100	1.543	100	10.484	62
LBKPIECE	0.086	100	1.486	100	4.650	100
PDST	1.648	85	5.822	61	8.602	3
STRIDE	2.183	93	4.414	81	10.424	6
FMT	1.789	90	5.791	100	22.036	17

- RRTConnect, SBL, BKPIECE, and LBKPIECE - exhibit impressive success rates across all three scenarios. The metric employed for evaluation is the normalized time efficiency score, calculated as a weighted average for each scenario. The resulting scores place RRTConnect (0.88) at the forefront, followed by SBL (0.82), LBKPIECE (0.75), and BKPIECE (0.04). Notwithstanding BKPIECE's high success rate, its planning time of approximately 10 seconds is noticeably longer than its counterparts, most apparent in clutter-dense scenarios such as 2 and 3. RRTConnect emerges as the most time-efficient planner across these scenarios, trailed closely by SBL. LBKPIECE ranks third, while BKPIECE falls behind due to its longer planning time.

According to TABLES II, III, and IV, planners such as PDST, STRIDE, EST, KPIECE, and FMT demonstrate moderate success rates across all scenarios. The performance ranking for this group of planners is PDST (0.98), STRIDE (0.90), EST (0.77), KPIECE (0.68), and FMT (0.00). PDST and STRIDE, in particular, display time efficiency almost matching that of the high-performing RRTConnect, SBL, BKPIECE, and LBKPIECE, and are even marginally quicker by 2 seconds in scenario 2. This showcases their commendable motion planning potential in moderately cluttered environments. While EST and KPIECE perform satisfactorily, securing third and fourth ranks respectively, FMT disappointingly underperforms, suggesting it may be less suited for these scenarios.

The TABLES II, III, and IV also highlight RRT, TRRT, and RRTstar, which demonstrate lower success rates in all scenarios. In this category, RRT (1.00) outperforms with the highest score, signifying its competency in handling low success rate scenarios. TRRT, while ranking second, does not quite match RRT's efficiency. RRTstar struggles with the lowest score, indicating it may be inadequate for such scenarios.

Among all, RRTConnect, SBL, and LBKPIECE consistently illustrate superior robustness and efficiency, exhibiting scalability across varied clutter levels. They uphold high success rates and relatively low mean times, which positions them as suitable candidates for a wide range of environments. Despite this, BKPIECE, PDST, and STRIDE show potential and could benefit from further research. Conversely, RRT, RRTstar, and TRRT present less promising results, underlining the need for further optimization of these planners.

B. Parameter sensitivity benchmarking

In the benchmarking analysis illustrated in Fig. 4, LBKPIECE, RRTConnect, and BKPIECE consistently delivered superior time efficiency across all scenarios for Franka, UR5, and Kuka. These planners outperformed others across all test environments. Meanwhile, SBL exhibited strong performance in less complex situations and showed potential for improved results with optimal *range* adjustments.

However, EST, TRRT, and STRIDE generally had higher mean times and lower success rates across all robotic arms. This result points to these planners' comparative inefficiency.

An additional observation was that the majority of the planners displayed lower mean times and higher success rates

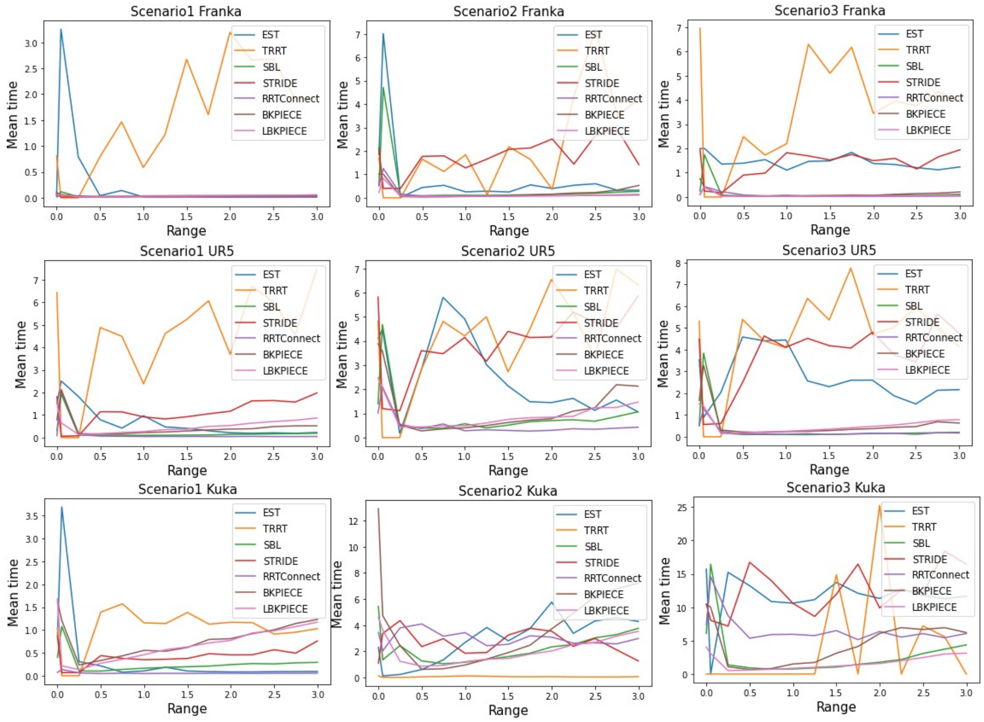


Fig. 4: The present study employed six motion planners, namely EST, TRRT, SBL, STRIDE, RRTConnect, BKPIECE, and LBKPIECE to analyze the difference in mean time, measured in seconds, among robotic arms and planners by varying the parameter *range*. Specifically, the parameter is systematically varied from 0 to 3 with an incremental interval of 0.25. The trials for each parameter range consist of 100 runs with a planned timeout of 30 seconds, where a mean time of 0 seconds indicates that all failed.

within the parameter *range* interval of 0.25 to 0.5. Most of the response curves appeared to stabilize regionally once the parameter value exceeded 1.5. This trend was even noticed for the typically unstable TRRT, which showed a decline in amplitude frequency beyond this point.

In summary, LBKPIECE, RRTConnect, and BKPIECE showcased robust and reliable performance across all robotic arms and scenarios, indicating their versatility. Although SBL required optimal *range* adjustments, it showed promise in specific conditions. In contrast, EST, TRRT, and STRIDE were generally less efficient. It is also important to note that performance varied across robotic arms—Franka, UR5, and Kuka—based on the planner used and the specific *range* parameters.

C. Discussion

In conclusion, LBKPIECE, RRTConnect, and BKPIECE consistently demonstrate time-efficient performance for all three robot arms in the tested scenarios, featuring low mean times, high success rates, and robust performance across various *range* parameter values. EST, and SBL could also be viable options, depending on the desired performance, but their effectiveness may vary depending on specific *range* values and environmental complexity. The influence of the *range* parameter on LBKPIECE, RRTConnect, and BKPIECE's performance is less pronounced in simpler cluttered environments, where these planners can often find feasible paths quickly, regardless of the *range* value. Conversely, for EST and SBL, a smaller *range* value may slow the planning process due to increased samples and connections, while a larger value could accelerate the convergence of trees.

In Scenario 1's open space, planners benefit from multiple path options, resulting in high timeliness and success rates, although obstacle configurations can still impact performance. Scenarios 2 and 3 illustrate that workspace size significantly affects robot arm motion planning. Restricted spaces constrain arm joint mobility, reducing both arm and planner performance in cluttered environments, where obstacles also impede computational efficiency. Nevertheless, fine-tuning the *range* parameter in complex environments can enhance planning performance and enable planners to thrive in otherwise challenging settings. Empirical studies suggest adjusting the *range* parameter to optimize performance across different environments, with a *range* of 0.25 to 0.5 typically yielding good performance. However, in constrained and cluttered environments like Scenario 3, increasing the parameter to 1.5 or higher has shown improved robustness and success rates. This adjustment should account for the specific characteristics and constraints of the environment. Further research is needed to investigate the impact of *range* parameter adjustments on performance in diverse environments.

Concerning the robot arms in Table 1, the Franka, with its 7

DOFs, is well-suited for researchers working on complex tasks requiring precise positioning when paired with LBKPIECE, RRTConnect, and BKPIECE. For those focusing on slightly simpler experiments, the UR5 offers a user-friendly interface and safety features, making it an excellent choice for beginners and algorithm testing. In contrast, the Kuka is more appropriate for applications with higher technical requirements, targeting industrial production and automation.

VI. CONCLUSIONS

In conclusion, this paper presents a comprehensive study on the performance of various motion planners in cluttered environments using three robotic arms: Franka, UR5, and Kuka. The primary focus is to investigate time efficiency, robustness, and parameter sensitivity.

Experimental results show that LBKPIECE, RRTConnect, and BKPIECE consistently exhibit the best time efficiency and robustness across all robotic arms. SBL is potential candidates with reasonable performance, particularly for Franka and UR5 in challenging scenarios. The Franka, paired with LBKPIECE, RRTConnect, and BKPIECE, is ideal for complex tasks and precision. The UR5 suits simpler experiments, beginners, and algorithm testing. Meanwhile, the Kuka targets industrial production and automation.

In summary, our research contributes to the field of motion planning by providing a thorough analysis of the performance of various motion planners and robotic arms in cluttered environments. The insights gained from this study can serve as a valuable recommendation for researchers and practitioners in selecting the most appropriate motion planners and robotic arms for their specific tasks and applications. Future work may explore various environment configurations, such as different obstacle types, sizes and distributions, as well as the interaction between static and dynamic obstacles.

REFERENCES

[1] S. Liu and P. Liu, "A review of motion planning algorithms for robotic arm systems," in *RiTA 2020: Proceedings of the 8th International Conference on Robot Intelligence Technology and Applications*. Springer, 2021, pp. 56–66.

[2] P. Liu, M. N. Huda, L. Sun, and H. Yu, "A survey on underactuated robotic systems: bio-inspiration, trajectory planning and control," *Mechatronics*, vol. 72, p. 102443, 2020.

[3] P. Liu, H. Yu, and S. Cang, "On periodically pendulum-diven systems for underactuated locomotion: a viscoelastic jointed model," in *2015 21st International Conference on Automation and Computing (ICAC)*. IEEE, 2015, pp. 1–6.

[4] P. Liu, G. Neumann, Q. Fu, S. Pearson, and H. Yu, "Energy-efficient design and control of a vibro-driven robot," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1464–1469.

[5] S. Liu and P. Liu, "Benchmarking and optimization of robot motion planning with motion planning pipeline," *The International Journal of Advanced Manufacturing Technology*, pp. 1–13, 2021.

[6] B. Zhang and P. Liu, "Control and benchmarking of a 7-dof robotic arm using gazebo and ros," *PeerJ Computer Science*, vol. 7, p. e383, 2021.

[7] C. Chamzas, C. Quintero-Pena, Z. Kingston, A. Orthey, D. Rakita, M. Gleicher, M. Toussaint, and L. E. Kavraki, "Motionbenchmaker: A tool to generate and benchmark motion planning datasets," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 882–889, 2021.

[8] M. Moll, I. A. Sucan, and L. E. Kavraki, "Benchmarking motion planning algorithms: An extensible infrastructure for analysis and visualization," *IEEE Robotics & Automation Magazine*, vol. 22, no. 3, pp. 96–102, 2015.

[9] I. A. Sucan, M. Moll, and L. E. Kavraki, "The open motion planning library," *IEEE Robotics & Automation Magazine*, vol. 19, no. 4, pp. 72–82, 2012.

[10] S. Haddadin, S. Parusel, L. Johannsmeier, S. Golz, S. Gabl, F. Walch, M. Sabaghian, C. Jähne, L. Hausperger, and S. Haddadin, "The franka emika robot: A reference platform for robotics research and education," *IEEE Robotics & Automation Magazine*, vol. 29, no. 2, pp. 46–64, 2022.

[11] A. Vivas and J. M. Sabater, "Ur5 robot manipulation using matlab/simulink and ros," in *2021 IEEE International Conference on Mechatronics and Automation (ICMA)*. IEEE, 2021, pp. 338–343.

[12] P. J. Tarwadi, A. S. Arockia, R. Corrales, and A. Juan, "Manipulation and path planning for kuka (lwr/lbr 4+) robot in a simulated and real environment," *Journal of Automation, Mobile Robotics and Intelligent Systems*, pp. 15–21, 2017.

[13] K. Bekris, B. Chen, A. Ladd, E. Plaku, and L. Kavraki, "Multiple query motion planning using single query primitives. ieee/rsj int," in *Conf. Intell. Robot. Syst*, 2003, pp. 656–661.

[14] K. Gochev, A. Safonova, and M. Likhachev, "Planning with adaptive dimensionality for mobile manipulation," in *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 2944–2951.

[15] S. Klemm, J. Oberländer, A. Hermann, A. Roennau, T. Schamm, J. M. Zollner, and R. Dillmann, "Rrt-connect: Faster, asymptotically optimal motion planning," in *2015 IEEE international conference on robotics and biomimetics (ROBIO)*. IEEE, 2015, pp. 1670–1677.

[16] D. Devaurs, T. Siméon, and J. Cortés, "Enhancing the transition-based rrt to deal with complex cost spaces," in *2013 IEEE international conference on robotics and automation*. IEEE, 2013, pp. 4120–4125.

[17] D. Hsu, J.-C. Latombe, and R. Motwani, "Path planning in expansive configuration spaces," in *Proceedings of international conference on robotics and automation*, vol. 3. IEEE, 1997, pp. 2719–2726.

[18] G. Sánchez and J.-C. Latombe, "A single-query bi-directional probabilistic roadmap planner with lazy collision checking," in *Int. Symp. Robotics Research*, 2001, pp. 403–417.

[19] I. A. Sucan and L. E. Kavraki, "Kinodynamic motion planning by interior-exterior cell exploration," *Algorithmic Foundation of Robotics VIII*, vol. 57, pp. 449–464, 2009.

[20] B. Gipson, M. Moll, and L. E. Kavraki, "Resolution independent density estimation for motion planning in high-dimensional spaces," in *2013 IEEE international conference on robotics and automation*. IEEE, 2013, pp. 2437–2443.

[21] J. A. Starek, J. V. Gomez, E. Schmerling, L. Janson, L. Moreno, and M. Pavone, "An asymptotically-optimal sampling-based algorithm for bi-directional motion planning," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 2072–2078.

[22] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, A. Y. Ng, *et al.*, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2. Kobe, Japan, 2009, p. 5.

[23] I. A. Sucan and S. Chitta, "Moveit!" 2013.

[24] A. Yershova and S. M. LaValle, "Deterministic sampling methods for spheres and so (3)," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, vol. 4. IEEE, 2004, pp. 3974–3980.