Journal of
Engineering Sciences
Faculty of Engineering
Assiut University
مجلة
العلوم الهندسية
كلية الهندسة
جامعة أسيوط
journal homepage: http://jesaun.journals.ekb.eg

Kinematic Analysis and Simulation of an Industrial Rail-Mounted Robot Manipulator Using Ruckig for Enhanced Path Planning

Received 20 17 February 2024; Revised 30 April 2024; Accepted 30 April 2024

Abrar Mahrous [1]
Mahmoud Abdelrahim [2]
Ahmed S Ali [3]
Shuaiby Mohamed [4]

Keywords
Robotic manipulator, Seventh axis, Coppeliasim, Kinematics Analysis, Online Trajectory generation, Ruckig.

Abstract: Robotic manipulators are being widely used in industrial operations and healthcare due to their versatile functionalities. However, the confined workspace of fixed robotic arms limits their applicability in scenarios requiring a broader range of configurations. To overcome this limitation, this research provides a case study on a robotic system composed of two primary subsystems an articulated robotic arm and a linear rail. A simple path planning task was carried out using CoppeliaSim simulation software to study the effect of Ruckig, an advanced online trajectory generation algorithm, alongside the RRT-Connect path planning algorithm. This study demonstrates the capacity of Ruckig to improve the efficiency of path planning regarding the process time and path length. The results showed that Ruckig helped reducing the process time by 90% with an exceptional improvement to the motion profiles of the system. Regarding the path length, it seems that it was able to decrease the length in certain cases, but not all. The reduction reached a maximum of 50% compared to the original length.

1. Introduction

Robotic manipulators, with their human-like arm structure, have become integral in numerous sectors, including space exploration, industry, social applications, and healthcare [1]. However, their fixed configurations often result in severely limited workspaces, restricting their suitability for certain applications where greater maneuverability is required

[1] Department of Mechanical Engineering, Assiut University, Assiut, Egypt
[2] Department of Mechanical Engineering, Assiut University, Assiut, Egypt
[3] Department of Mechanical Engineering, Assiut University, Assiut, Egypt
[4] Department of Mechanical Engineering, Assiut University, Assiut, Egypt

https://doi.org/10.21608/JESAUN.2024.268475.1312

[2]. To overcome this limitation, locomotion-based solutions have emerged, where robotic arms are mounted on mobile platforms like automated guided vehicles or linear trains. By adopting this approach, the manipulator's workspace can be significantly expanded, enabling operations in fields such as space and underwater research that were previously challenging [3]. Moreover, the utilization of a seven-axis robot, i.e., a robot manipulator mounted on a linear rail, offers distinct advantages. It enhances tooling positioning for complex geometries, provides increased mobility to maneuver around parts, and ensures tooling stability through elbow bending without necessitating end-effector movement [4]. Additionally, employing linear robots can reduce the need for peripheral equipment, resulting in cost savings in system integration [5].

1.1 Related Works

Over the past few years, numerous researchers have made significant contributions to resolving the inverse kinematics problem of robotic systems and determining the ideal parameters for achieving a minimum jerk trajectory. The methods of solving the inverse kinematics are usually categorized as closed-form methods and numerical methods. The closed form method helps find all the possible exact solutions for a given end-effector's pose either by algebraic methods which is very efficient but faces problems and not applicable to robots with complex kinematic structures [6], [7], geometric strategies which considered to be Intuitive and easy to understand, but has a very slow convergence, especially for high-dimensional problems and might not yield to optimal solutions [8] or machine learning techniques which still an active field of research with very promising outcomes the only problem is that it requires significant training data [9], [10], [11].

Another significant issue is the motion planning challenge, which has been the focus of numerous academics aiming to develop motion planners that minimize planning time, enhance motion performance, and generate seamless and improved motion profiles. in [12] the performance of the ABB IRB140 industrial robot has improved by using a velocity planning model developed by artificial intelligence in the simulation system. Using a novel methodology [13] provides optimal measurement configurations adapted to the experimental setup for the KUKA KR500 MT robot mounted on a rail. in [4] The numerical validation of the motion planner's effectiveness and the consequent control laws based on acceleration are conducted using the Runge-Kutta Method. Computer simulations are employed to show these findings for a robot with n-links positioned on a slider. in [5], [14] the time optimal trajectory planning for a redundant manipulator with a 6-axis robot manipulator mounted on a rail as a case study was discussed. In [15] a time optimal solution based on iterative optimization has been evaluated with promising outcomes where it manages to reduce the average absolute values of the position error and the robot's jerk by a huge factor compared with [16]. In [17] another algorithm is presented to ensure motion smoothness which has also taken the robot jerk constrained into account.

1.2 Motivations and Contributions

The research provided examines a case study involving the utilization of a robotic arm affixed to a linear rail or mobile slider. This scenario has not been extensively explored in previous research, which predominantly concentrated on serial manipulators with solely revolute joints. Integrating a rail into a robotic manipulator system greatly increases the workspace of the robotic arm by allowing it to smoothly move along the rail towards difficult targets. However, the additional degree of freedom provided by the rail can make the robot redundant for certain tasks and complicate the calculation of its inverse kinematics. For that reason, the closed-form solution has been found by selecting the prismatic joint as a redundant joint and then solving the inverse kinematics using Pieper's method [18]. The outcomes have been simulated and tested using a robot simulation software named CoppeliaSim which offers extensive language support, physics engines, and versatile model types [19]. Ultimately, this study discusses the outcomes derived from the application of the Ruckig online trajectory generation algorithm, integrated with the RRT-Connect path planning algorithm. This analysis highlights the efficacy of combining these advanced algorithms in optimizing path planning and trajectory execution for the rail-mounted robotic manipulator. In summary the contribution of this study can be stated as follows:

- Noval Case Study: This work explores a novel scenario of mounting a robotic manipulator on a linear rail, expanding beyond traditional fixed-arm manipulators offering a solution for applications requiring enhanced workspace and maneuverability.
- Algorithm Integration: It evaluates the combination of the Ruckig online trajectory generation algorithm with the RRT-Connect path planning algorithm, aiming to optimize path planning for the rail-mounted robotic system.
- Simulation-Based Analysis: Through CoppeliaSim simulations, it quantitatively compares the performance of different trajectory generation algorithms, demonstrating the efficacy of the Ruckig algorithm in reducing processing time and path length.
- Robust Methodology: The study presents a rigorous methodology for modeling and simulating the rail-mounted robot system, ensuring accuracy and stability in the simulation results.

The structure of the paper is as follows: The kinematic analysis of the system is covered in Section 2. In Section 3, the implementation strategies that were used to prepare the system for study in CoppeliaSim are discussed. After completing a path planning assignment, the simulation results are displayed in Section 4, and the paper's conclusion is provided in Section 5.

1.3 Problem Definition

In an n-DoF robotic system, the configuration of the system is denoted as q where $q \in R_n$ and contains the set of joints' positions for the joints 1 to n.

$$q = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ \vdots \\ q_n \end{bmatrix}$$

While the state of the system is represented as $\tilde{x}$ where $\tilde{x}$ denotes the position of each configuration and its derivatives the velocity, acceleration, and jerk.

$$\tilde{x}(t) = [q(t);\ \dot{q}(t);\ddot{q}(t);\ \dddot{q}(t)\ ]$$

Given an initial state $\tilde{x}_0$ and a target (final) state $\tilde{x}_f$, The problem is to find the effect of the online trajectory generation algorithm named Ruckig on a robotic system with 6-DoF satisfying the velocity, acceleration, and jerk constraints. For all times $t \in [0, T_f]$ where T_f represents the total path time.

$$x_{max} \leq x(t) \leq x_{min}$$

1.4 System Modelling

The inverse kinematics of serial manipulators could be defined as the relationship between the robot's joint angles and the desired position of its end-effector in the robot's Cartesian space. The robotic system under consideration is configured as PRRRRR, representing a series of one prismatic joint and five revolute joints. Figure 1 illustrates the open chain of the system, emphasizing its essential parameters.

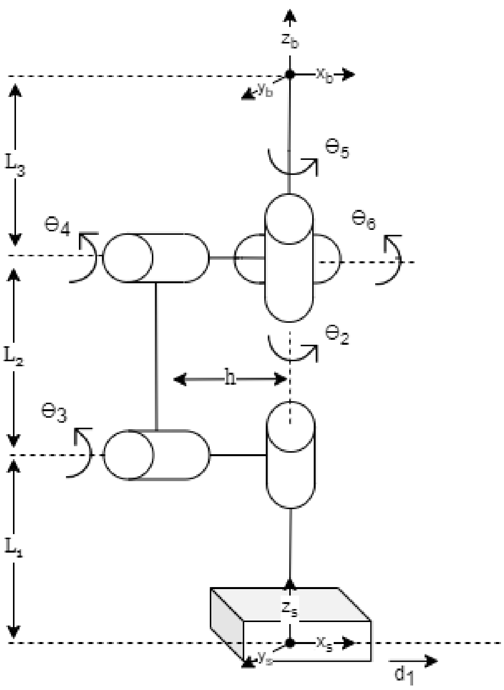
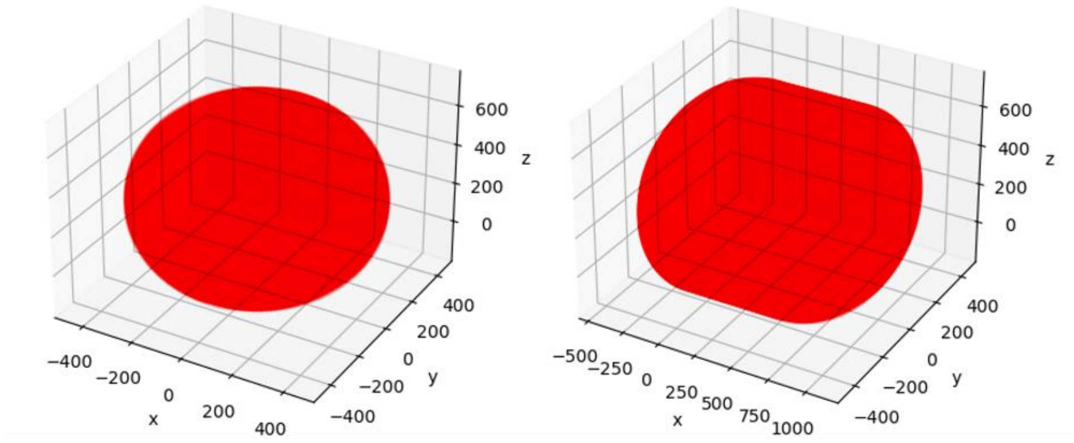


Fig. 1 A PRRRR spatial open chain in its zero configuration. L_1 = 0.0718m, L_2 = 0.025m, L_3 = 0.1133m, d_1 = 0:800mm, θ_2 = 0°:360°, θ_3 = -80°: 20°, θ_4 = -180°: 180°, θ_5 = 0°: 360°, θ_6 = 0°: 360°

Since the kinematic model of the serial manipulator gets more complex when the number of degrees of freedom increases and it becomes more difficult to find a closed-form solution to its inverse kinematic problem [2], [20], the solution proposed is to divide the system under consideration into two subsystems, the linear rail, and the robotic arm. The linear rail serves as the base for the entire system. It provides horizontal motion for the robotic arm along the x-axis and has a predetermined length, denoted as l. This horizontal motion of the linear rail does not alter the geometry of the workspace of the mounted robotic arm but rather results in a translation of the volume of the created workspace (refer to Figure 2). Therefore, it could be considered a redundant degree of freedom if the desired end-effector position is located in the robotic arm workspace and could be achieved using the robotic arm only.



(a) (b)

Figure 2 (a) Workspace of 5R robot manipulator. (b) Workspace of P5R robot manipulator

Figure 3 shows two distinct configurations of the integrated system. The first configuration represents the initial or home position, while the second configuration illustrates the system after the robot has successfully moved to reach the target position denoted as P (PX, p_y, p_z).

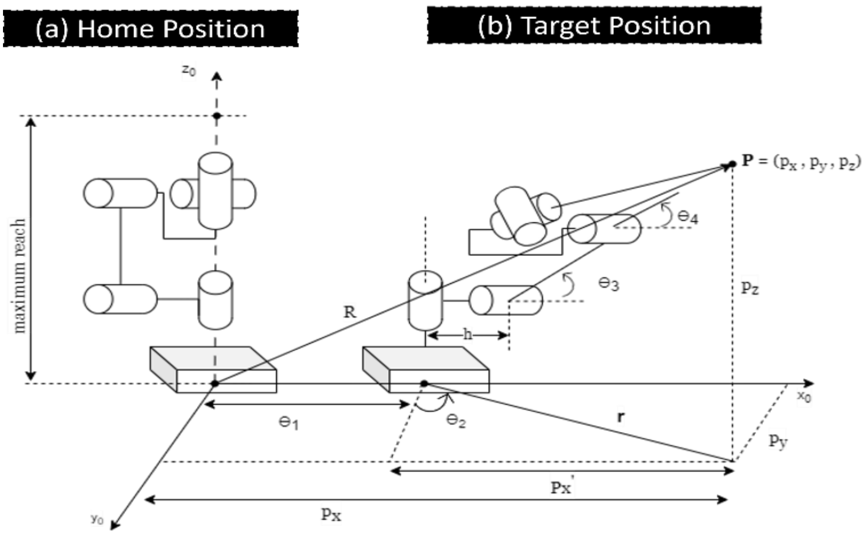


Fig. 3 Illustration of the Integrated System - 5DOF Robot Arm Mounted on linear rail (a) Home Position: The system is depicted in its initial configuration, where the robot is positioned at its starting point. (b) Target Position: The system showcases the robot.

To determine whether the first joint, i.e. the linear rail, is a redundant joint or not, a reachability test is conducted. with the subsequent equations:

$$R = \sqrt{p_x^2 + p_y^2 + p_z^2} \quad (1)$$

$$L_{max} = L_1 + L_2 + L_3 \quad (2)$$

Where R represents the distance between the target position (P) and the origin of the robot arm, and p_x, p_y, and p_z represent the corresponding Cartesian coordinates of the point (P). L_{max} represents the maximum reach of the arm and L_1 , L_2 and L_3 represents links' lengths as shown in figure 1.
Here, two cases are considered:
Case 1. If $R < L_{max}$ and the target position not a singular position, then: θ1 = 0 and the Inverse kinematics will be calculated only for the robot manipulator.
Case 2. if $R >= L_{max}$ then θ1 could be calculated as follow:

$$\theta_1 = R - L_{max} \quad (3)$$

The second subsystem consists of a 5-degree-of-freedom (5DOF) robotic arm. This robotic arm functions in the XYZ plane, enabling it to perform given tasks within a three-dimensional confined workspace. The 5DOF robotic arm consists of a set of connected links, each defined by its length li and angular position θ_i relative to the $i-1$ axis. The coordinates (p_x, p_y, p_z) indicate the position of the end-effector of the robot system at a specific time t. To obtain the forward kinematics, we utilize the four link parameters from the DH table (refer to table 1).

Table 1: DH parameter of the rail mounted robot arm.

i	a_{i-1}	α_{i-1}	d_i	θ_i
1	0	$\pi/2$	d_1	0
2	0	$\pi/2$	0	θ_1
3	0	$\pi/2$	h	θ_2
4	L_1	0	0	θ_3
5	0	$-\pi/2$	L_2	θ_4
6	0	0	*0*	θ_5

The DH parameters help us determine the transformations between each pair of frames, denoted as {i-1} and {i}, using equation (4). By multiplying these transformations, we can find the overall transformation that connects frame {5} to frame {0}. Thus, the ${}^1_5T$ will be as in (5):

$${}^{i-1}_{i}T = Screw_x(a_{i-1}, \alpha_{i-1}) Screw_z(d_i, \theta_i) \quad (4)$$

$${}^1_5T = {}^1_2T\,{}^2_3T\,{}^3_4T\,{}^4_5T \quad (5)$$

The inverse kinematics problem will then be solved geometrically. Since there's a shoulder offset with value of h the joint angle θ_2 will have 2 solutions see equations (6) and (7)

$$\theta_2 = atan2(p_y, p_x') + atan2(-\sqrt{px'^2 + py^2 - h^2}, h) \quad 6)$$

$$\theta_2 = \pi + atan2(py, px') + atan2(-\sqrt{px'^2 + py^2 - h^2}, h) \quad 7)$$

$$\cos\theta_4 = \frac{px'^2 + py^2 + pz^2 - h^2 - L_2{}^2 - L_3{}^2}{2.L_2.L_3} = D \quad 8)$$

$$\theta_4 = atan2\left(\pm\sqrt{1 - D^2}, D\right) \quad 9)$$

$$\theta_3 = atan2(pz, \sqrt{px'^2 + py^2 - h^2}) - atan2(L_3\sin\theta_4, L_2 + L_3\cos\theta_4) \quad 10)$$

2. System Setup in CoppeliaSim

2.1 The *Linear Rail*

To have a clean model in Coppeliasim and achieve fast and stable simulation, it is essential to avoid complexity and eliminate any unnecessary elements. This can be accomplished by:

1. Representing bolts or screws as cylindrical bodies.
2. Saving sub-assemblies as part assemblies in Solidworks before exporting them to CoppeliaSim in the STL mesh file format, which has demonstrated significant simplification and stability in the model.

Here are some of the main parts or assembly parts that have been used in the simulation model:

1. The aluminum base with both guides excluding the four linear bearings and the two supports of the ball screw (BF1 and BK1)
2. The motor, the pulleys, and the belt.
3. Each linear bearing has been treated as a single part.
4. Finally, the worktable with the ball screw nut and the nut housing has all been treated as a combined part.

All these parts are exported to Coppeliasim as STL files, and the final hierarchy of the model is established as shown in Figure 5. The aluminum structure (1) is considered as the base link, under which there are five distinct elements. Each linear bearing is treated as a separate link and connected with prismatic joint to the worktable using dummies as Coppeliasim only supports Serial Structures, as for the motor (2) with the drive train part is the fifth link. This link serves as the parent for the ball screw with a revolute joint connection. then the ball screw is parent to the worktable with a prismatic joint connection.

Given that the ball screw mechanism is considered a universal joint with two degrees of freedom (one revolute and one prismatic), a script is created for the prismatic joint between the ball screw and the ball nut to establish a connection between it and the revolute joint connected from the motor to the ball screw.

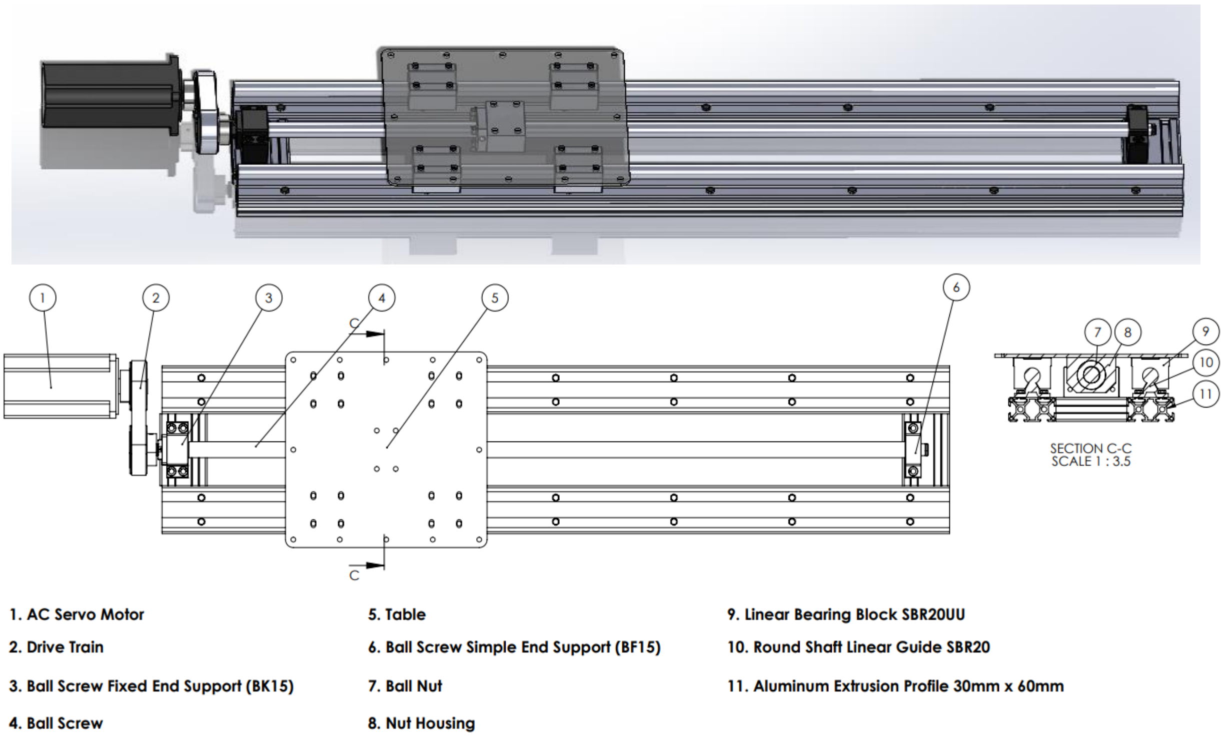


Fig. 4 A detailed drawing of the linear slider with a section showing the connection of the worktable to the ball screw.

2.2 Robot Manipulator

The robot arm is exported to CoppeliaSim with the same considerations that were taken when importing the linear slider as discussed in Section 3.1. The arm has 5 revolute joints each joint has its own assembly consisting of the joint body the motor and the gearbox the assembly of each joint has been saved as a solid part and then assembled into one assembly that with the use of the SW2URDF plugin in SolidWorks created a URDF file for the full-arm and then it was imported into CoppeliaSim. Before combining both the linear slider and the robot arm into one model the base link of the robot arm has its scene proprieties changed to a model so that the whole arm is then attached to the worktable of the linear slider using a force sensor. A complete and exploded view of the robot arm is shown in figure 4 and the corresponding scene hierarchy of the robot arm with its attachment to the worktable is dissipated in figure 6.

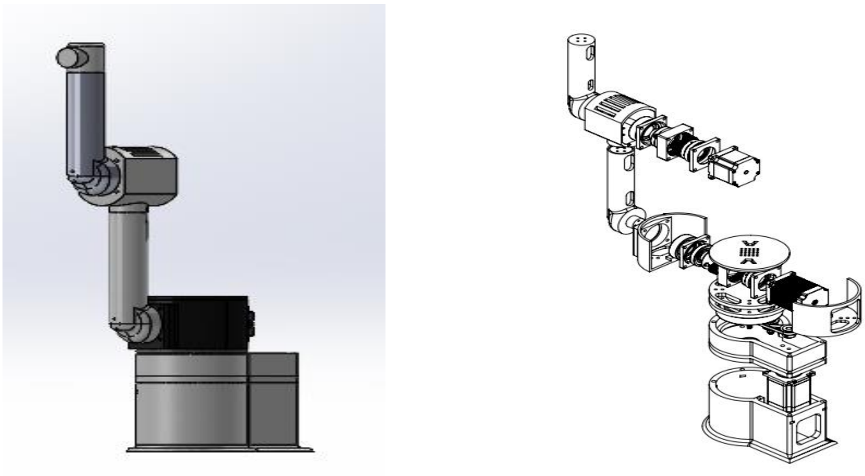


Fig. 5 Complete and exploded views of robot arm. Left shows the assembled design, right shows the exploded view.

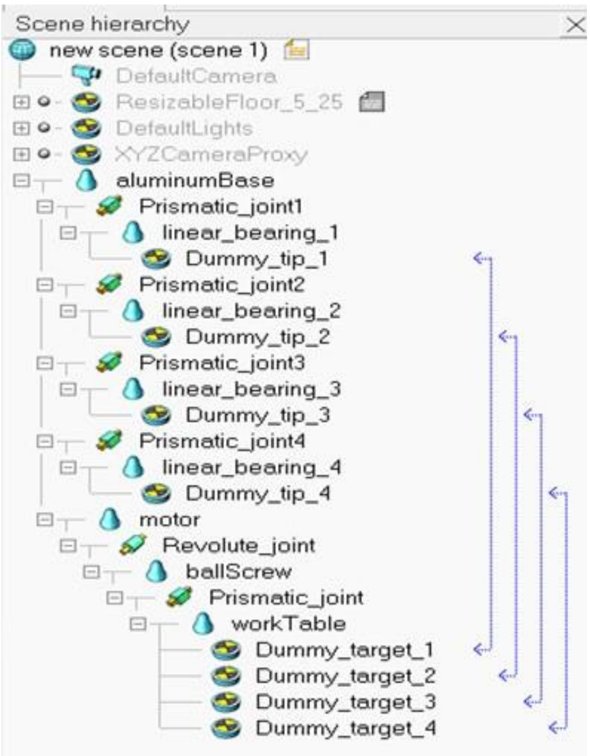


Fig. 6 The Scene hierarchy of the linear rail used in CoppeliaSim.

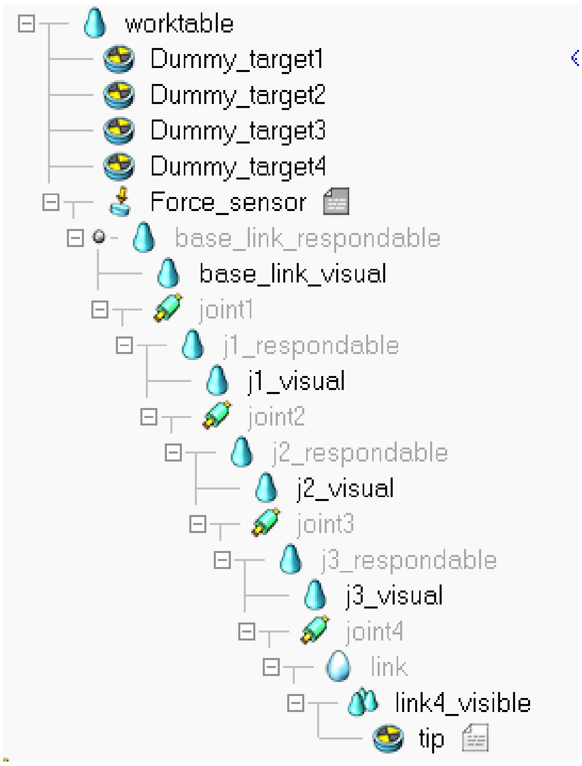


Fig. 7 The Scene hierarchy of the robot arm used in CoppeliaSim.

3. Simulation Results

The study described focuses on evaluating the effectiveness of the Ruckig online trajectory generation (OTG) algorithm in a simple path planning task, implemented in CoppeliaSim using LUA programming language and executed on a Windows PC with an Intel i7-4770 CPU and 8GB RAM.

Initially, the task was structured to investigate four distinct paths: path 1, 2, 3 and four each with different start and target position (see figure 8 and table 2) using the RRT-Connect algorithm, a widely recognized method predicated on the concept of Rapidly Exploring Random Trees (RRTs).

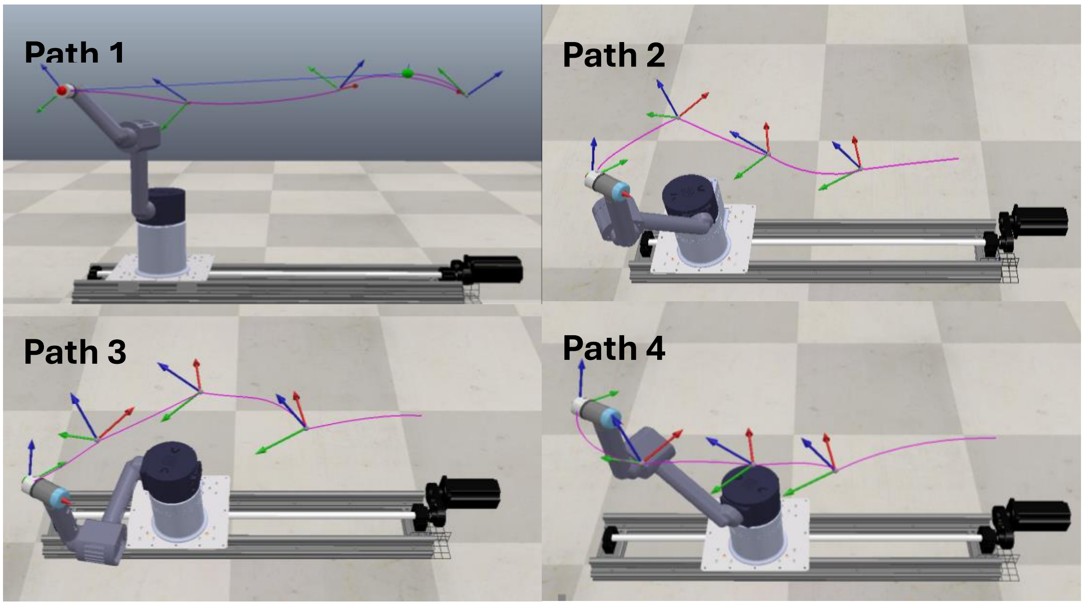


Fig. 8 The four Paths accomplished by the robot system.

This algorithm guided the robot from a predefined starting point to a target point, navigating through two intermediates via points (see figure 7).

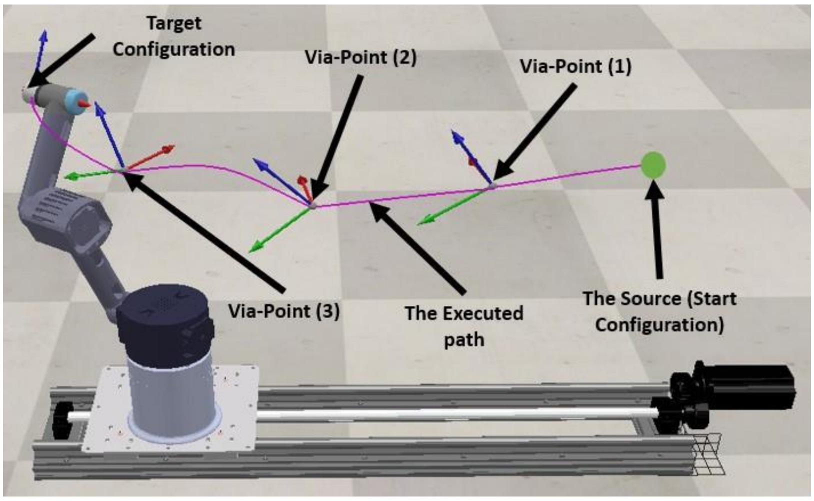


Fig. 9 The Resulted Path accomplished by the system is shown with all the acquired points as follows: (1) Start Configuration, (2) Via-Point 1, (3) Via-point 2, (4) Via-point 3, and (5) Target Configuration.

During the first stage of the experiment, the robot traversed the paths generated by RRT-Connect at a constant speed. While satisfactory in terms of actuator's acceleration and jerk, led to motion instability and rough vibrations, negatively affecting system performance. The motion profiles for path 1 showed significant variations in joints' velocities, acceleration, and jerk, which are detailed in figure 9 (a), (b), (c) and (d) for the linear rail and figure 10 (a), (b), (c) and (d) for the remaining part of the robot arm.

To address these issues, the Ruckig algorithm was employed. Ruckig, known for its capacity to generate time-optimal trajectories considering constraints on velocity, acceleration, and jerk for multiple degrees of freedom, significantly improving the system's performance. The motion profiles for path 1 have been regenerated after applying Ruckig and is depicted in figure 9 ($\bar{a}$), ($\bar{b}$), ($\bar{c}$) and ($\bar{d}$) for the rail and figure 10 ($\bar{a}$), ($\bar{b}$), ($\bar{c}$) and ($\bar{d}$) for the robot arm which exhibit smoother trajectories and the absence of abrupt shifts, which prevents system fluctuations from occurring.

The study's comparative analysis, detailed in Table 1, quantitatively illustrates the benefits of using Ruckig over constant speed. For each path, the table compares the path lengths and processing times between the two cases. The results clearly show that Ruckig significantly reduces both the path length and the time required for execution. For example, in the first scenario, Ruckig managed to halve the path length and cut the execution time by more than 90%, a substantial improvement in efficiency.

Table 2: A comparison between using Ruckig, OTG algorithm, with RRT- Connect path planning algorithm and using constant speed in terms of path length and processing time.

Coordinates		**RRT- Connect**			
		Constant Speed		**Ruckig**	
Start Position	**Target Position**	**Path length**	**Time(s)**	**Path length**	**Time(s)**
(0,0,0.7)	(0.1,0.6,0.9)	5.6	62.6	2.6	6.55
(0,0,0.7)	(-0.17,0.87,0.62)	3.8	70.1	3.15	7.67
(0,0,0.7)	(-0.17,0.85,0.49)	8.4	70	3.45	6.96
(0,0,0.7)	(0,0.85,0.59)	5	80	2.2	6.34

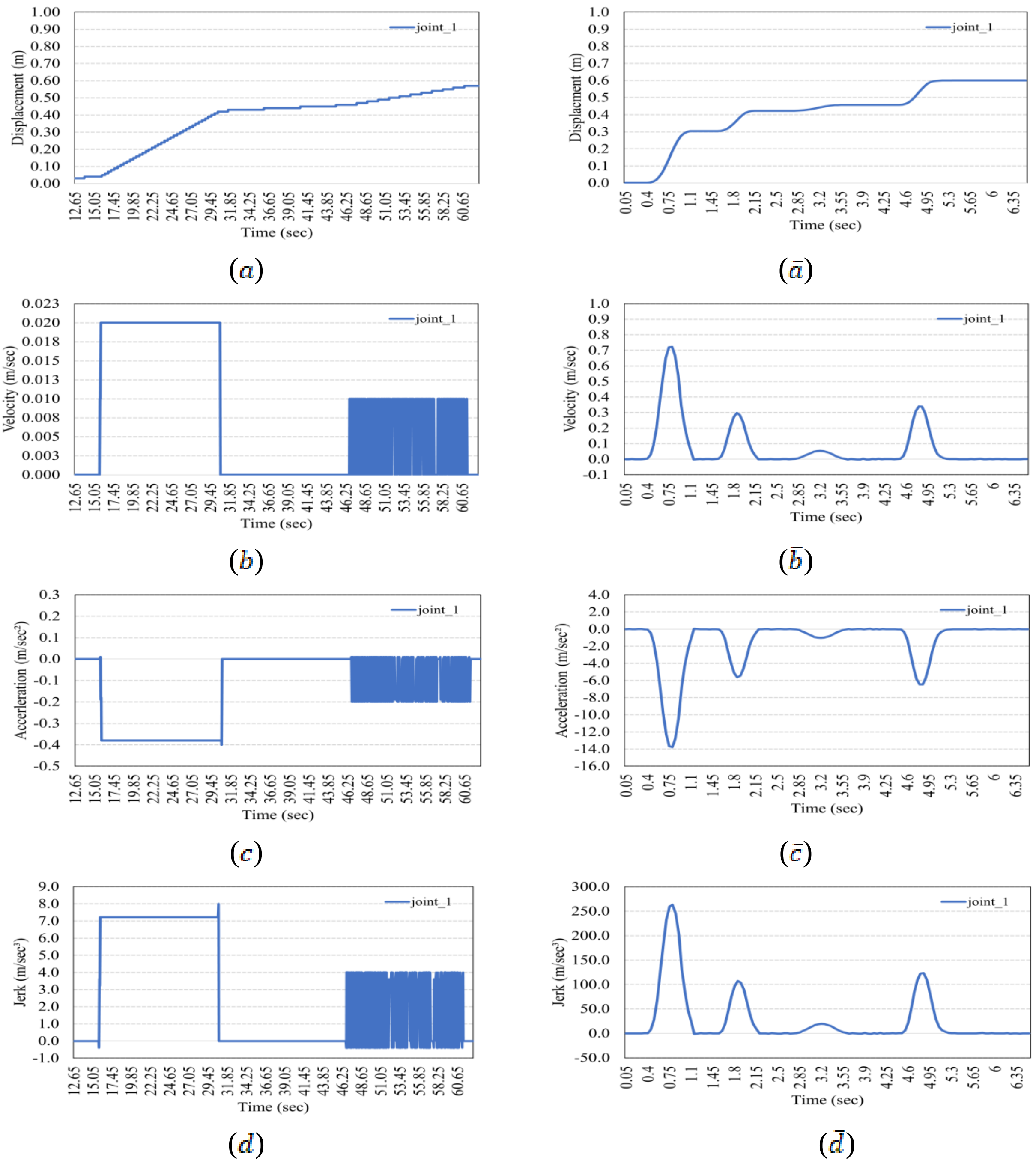


Fig. 10 Motion profile of the linear rail showing displacement, velocity, acceleration, and jerk. Graphs (a), (b), (c), and (d) for constant speed, and graphs ($\bar{a}$), ($\bar{b}$), ($\bar{c}$) and ($\bar{d}$) for Ruckig algorithm.

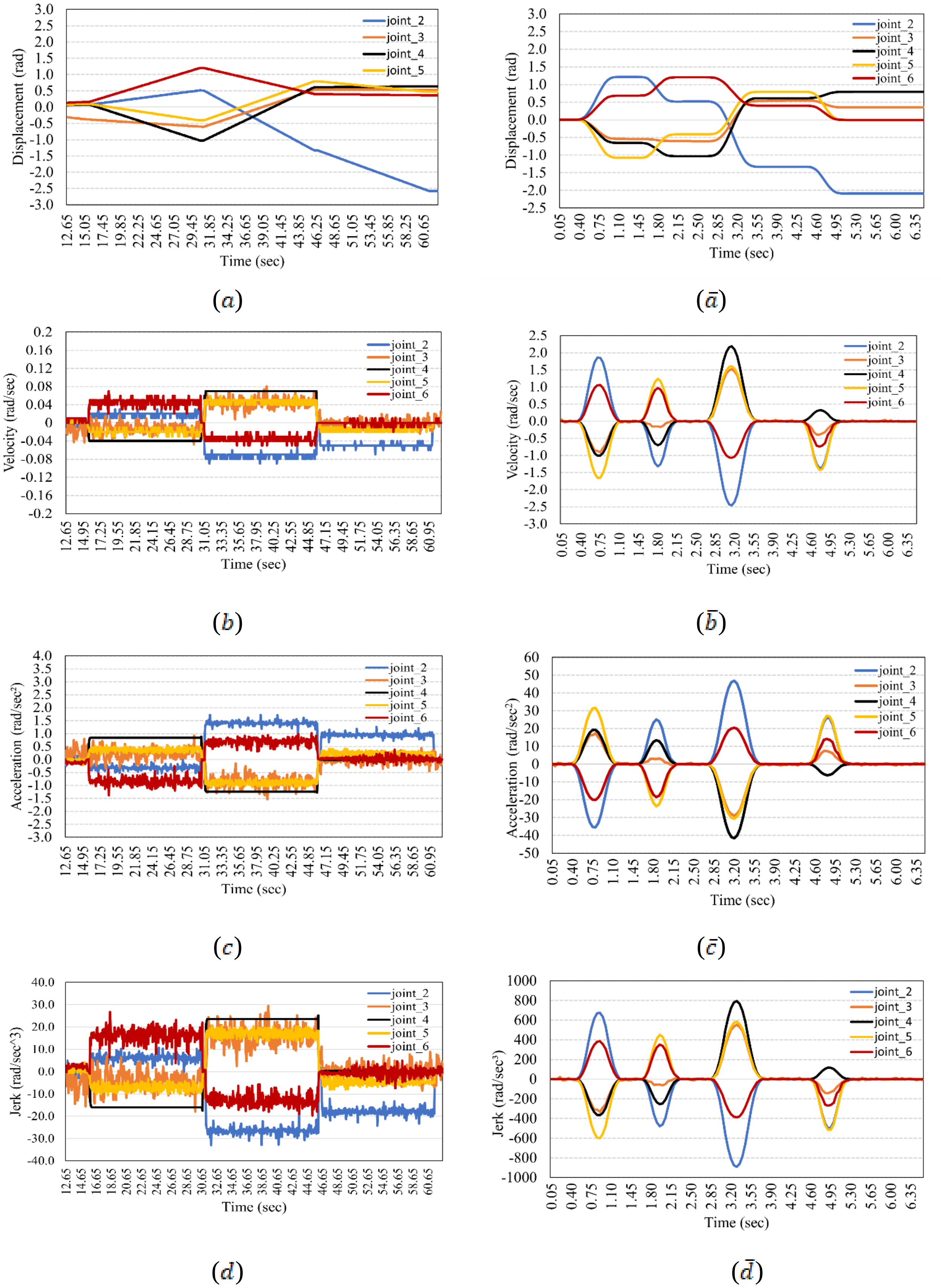


Fig. 11 Motion profile of the robot arm showing displacement, velocity, acceleration, and jerk. Graphs (a), (b), (c), and (d) for constant speed, and graphs ($\overline{a}$), ($\overline{b}$), ($\overline{c}$) and ($\overline{d}$) for Ruckig algorithm.

4. Conclusions

This study successfully utilized CoppeliaSim to model and simulate a rail-mounted robot manipulator, adopting a PRRRR configuration. Initially employing the RRT- Connect algorithm for path planning, the research then focused on assessing the impact of integrating the Ruckig online trajectory generation algorithm. The findings conclusively demonstrated that Ruckig significantly enhanced path execution, both in terms of reduced processing time and shortened path length, achieving a remarkable reduction to nearly 90% of the original execution time which in comparison to related work, within the field of OTG, Ruckig satisfy what it claims as it successfully supported different initial and target state, achieved time-optimality for a multiple DoFs robot setup. For future research, these results can be verified in an actual setup after applying dynamic analysis. Additionally, further fine-tuning of Ruckig's parameters could potentially unlock even greater efficiency and performance in robotic path planning.

References

[1] B. Siciliano and O. Khatib, Eds., *Springer Handbook of Robotics*. in Springer Handbooks. Springer, 2016. Doi: 10.1007/978-3-319-32552-1.

[2] J. J. Craig, *Introduction to Robotics*. Addison-Wesley Longman, 2005.

[3] K. M. Wurm, C. Stachniss, and W. Burgard, “Coordinated multi-robot exploration using a segmentation of the environment,” in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008, pp. 1160–1165. Doi: 10.1109/IROS.2008.4650734.

[4] S. A. Kumar, R. Chand, R. P. Chand, and B. Sharma, “Linear manipulator: Motion control of an n-link robotic arm mounted on a mobile slider,” *Heliyon*, vol. 9, no. 1, p. e12867, Jan. 2023, Doi: 10.1016/j.heliyon. 2023.e12867.

[5] A. Reiter, *Time-Optimal Trajectory Planning for Redundant Robots*. Springer Fachmedien Wiesbaden, 2016. Doi: 10.1007/978-3-658-12701-5.

[6] T. Ho, C.-G. Kang, and S. Lee, “Efficient closed-form solution of inverse kinematics for a specific six-DOF arm,” *Int J Control Autom Syst*, vol. 10, pp. 567–573, Jun. 2012, Doi: 10.1007/s12555-012-0313-9.

[7] H. A. PARK, M. A. ALI, and C. S. G. LEE, “CLOSED-FORM INVERSE KINEMATIC POSITION SOLUTION FOR HUMANOID ROBOTS,” *International Journal of Humanoid Robotics*, vol. 09, p. 1250022, Sep. 2012, Doi: 10.1142/s0219843612500223.

[8] I. Zaplana, H. Hadfield, and J. Lasenby, “Closed-form solutions for the inverse kinematics of serial robots using conformal geometric algebra,” *Mech Mach Theory*, vol. 173, p. 104835, 2022.

[9] P. Srisuk, A. Sento, and Y. Kitjaidure, “Inverse kinematics solution using neural networks from forward kinematics equations,” *2017 9th International Conference on Knowledge and Smart Technology (KST)*, vol. Not available, p. Not available, Feb. 2017, Doi: 10.1109/kst.2017.7886084.

[10] A. Malik, Y. Lischuk, T. Henderson, and R. Prazenica, “A Deep Reinforcement-Learning Approach for Inverse Kinematics Solution of a High Degree of Freedom Robotic Manipulator,” *Robotics*, vol. 11, no. 2, p. 44, 2022.

[11] N. Wagaa, H. Kallel, and N. Mellouli, “Analytical and deep learning approaches for solving the inverse kinematic problem of a high degrees of freedom robotic arm,” *Eng Appl Artif Intell*, vol. 123, p. 106301, Aug. 2023, Doi: 10.1016/j.engappai.2023.106301.

[12] H.-H. Huang, C.-K. Cheng, Y.-H. Chen, and H.-Y. Tsai, “The Robotic Arm Velocity Planning Based on Reinforcement Learning,” *International Journal of Precision Engineering and Manufacturing*, vol. 24, no. 9, pp. 1707–1721, Aug. 2023, Doi: 10.1007/s12541-023-00880-x.

[13] D. Guerin, S. Caro, S. Garnier, and A. Girin, “Optimal measurement pose selection for joint stiffness identification of an industrial robot mounted on a rail,” in *2014 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, IEEE, Jul. 2014. Doi: 10.1109/aim.2014.6878332.

[14] A. Reiter, A. Muller, and H. Gattringer, “On Higher Order Inverse Kinematics Methods in Time-Optimal Trajectory Planning for Kinematically Redundant Manipulators,” *IEEE Trans Industr Inform*, vol. 14, no. 4, pp. 1681–1690, Apr. 2018, Doi: 10.1109/tii.2018.2792002.

[15] Z. Fan *et al.*, “A Cartesian-Based Trajectory Optimization with Jerk Constraints for a Robot,” *Entropy*, vol. 25, no. 4, p. 610, Apr. 2023, Doi: 10.3390/e25040610.

[16] H. Pham and Q.-C. Pham, “A New Approach to Time-Optimal Path Parameterization Based on Reachability Analysis,” *IEEE Transactions on Robotics*, vol. 34, no. 3, pp. 645–659, Jun. 2018, Doi: 10.1109/TRO.2018.2819195.

[17] G. Wu and N. Zhang, “Kinematically Constrained Jerk Continuous S-Curve Trajectory Planning in Joint Space for Industrial Robots,” *Electronics (Basel)*, vol. 12, no. 5, p. 1135, Feb. 2023, Doi: 10.3390/electronics12051135.

[18] D. Pieper, “The kinematics of manipulation under computer control,” Stanford University Stanford, CA, USA, 1968.

[19] A. Farley, J. Wang, and J. A. Marshall, “How to pick a mobile robot simulator: A quantitative comparison of CoppeliaSim, Gazebo, MORSE and Webots with a focus on accuracy of motion,” *Simul Model Pract Theory*, vol. 120, p. 102629, Nov. 2022, Doi: 10.1016/j.simpat.2022.102629.

[20] R. Paul and B. Shimano, “Kinematic control equations for simple manipulators,” in *1978 IEEE Conference on Decision and Control including the 17th Symposium on Adaptive Processes*, IEEE, Jan. 1978. Doi: 10.1109/cdc.1978.268148.

التحليل الحركي والمحاكاة للروبوت المناول المثبت على مسار أفقي متحرك باستخدام Ruckig لتعزيز تخطيط المسار

الملخص : تُستخدم المنالات الآلية على نطاق واسع في العمليات الصناعية والرعاية الصحية نظرًا لوظائفها المتنوعة. ومع ذلك، فإن مساحة العمل المحصورة للأذرع الآلية الثابتة تحد من إمكانية تطبيقها في السيناريوهات التي تتطلب نطاقًا أوسع من الحركة. للتغلب على هذا القيد، يقدم هذا البحث دراسة حالة على نظام آلي يتكون من نظامين فرعيين أساسيين: ذراع آلية مفصلية ومسار أفقي متحرك. تم تنفيذ مهمة تخطيط مسار بسيطة باستخدام برنامج محاكاة CoppeliaSim لدراسة تأثير Ruckig، وهي خوارزمية متقدمة لتوليد المسار عبر الإنترنت، إلى جانب خوارزمية تخطيط المسار RRT-Connect. توضح هذه الدراسة قدرة Ruckig على تحسين كفاءة تخطيط المسار فيما يتعلق بوقت العملية وطول المسار. أظهرت النتائج أن Ruckig ساعد في تقليل وقت العملية بنسبة ٩٠٪ مع تحسن استثنائي في ملفات تعريف الحركة للنظام. وفيما يتعلق بطول المسار، فيبدو أنه تمكن من تقليل الطول في بعض الحالات، وليس كلها. وصل التخفيض إلى حد أقصى قدره ٥٠٪ مقارنة بالطول الأصلي.