2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

A MySQL Database for the Systematic Configuration Selection of Redundant Manipulators when Path Planning in Confined Spaces

Kat Styles Wood[1,2], Thomas B. Scott [3] and Antonia Tzemanaki[1]

Abstract— **Redundant manipulators offer a continuum of joint configurations which satisfy a specific end-effector pose, an advantage when operating within confined spaces. This, however, challenges a controller to select a single goal configuration from a wide range when path planning. This paper outlines the use of the MySQL database management system for systematic goal selection during redundant manipulator path planning in confined spaces. We outline a sampling method to envelope all configurations of a redundant manipulator and utilise it to generate a complete database of configurations. We demonstrate the application of this method to generate a large data-set of (1 billion) manipulator configurations for a KUKA LBR iiwa 14 equipped with a Robotiq 2F-85 gripper. With this database, the controller systematically selects goal configurations during 50 path planning scenarios within the confined space of a glovebox. We compare this to an iterative method using existing kinematic solvers to select goal configurations as a baseline. The database method achieves a 100% success rate in 42% of the scenarios attempted. In comparison, the baseline method achieves >50% success rate in just 6% of the scenarios attempted. Our proposed method also produces repeatable paths, which are similar in length and link swept area for each attempt of the same scenario, whereas the baseline method generates a different path in every attempt.**

I. INTRODUCTION

Across a number of applications and fields, gloveboxes are used to isolate material for handling in a controlled environment, minimising risk to either the human operator in the case of hazardous materials or to the sample itself where the material needs to be kept under strict environmental conditions. In many cases, it is safer and more efficient for a robot to perform the task [1], [2]. This presents a robotic challenge of performing complex motions within a confined space [3] due to the limited configurations achievable by the robot compared to operation in a more open environment.

Most often, for a robotic manipulator arm to perform a task, the required motion is determined in the task-space, by its end-effector. For example, in order to grasp an object, the task goal is defined by the desired gripper position and orientation (the gripper's pose). To achieve this pose, the controller must identify a set of manipulator joint positions (the manipulator's configuration) which satisfies the gripper pose requirements, and subsequently plan and execute a motion from the manipulator's start configuration to this goal configuration. A redundant manipulator is suitable for confined space operation due to its inherent ability to avoid singularities and satisfy required end-effector poses with a continuum of configurations [4]. However, this also creates a challenge for the controller, where a single manipulator configuration must be chosen as the goal from this wide range of possible solutions. There are three main methods of selecting a single manipulator goal configuration when handling redundancy. One method is to set a joint as 'passive' or 'free', setting its joint angle to an operator-defined value, and excluding it from the kinematic chain, solving the inverse kinematics (IK) problem as a non-redundant system using a closed form method [5], [6]. Another method is to incorporate some element of randomness or a user selected fixed parameter and iterate through various manipulator goal configurations to find one for which a path can be planned [7], [8]. While in a non-confined space these methods are appropriate, in a highly confined space, the difficulty of selecting a goal configuration to which a valid, non-colliding path may be found increases. Another option is to provide an operator-selected goal configuration to the planner as used on: learning-based planners for redundant manipulators, [9], planners for redundant dual armed robots, [10], redundant manipulators with mobile bases, [11] and redundant manipulators in constrained work environments, [12]. Similarly, while this may be sufficient in some applications, for complex tasks in confined spaces the user cannot identify and define a valid manipulator goal configuration for every motion. In such environments, the informed selection of a manipulator goal configuration is critical to planning success [13].

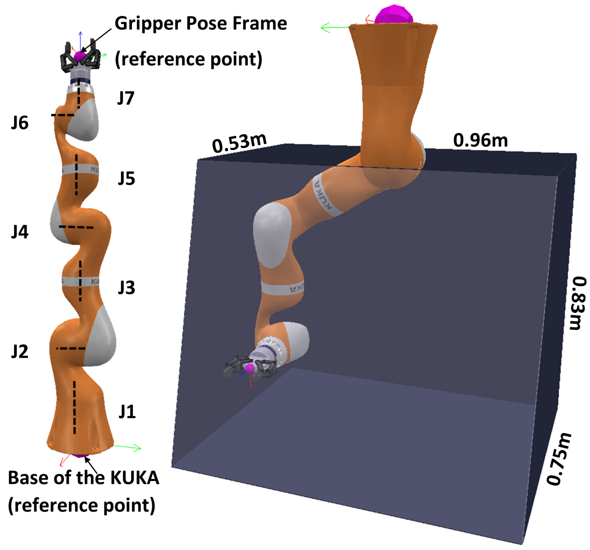

Fig. 1. Left: KUKA joint axes, gripper pose and KUKA base reference frames. Right: Experimental setup within Coppeliasim. For all frames, Z is blue, X is red, Y is green.

---

[1] Bristol Robotics Laboratory, University of Bristol, UK
[2] University of the West of England, UK
[3] Interface Analysis Centre, University of Bristol, UK
`kat.styles@bristol.ac.uk`

979-8-3503-2365-8/23/$31.00 ©2023 IEEE

Previous work has suggested intelligently selecting an appropriate manipulator goal configuration using a predefined database [14], [15]. A large number of manipulator configurations are recorded, mapping the joint angles to the corresponding end-effector poses. This information is collected and stored in advance of performing tasks. When a gripper pose is requested, the robot controller looks up a suitable manipulator configuration from this database which satisfies the requested gripper pose. Utilising this method presents two key challenges: firstly, the time and computational cost to sample the workspace and populate the database with the required number of configurations for sufficient resolution, and secondly, the time taken to search for and retrieve the correct configuration from such a large database in real time. In previous work, these challenges have been addressed based on the circumstances in which the database method was deployed. In [14], the focus was on a non-redundant manipulator and producing ‘approximate’ solutions to a grasping problem. This successfully identified configurations for a planner to move the gripper sufficiently close to the object to grasp it rather than to a precisely defined grasping pose. The database size was manageable to generate and lookup during planning because firstly non-redundant arms have inherently fewer configurations than redundant arms and secondly by requiring an approximate solution, the resolution of the database can be lower. In [15], a database method was implemented on the redundant legs of a mobile robot. In this case, the database generated prioritised only pre-determined configurations based on operator-selected parameters defined by the task and no unnecessary configurations were included.

In this paper, we explore and address these challenges for a redundant manipulator operating in scenarios where no such operational constraints on the motion can reduce the database size. While generating the database is a large computational undertaking, the benefit of doing so is that a certain end-effector/manipulator system only requires it to be generated once. The database can then be used for that robot regardless of environment. For a specific manipulator, a new database is only needed if the end-effector design changes considerably (for example a change only in length would require only a simple transformation matrix to determine the equivalent end-effector pose stored in the database). Furthermore, by using appropriate software and optimising indexing and query practices, the look-up time can be reduced. Due to its widespread use, we utilise the MySQL database management system [16].

We demonstrate this methodology using the KUKA LBR iiwa 14 7DoF (degree of freedom) redundant manipulator [17] equipped with a Robotiq2F-85 gripper [18], operating within a virtual confined space of a 0.51m^3 glovebox (Figure 1). The database of configurations is used to return appropriate manipulator goal configurations during 50 simulated path planning scenarios within the confined space (e.g., a glovebox). The simulations are performed in Coppeliasim [19]. Section II describes the sampling of the workspace and population of the database, followed by the querying strategy and a description of the experimental verification against the existing iterative method utilising kinematic solvers: BioIK [20], TRAC-IK [21] and KDL [8]. The results of the 50 planning scenarios are presented in Section III and discussed in Section IV.

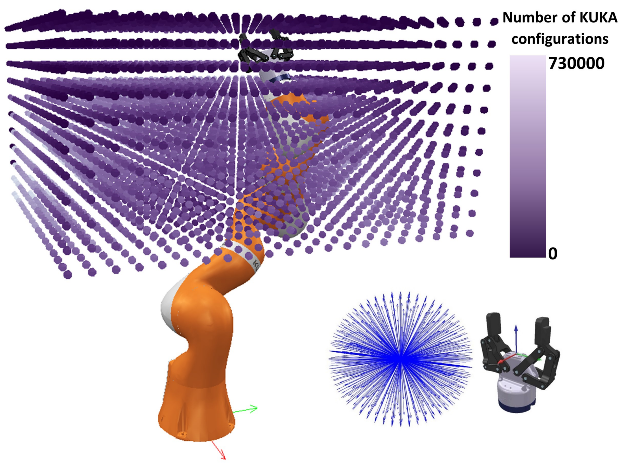

Fig. 2. Workspace sample of gripper poses by the number of KUKA configurations at each position. Lower Right: 7776 orientations of the gripper pose taken at each Cartesian position, shown in terms of the Z axis of the gripper pose frame.

The contribution of this paper is firstly, a method to discreetly sample the complete workspace of a redundant manipulator, enveloping all possible configurations and secondly, to demonstrate the feasibility of utilising this large pre-calculated data-set during path planning tasks.

Throughout this paper, the position and orientation of the grasp point of the gripper, is referred to as the ‘gripper pose’ and the complete set of joint angles of the KUKA robot is referred to as the ‘KUKA configuration’.

II. METHODOLOGY

A. *Sampling the Workspace of a Redundant Manipulator*

The workspace of the KUKA is sampled in 5cm increments in the X, Y and Z directions relative to the base of the KUKA, shown in Figure 1, giving a sample size of 3927 Cartesian points. At each coordinate point, KUKA configurations are generated which satisfy a sample of orientation requirements on the gripper. The gripper orientation is defined in Tait-Bryan angles (α, β, γ) and it is sampled at 10° increments in the α and β angles and at 60° increments in the γ angle. The γ angle is responsible for the gripper rotation about its Z axis, shown in Figure 1. This rotation is easily resolved by Joint 7 of the KUKA and, as such, sampling this orientation does not need a higher resolution. The orientation sample size is 7776 orientations at each Cartesian position, shown in Figure 2.

As the KUKA is a redundant 7DoF manipulator, there is a continuum of possible configurations which satisfy each of these gripper poses. The method for discretely sampling all of the possible configurations achievable at each gripper pose is dependent on how these configurations can vary. Its redundancy allows the ‘elbows’ of the KUKA to swing whilst holding the gripper still. This swing may be characterised by placing any joint of the KUKA in ‘passive’ mode, moving

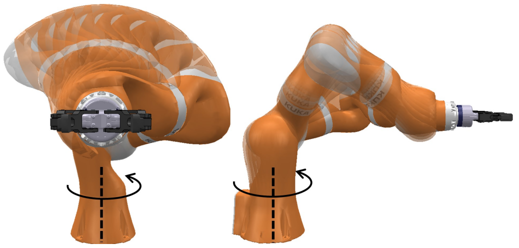

Fig. 3. KUKA elbow 'sweep' due to rotation of Joint 1, displayed in 10° increments.

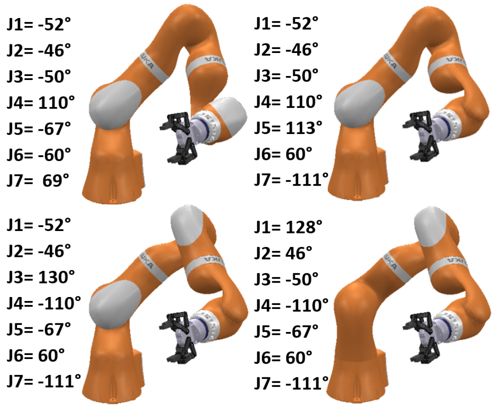

Fig. 4. KUKA configuration symmetry: 4 configurations which occupy very similar positions in the workspace. In each, the gripper is at the same pose but the KUKA has a different set of joint angles.

it through its range of motion whilst solving the IK for the other 6 joints. To sample discretely, enveloping all possible elbow swings at any given gripper pose, each of the 7 joints are set as passive in turn and iterated through their range of motion in steps of 10°. Figure 3 shows the elbow positions due to Joint 1 iteration, whilst holding the gripper in the same pose throughout.

A second parameter by which the configuration can vary is that the KUKA is a manipulator which exhibits symmetry [13]. This is where the manipulator can occupy a very similar physical position (the location of its links) in the Cartesian space for multiple joint configurations. Specifically, the joint angles across several configurations are related by a combination of ±180° on any of the 'inline' type joints (Joints 1, 3, 5, 7) and a -1 multiplier on the 'elbow' type joints (Joints 2, 4, 6). This symmetry property is shown in Figure 4. This is important, in practice, because collision validation will return for each of these configurations, such as the group shown in Figure 4, the same validity with respect to the environment (except for collisions due to the non-uniform shape of the links). However, for a path planner, the distance between them in joint space is large. This means that for any given start configuration, planning to one of these as a goal configuration will be much easier than the others. It is, therefore, important to find and record in the database the symmetrical alternatives of all configurations found.

The complete redundant manipulator sampling method is shown in Figure 5. All Cartesian data are recorded relative to the KUKA base (Figure 1). The redundant manipulator sampling method generated 1,052,865,986 unique KUKA configurations, shown in Figure 2. These configurations were generated utilising Amazon Web Services [22]. For the purpose of this work, configurations are generated only for gripper poses inside of the glovebox region. The database is openly available at the University of Bristol repository [23].

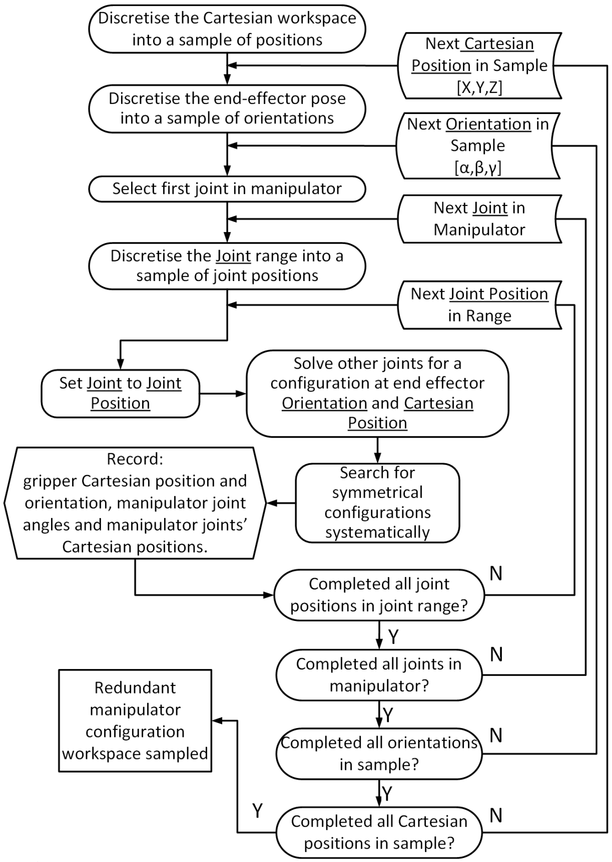

Fig. 5. The redundant manipulator configuration workspace sampling method

B. MySQL Database

The KUKA configurations are stored in a MySQL database as an InnoDB table. The data are indexed by Cartesian region relative to the KUKA base. Searching the database by gripper pose will return all KUKA configurations which satisfy that gripper pose, to within the resolution of the sampled workspace.

To plan a path from the start KUKA configuration to the gripper goal pose, the database is queried to provide a configuration which satisfies the gripper goal pose. The query inputs are the gripper goal pose, the location of obstacles in the environment (e.g. the walls of the glovebox) and the start configuration. As the Cartesian positions of the joints are stored for each configuration in the database, we remove from consideration any solutions which are in collision with the environment, without performing computationally costly collision checking. We then order the remaining configurations to determine which will be most 'optimal' for the planning task. In this work, we define the most 'optimal' configuration as the nearest to the start configuration in joint space. This is to favour a shorter joint path and more efficient movement of the joints in terms of the swept area of the central axis of the KUKA links [24].

C. Path Planning Experiment

To validate the resolution of the redundant manipulator workspace sampling method and determine the suitability of

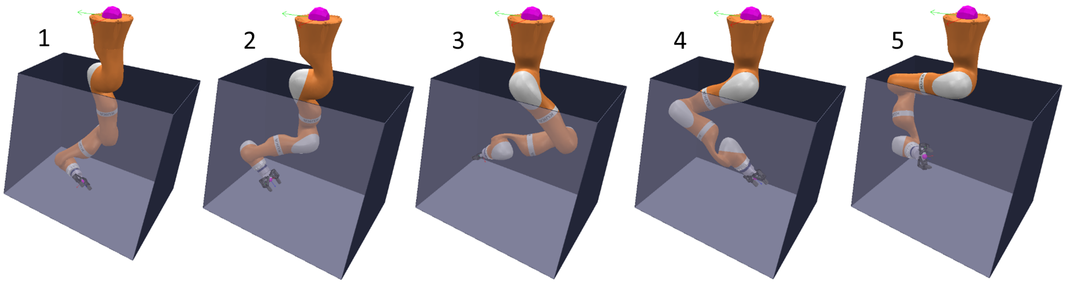

Fig. 6. 5 randomly selected starting KUKA configurations, configuration ID 1-5

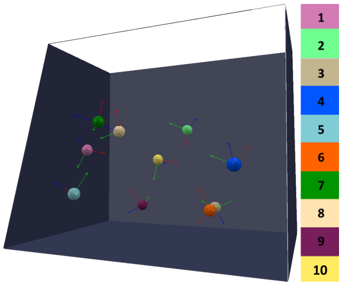

Fig. 7. 10 randomly selected gripper goal poses, gripper goal ID 1-10. Target pose of the gripper reference frame, Figure 1.

the data-set generated for goal selection in path planning, we have investigated 50 planning scenarios. These scenarios comprise planning a path, with 5 randomly selected start KUKA configurations and 10 randomly selected gripper goal poses within the glovebox. These are labeled in Figures 6 and 7. Each planning scenario is defined by a start KUKA configuration and a gripper goal pose within the glovebox. A goal KUKA configuration is then selected and passed to the OMPL path planner KPIECE1 [25], which attempts to plan a path from the start configuration to the goal configuration. The maximum planning time is 4s, after which the planner is considered to have 'failed'. We compare the systematic Database Lookup Method (DLM) to an iterative method using each of the following known kinematic solvers: TRAC-IK, BioIK and KDL. The systematic and iterative methods of goal configuration selection are assessed by the resulting path planner performance. BioIK and TRAC-IK were set to preference a minimal distance to the known starting configuration when returning a solution.

1) Database Lookup Method (DLM): The DLM searches the database for all configurations which satisfy the requested gripper goal pose. Since the database may not contain the exact gripper pose requested, the configurations returned by the database will give a gripper pose to within the position and orientation resolution of the data-set; in this work the resolution of the position is 5cm and the orientation is 10° as specified in Section II-A. We can then sort the returned configurations by a search criteria to intelligently select the 'optimal' configuration for the planning task. In this work, we sort the returned configurations by distance to the starting KUKA configuration in joint space and return the nearest 10 configurations. Using the nearest configuration as a start point, a configuration satisfying the nearby gripper goal pose can easily be found. This is achieved with the Coppelia Kinematics Routines (CKR) 'Generate IK Path' function which is used to calculate a straight line between the returned configuration's gripper pose (near to the goal) and the gripper goal pose. The resulting configuration is returned as the KUKA goal configuration. Once the goal configuration is found, the path planner attempts to find a path to it from the start configuration. If it is unsuccessful, the next configuration in the 10 returned by the database query is attempted. If a path cannot be found after 10 goal configurations are attempted, this scenario attempt is unsuccessful. Each scenario is repeated 100 times. As the search is systematic, the same 10 configurations are returned from the database for every repeat of the same scenario.

2) Iterative Search Method: The iterative search method finds a goal by iterating over solutions generated by an existing kinematic solver (this method has been applied using the solvers BioIK, TRAC-IK and KDL). For each scenario, the iterative method receives a goal configuration which satisfies the requested gripper goal pose from the kinematic solver and then checks for collision with the environment. If the goal configuration collides with the environment, the kinematic solver is rerun for another goal configuration. This iteration is performed up to 10 times. Once a collision free goal configuration is generated, the path planner attempts to find a path to it from the start configuration. If a path cannot be found, or if no collision free goal configuration can be found, this scenario attempt is unsuccessful. Each scenario is attempted 100 times.

III. RESULTS

The results of the path planning experiments are summarised in Table I. The number of successful attempts (out of 100 repeats) of each planning scenario are presented in Figure 8. The paths generated in these successful attempts for each goal selection method are compared by the link swept area, Figure 9, and the end-effector path length, Figure 10. The link swept area is the distance traveled by the links of the robot arm in terms of the total surface swept by the central axis of each link combined, described in [24]. The end-effector path length shows the distance traveled by the gripper reference point, Figure 1, for each path. The attempt time is the average time taken to identify the goal and find a collision free path to that goal. The attempt time, excluding the time taken to iterate through failures, is presented in Figure 11. The attempt time, including the time taken to iterate through failures, is presented in Figure 12.

The DLM demonstrated good performance compared to the iterative method in both the success rate and the repeatability of the paths found. For planning scenarios where the DLM was successful, it was successful in all 100 attempts, with the exception of scenarios 1-1 and 1-7. In scenario 1-1, the DLM succeeded in 99 of the 100 attempts. In

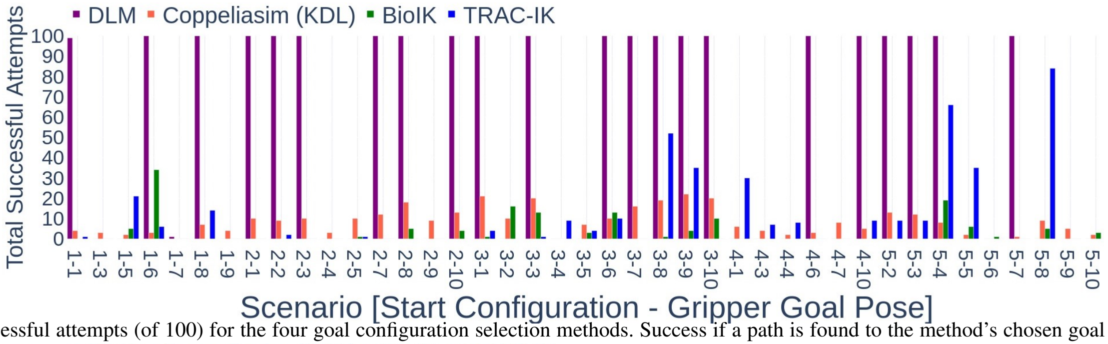

Fig. 8. Successful attempts (of 100) for the four goal configuration selection methods. Success if a path is found to the method's chosen goal configuration.

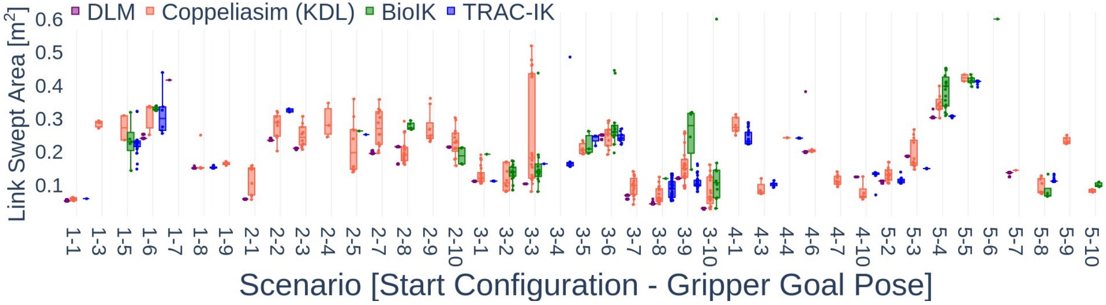

Fig. 9. Link central axis swept area for all successful attempts in each scenario. Lower means the manipulator links travel a shorter distance over the path. Smaller spread means the path swept area is similar for every attempt. The number of data points in each scenario is shown in Figure 8.

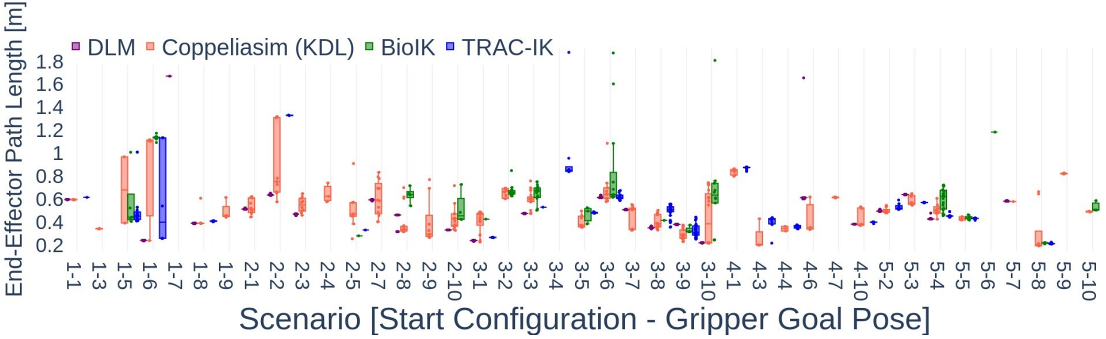

Fig. 10. Gripper reference point, Fig. 1, path distance for all successful attempts in each scenario. Lower means the gripper travels a shorter distance over the path. Smaller spread means the path length is similar for every attempt. The number of data points in each scenario is shown in Figure 8.

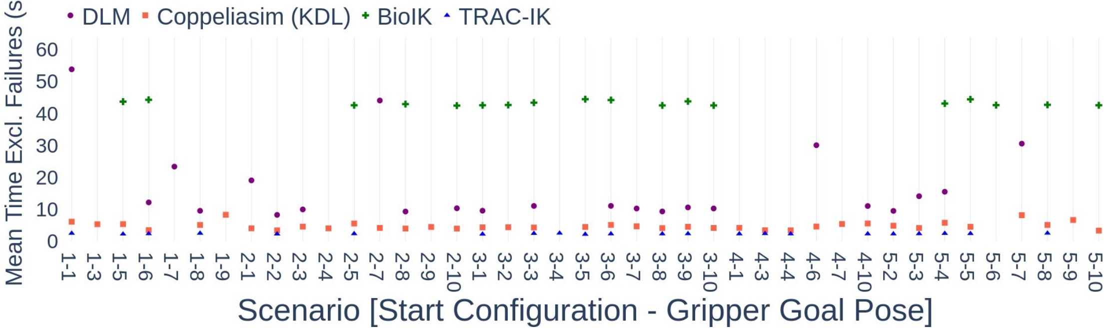

Fig. 11. Average time taken to select a goal and find a path for each successful scenario - excluding time taken during failed attempts

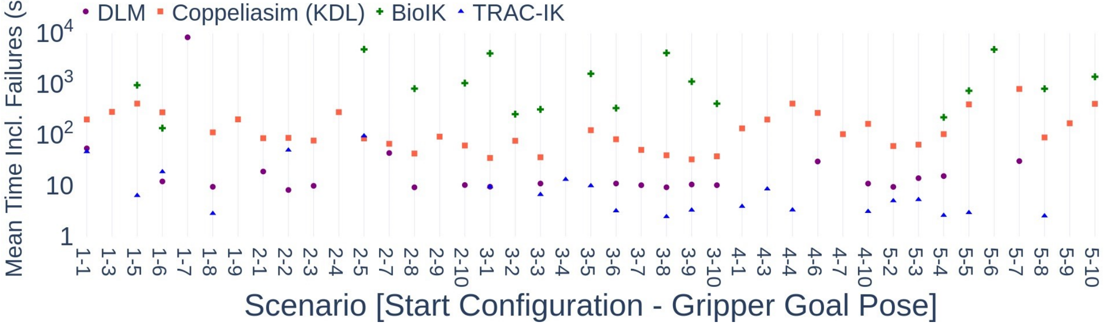

Fig. 12. Average time taken to select a goal and find a path for each successful scenario - including time taken during failed attempts

TABLE I

PLANNER GOAL CONFIGURATION SELECTION METHOD PERFORMANCE

Success Rate	Number of Scenarios (of 50)			
	DLM	KDL	BioIK	TRAC-IK
90-100%	22	0	0	0
50-90%	0	0	0	3
25-50%	0	0	1	3
20-25%	0	2	0	1
15-20%	0	5	2	0
10-15%	0	4	2	1
5-10%	0	13	2	8
<5%	1	14	11	6
0% (Failed)	27	12	32	28

scenario 1-7, the path planner was successful in only 1 of the 100 attempts when using the DLM. This is due to the variability in the performance of the path planner. As the DLM is systematic in searching for a goal configuration, the same goal configurations were found by the DLM for every attempt of the same scenario. This means that the same planning task (start configuration to goal configuration) was attempted by the planner in every attempt of the same scenario. Differences in success between each attempt with the DLM is due to the performance of the planner (KPIECE1) itself. In comparison, KDL succeeded in 4 of the 100 attempts in scenario 1-1, BioIK succeeded in 0 attempts and TRAC-IK succeeded in 1 attempt. None of the iterative goal selection methods succeeded in any attempts of scenario 1-7. The paths generated using the DLM in each successful planning scenario were highly repeatable as can be seen by the small spread shown in Figures 9 and 10. The largest standard deviation in link swept area, Figure 9, when using the DLM is 0.018m^2 in scenario 4-6 and the average across the scenarios is 0.003m^2. With KDL, the largest standard deviation is 0.159m^2 in scenario 3-3 and the average is 0.031m^2. With BioIK, the largest standard deviation is 0.155m^2 in scenario 3-10 and the average is 0.034m^2. With TRAC-IK, the largest standard deviation is 0.102m^2 in scenario 3-4 and the average is 0.016m^2. The largest standard deviation in end effector path length, Figure 10, when using the DLM is 0.104m in scenario 4-6 and the average across the scenarios is 0.008m. With KDL, the largest standard deviation is 0.411m in scenario 1-6 and the average is 0.093m. With BioIK, the largest standard deviation is 0.410m in scenario 3-10 and the average is 0.088m. With TRAC-IK, the largest standard deviation is 0.393m in scenario 1-6 and the average is 0.053m. Figures 11 - 12 show the time taken to complete each successful attempt excluding and including the time taken during failures respectively. Although the time taken during the iterative successful attempts is lower than that using the DLM in Figure 11, the lower success rate means they would need to iterate many times before finding a path in any given planning scenario. Figure 12 shows that for successful attempts, when accounting for failures, the DLM is faster than the iterative method when KDL or BioIK is used to select the goal in all scenarios. In 10 scenarios, the iterative method using TRAC-IK to select the goal is faster than the DLM.

The success of the DLM in any planning scenario, where there exists a solution, is determined by the database resolution and the DLM search criteria. In this work, we use the search criteria of ‘most similar to the start configuration’ however, depending on the planning task and environment set up, this will not always return the most optimal goal configuration in the database. Further work shall explore a dynamic goal configuration selection method dependant on the planning task parameters.

IV. CONCLUSION

In this work, we have presented a method for generating an enveloping, discretely sampled data-set of manipulator configurations. This redundant manipulator workspace sampling method, outlined in Figure 5, has been shown to generate a database of sufficient resolution to utilise in goal configuration selection during path planning. As such, this sampling method could readily be applied to other redundant manipulators and setups for use in confined or cluttered environments where an operator-selected goal configuration is not feasible. Although the computational effort required to generate a configuration database for a given redundant manipulator is high, once generated, it is valid to use for that manipulator operating in any environment.

This work shows the advantage in reliability and repeatability of using a pre-calculated data-set to systematically select the planning task goal configuration over iterative search methods. This is particularly important in confined space operations, where the goal configuration selected strongly impacts path planner success. Repeatability is also important in many applications involving hazardous materials or remote environments where the robot operates in locations unreachable to humans. In such cases, demonstrating repeatability of generated paths could aid the operator in trusting the robot to reliably reach requested goals.

When querying the database for a configuration to satisfy a required end-effector pose, a discrete sample is returned which envelopes the large continuum of solutions that exist. This sample can then be sorted by preference to return a configuration which has been intelligently selected for a given planning task. In this work, the criterion of ‘nearest in joint space to the starting configuration’ has been used, as described in Section II-B. Although this is successful in 46% of tested scenarios, a single criterion by which to select the goal cannot be successful in all planning tasks and environments. Ideally, the selection criteria should be dynamic and dependant on the known planning and environment parameters. An ideal approach would allow for a dynamic selection criteria based on the starting configuration, the environment geometry in relation to the manipulator and the gripper goal pose. Further work shall investigate the selection of database search criteria dependant on these planning task elements.

ACKNOWLEDGEMENT

This work was supported by EPSRC (EP/S021795/1) and by the Robotics and AI in Nuclear (RAIN) research hub (EP/R026084/1). For the purpose of open access, the authors have applied a Creative Commons Attribution (CC BY) license to any accepted manuscript version arising.

REFERENCES

[1] M. Talha et al., ‘Towards robotic decommissioning of legacy nuclear plant: Results of human-factors experiments with tele-robotic manipulation, and a discussion of challenges and approaches for decommissioning,’ 2016 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR), 2016, pp. 166-173, doi: 10.1109/SSRR.2016.7784294.

[2] O. Tokatli et al., ‘Robot-Assisted Glovebox Teleoperation for Nuclear Industry’, Robotics, vol. 10, no. 3, Art. no. 3, Sep. 2021, doi: 10.3390/robotics10030085.

[3] A. Sharp, M. W. Hom and M. Pryor, ‘Operator training for preferred manipulator trajectories in a glovebox,’ 2017 IEEE Workshop on Advanced Robotics and its Social Impacts (ARSO), 2017, pp. 1-6, doi: 10.1109/ARSO.2017.8025193.

[4] D. Baker, C. Wampler. (1988). ‘On the Inverse Kinematics of Redundant Manipulators.’ I. J. Robotic Res.. 7. 3-21. doi: 10.1177/027836498800700201.

[5] I. Zaplana and L. Basanez, ‘A novel closed-form solution for the inverse kinematics of redundant manipulators through workspace analysis’, Mechanism and Machine Theory, vol. 121, pp. 829–843, Mar. 2018, doi: 10.1016/j.mechmachtheory.2017.12.005.

[6] R. Diankov. 2010. ‘Automated construction of robotic manipulation programs.’ Ph.D. Dissertation. Carnegie Mellon University, USA. Advisor(s) Takeo Kanade and James Kuffner.

[7] X. Tian, Q. Xu, and Q. Zhan, ‘An analytical inverse kinematics solution with joint limits avoidance of 7-DOF anthropomorphic manipulators without offset’, Journal of the Franklin Institute, vol. 358, no. 2, pp. 1252–1272, Jan. 2021, doi: 10.1016/j.jfranklin.2020.11.020.

[8] Bruyninckx, H, ‘Open robot control software: the OROCOS project’, Proceedings - IEEE International Conference on Robotics and Automation, Mar. 2001, pp.2523 - 2528 vol.3 doi:10.1109/ROBOT.2001.933002.

[9] A. H. Qureshi, Y. Miao, A. Simeonov, and M. C. Yip, ‘Motion Planning Networks: Bridging the Gap Between Learning-Based and Classical Motion Planners’, IEEE Transactions on Robotics, vol. 37, no. 1, pp. 48–66, Feb. 2021, doi: 10.1109/TRO.2020.3006716.

[10] J. Wang, S. Liu, B. Zhang, and C. Yu, ‘Inverse kinematics-based motion planning for dual-arm robot with orientation constraints’, International Journal of Advanced Robotic Systems, vol. 16, no. 2, p. 1729881419836858, Mar. 2019, doi: 10.1177/1729881419836858.

[11] J. Liao, F. Huang, Z. Chen, and B. Yao, ‘Optimization-based motion planning of mobile manipulator with high degree of kinematic redundancy’, Int J Intell Robot Appl, vol. 3, no. 2, pp. 115–130, Jun. 2019, doi: 10.1007/s41315-019-00090-7.

[12] P. Rajendran, S. Thakar, and S. K. Gupta, ‘User-Guided Path Planning for Redundant Manipulators in Highly Constrained Work Environments’, in 2019 IEEE 15th International Conference on Automation Science and Engineering (CASE), Aug. 2019, pp. 1212–1217. doi: 10.1109/COASE.2019.8843126.

[13] R. Seyboldt, C. Frese, and A. Zube, ‘Sampling-based Path Planning to Cartesian Goal Positions for a Mobile Manipulator Exploiting Kinematic Redundancy’, in Proceedings of ISR 2016: 47st International Symposium on Robotics, Jun. 2016, pp. 1–9.

[14] A. D.Henning, 2014. ‘Approximate inverse kinematics using a database’. Technical report, Worcester Polytechnic Institute.

[15] B. W. Satzinger, J. I. Reid, M. Bajracharya, P. Hebert, and K. Byl, ‘More solutions means more problems: Resolving kinematic redundancy in robot locomotion on complex terrain’, in 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, USA, Sep. 2014, pp. 4861–4867. doi: 10.1109/IROS.2014.6943253.

[16] M. Widenius, D. Axmark, and P. DuBois, ‘Mysql Reference Manual’, Jan. 2002.

[17] KUKA Roboter GmbH, ‘LBR iiwa LBR iiwa 7 R800, LBR iiwa 14 R820 Specification’. [Online]. Available: http://www.oir.caltech.edu/twiki_oir/pub/Palomar/ZTF/KUKARoboticArmMaterial/Spez_LBR_iiwa_en.pdf. [Accessed:19 March 2021].

[18] Robotiq. 2022. 2F-85 and 2F-140 Grippers - Robotiq. [online] Available at: https://robotiq.com/products/2f85-140-adaptive-robot-gripper [Accessed 24 February 2022].

[19] E. Rohmer, S. P. N. Singh, and M. Freese, ‘CoppeliaSim (formerly V-REP): a Versatile and Scalable Robot Simulation Framework’

[20] S. Starke, N. Hendrich and J. Zhang, ‘Memetic Evolution for Generic Full-Body Inverse Kinematics in Robotics and Animation’, in IEEE Transactions on Evolutionary Computation, vol. 23, no. 3, pp. 406-420, June 2019, doi: 10.1109/TEVC.2018.2867601.

[21] P. Beeson and B. Ames, ‘TRAC-IK: An open-source library for improved solving of generic inverse kinematics’, 2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids), Seoul, Korea (South), 2015, pp. 928-935, doi: 10.1109/HUMANOIDS.2015.7363472.

[22] Amazon Web Services, Inc. 2022. Cloud Services - Amazon Web Services (AWS). [online] Available at: https://aws.amazon.com [Accessed 28 February 2022].

[23] Tzemanaki, A., Styles, K. (2023). ‘KUKA LBR iiwa configuration database’ doi: 10.5523/bris.1j7nw1aagdzw62w5t6l90hvf82

[24] Styles Wood, K. L., Scott , T. B., and Tzemanaki, A. ‘End-Effector Mobility for Manipulators in Confined Spaces’, in UKRAS21 Conference: “Robotics at home” Proceedings, 2nd June 2021 (pp. 37-38) doi: 10.31256/Ut2Qg2T

[25] 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, Dec. 2012, doi: 10.1109/MRA.2012.2205651.