IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 5 July 2023, accepted 13 August 2023, date of publication 8 September 2023, date of current version 13 September 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3306793

RESEARCH ARTICLE

Dual-Arm Path-Planning Algorithm for Wiring Harness Assembly Using Redundantly Actuated Robotic Systems

JIYOUNG KIM[1,2], JIN-GYUN KIM[2], JONGWOO PARK[1], BYUNG-KIL HAN[1], SANGHYUN KIM[2], (Member, IEEE), AND DONG IL PARK[1]

[1]Department of Robotics and Mechatronics, Korea Institute of Machinery and Materials, Yongin 17058, South Korea
[2]Department of Mechanical Engineering, Kyung Hee University, Yongin 17104, South Korea

Corresponding authors: Sanghyun Kim (kim87@khu.ac.kr) and Dong Il Park (parkstar@kimm.re.kr)

This work was supported in part by the Technology Innovation Program funded by the Ministry of Trade, Industry and Energy of the Korean Government under Grant KEIT20012602; and in part by Kyung Hee University, in 2023, under Grant KHU-20230877.

ABSTRACT The most challenging task in the wiring harness assembly process is the placement and connection of cables on harness boards. This task remains difficult to automate because it requires dual-arm manipulation to collaboratively grasp the harnesses. In this study, we propose a dual-arm manipulation system and a path-planning framework to automate the wiring harness task. The proposed framework enables the cooperative operation of dual-arm manipulator and plans efficient paths, while considering the kinematic constraints and obstacle avoidance conditions. The framework obtains paths in two steps: 1) the configuration of the robot arms is computed using an inverse kinematics solver; and 2) when the distance between obstacles and robots falls below a certain threshold, a sampling-based algorithm plans paths that satisfy the kinematic constraints and obstacle avoidance conditions. The merits of the framework include a significant reduction in the task time compared to existing methods. This was achieved by efficiently exploring the workspace within the constrained conditions using the proposed algorithm instead of determining the path-planning conditions throughout the entire workspace. The effectiveness of the proposed framework was validated through simulations using redundantly actuated robots with multiple DoF.

INDEX TERMS Collision avoidance, cooperative manipulation, kinematic constraints, dual-arm manipulator, path planning algorithm, redundantly actuated robotic systems, wiring harness assembly.

I. INTRODUCTION

Wiring harnesses are used in various fields as essential components, such as household appliances, automobiles, and automated devices, to transmit electricity between electronic and electrical devices [1]. Currently, the wiring task is predominantly performed manually, which incurs high labor costs and variations in the output quality based on the skill levels of the workers. Consequently, significant research has been conducted in the industrial sector to automate the wiring harness assembly process using robots for factory automation [2], [3]. To ensure the safe execution of such tasks, path-planning algorithms must be written to enable the movement of robot arms from the starting point to the target position [4], [5], [6].

The common approach to address this problem is to solve the inverse kinematics problem of the robot and obtain the joint angles at the desired positions. However, robots are complex integrated systems composed of articulated structures that require the availability of all the variables and conditions for kinematic methods to be successful. If the complete information is unavailable, the problem-solving process may fail. Moreover, kinematic methods alone cannot address the challenges posed by obstacles and similar factors [5]. Therefore, to successfully automate wiring harness assembly, robots require path-planning algorithms that not only preserve work constraints but also ensure collision avoidance and joint limit avoidance throughout the

The associate editor coordinating the review of this manuscript and approving it for publication was P. Venkata Krishna.

planned motion. This study proposes an efficient technique for path planning in flexible cable operations using dual-arm manipulation systems.

A. RELATED WORKS

Previous studies in this field primarily focused on single robots, low-degree-of-freedom (DoF) systems [7], or end-effector-based approaches [8] for performing wiring harness assemblies. However, it requires simultaneous grasping and cooperative operation of dual-arm manipulator with human-like dexterity, during processes such as cable placement and connection. Therefore, in wire harnessing, manipulators must collaborate and perform operations in complex work environments.

Because the joint configurations of each robot are interdependent in a dual-arm manipulation system, the paths of all robots must be planned simultaneously [9]. For example, in wire harnessing, robots must grasp and manipulate cables using their manipulators. To achieve this, they must share positional and status information, determine the partitioning of the workspace for each robot's responsibility, and employ path-planning algorithms that consider collision avoidance to ensure safe operation [10]. However, the search space is high dimensional [2]. To reduce the computational costs, this study proposes a method that generates paths only in collision-free areas, thereby shortening the task completion time.

Furthermore, a dual-arm manipulation system must consider the joint configuration of each robot for path planning. If a system with few DoF is employed, the feasible paths become highly restricted when additional constraints, such as joint angle limitations and collision avoidance conditions, are added. However, robots with larger DoF exhibit flexible movements even in complex work environments and can plan paths considering collision avoidance and joint constraints. Thus, robots with larger DoF are more suitable for complex wiring harness tasks.

Finally, motion-planning algorithms that satisfy the constraints of robots are being actively researched [11]. Sampling-based algorithms [12], [13] generate paths that minimize energy consumption and cover the shortest distance among numerous possible paths from the start point to the destination in a probabilistic manner. This enables effective exploration of paths in complex high-dimensional spaces. However, these algorithms are challenging to use when simultaneously considering kinematic constraints and collision avoidance tasks, because they construct paths based on the end-effector of the robot. Moreover, when employing a dual-arm manipulation system, randomly selecting grasping points and verifying their feasibility becomes time-consuming as the search space increases with the number of robots. Additionally, introducing constraints between dual-arm manipulator necessitates finding paths in a considerably narrower and more complex space than the conventional joint space. Thus, when performing cooperative motion planning for wiring harness tasks within the joint space of the robots, research is required to satisfy not only the path between the start and end points but also the node constraints that minimize the robot's implementable motion and energy consumption [14].

B. CONTRIBUTIONS

In this study, we propose a process-motion-planning algorithm that uses dual-arm manipulation to enhance the efficiency and safety of wiring harness tasks. The key contributions of this study are as follows:

- To prevent collisions between robots and external obstacles, we calculated the minimum distance between the robot link-link and link–camera centers and proposed a task-specific objective function based on the DoF. Additionally, we utilized inverse kinematic conditions to perform the task and employed a sampling-based algorithm to fetch collision-free paths for the robot without colliding with obstacles. We proposed a motion-planning algorithm that simultaneously considers the target poses and joint values for each task.
- While extant robot motion-planning methods primarily focus on performing tasks by manipulating the robot's end effectors, this study considers information regarding both the target poses and joint values for each task. Furthermore, we utilized inverse kinematics to propose a method for obtaining joint values for newly added nodes, thereby enabling the development of a motion-planning algorithm that considers all robot joints.
- To address the issue of unnecessary motions in path planning, which can occur in sampling-based algorithms, we used the Ramer–Douglas–Peucker (RDP) algorithm. This algorithm simplifies the paths generated by the proposed algorithm and removes unnecessary points, thereby resulting in an efficient representation of the path. Through simulation validation, the proposed technique demonstrated superior performance in terms of robot adaptability, task efficiency, and safety compared with existing methods.

The remainder of this paper is structured as follows. Section II provides the background information that is utilized throughout the paper. Section III outlines the algorithm framework and introduces the methodology. Section IV presents the simulation results of the proposed approach. Finally, Section 5 summarizes the evaluation of the method.

II. PROPOSED ALGORITHM FRAMEWORK

As shown in Fig. 1, this study presents an overall path-planning method for flexible cable manipulation using dual-arm manipulation. The flow of the algorithm is as follows.

1) Receive the current coordinates ($\mathbf{p}_i$), next coordinates ($\mathbf{p}_{i+1}$), target position, and direction as inputs for the task.
2) Solve the inverse kinematics problem using the position and angle information. (See Sec. II-A)

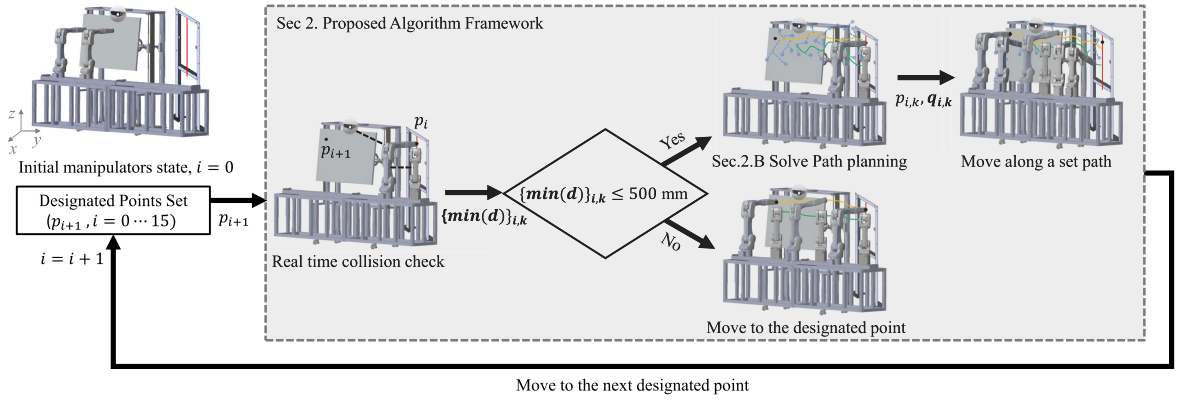


FIGURE 1. Overview of the proposed framework.

3) Verify if each manipulator is within the danger radius by utilizing information about the manipulators, obstacles, and self-collisions by calculating $\min(\mathbf{d}_{i,j})$.
4) Execute the path-planning algorithm considering the robot's joints and computational cost information. Among the candidate nodes, select the waypoint with the lowest computational cost, which includes joint angles ($\mathbf{q}_{NN}$) and distance information (dlc_{NN}) between the robot and the obstacles for both being at a random node and at the newly generated random node. Find the new current node n, which has the lowest computational cost among the generated tree at each cycle. Select a waypoint as a bypass point from arbitrarily chosen points that are a certain distance away from the obstacles and have the minimum Goal score. Connect the selected points to the starting and target points to generate the path. (See Sec. II-B)
5) Calculate the inverse kinematics values after obtaining the path.
6) If there are segments with waypoints that do not have inverse kinematics solutions, return to the segment with waypoints that have inverse kinematics solutions and randomly expand the tree. Next, select a random point and check if it satisfies the constraints and Goal score for the randomly generated sample at a certain distance from the nearest node. If satisfied, add it to the tree and connect the starting and target points to generate a path that avoids obstacles. If a Goal score exists without obstacles and is smaller than the existing Goal score, expand the new sample for each segment.
7) Move the manipulators.
8) End the algorithm if the current node gets sufficiently close to the target position.
9) Perform the harness operation and move to the next target coordinates.

Therefore, this method controls path planning and movements by considering the joint constraints of the robot, enabling the robot to perform tasks safely and efficiently.

A. REDUNDANCY RESOLUTION PROBLEM

The inverse kinematics with weighting matrix in the proposed scheme, as shown in Fig. 2. The joint configuration of the robot must be determined so that it can operate along a computed end-effector path. In this system, the weighted least-norm solution to generate joint movements, considering the range limitations of the joints. This approach enabled the robot to move to the desired position and orientation while adhering to the constraints imposed by the joint limits [3], [15], [16].

$$\dot{\mathbf{q}} = \mathbf{W}^{-1}\mathbf{J}^T(\mathbf{J}\mathbf{W}^{-1}\mathbf{J}^T)^{-1}(\dot{\mathbf{x}} + \mathbf{K}(\mathbf{x}_d - \mathbf{x})). \tag{1}$$

Here, the weighting matrix, $\mathbf{W}$, are represented by a 7×7 diagonal matrix, in which increasing the value of the i-th weight result, w_i in a smaller inverse weight in the inverse kinematics equation, which restricts the motion of the corresponding joint.

1) JOINT LIMIT AVOIDANCE

To generate weighting matrix for joint limit avoidance, the weight w_i corresponds to the i-th diagonal element of the weight matrix $\mathbf{W}$:

$$w_i = \begin{cases} 1 + |\dfrac{\partial H}{\partial q_i}|, & \text{if } \Delta|\dfrac{\partial H}{\partial q_i}| > 0, \\ 1, & \text{otherwise.} \end{cases} \tag{2}$$

In (2), the potential function was used to determine the proximity of the joint positions to the upper and lower limits. To capture the variations in the potential function with respect to each joint change, the gradient vector of the potential function is expressed as follows:

$$\frac{\partial H}{\partial q_i} = \frac{(q_{i,max} - q_{i,min})^2(2q_i - q_{i,max} - q_{i,min})}{4(q_{i,max} - q_i)^2(q_i - q_{i,min})^2} \tag{3}$$

Consequently, the weight term $|\frac{\partial H}{\partial q_i}|$ indicates that as its absolute value increases, the magnitude of the gradient, W, decreases, resulting in slower joint velocities. This means that, if the change in the absolute value of the gradient is

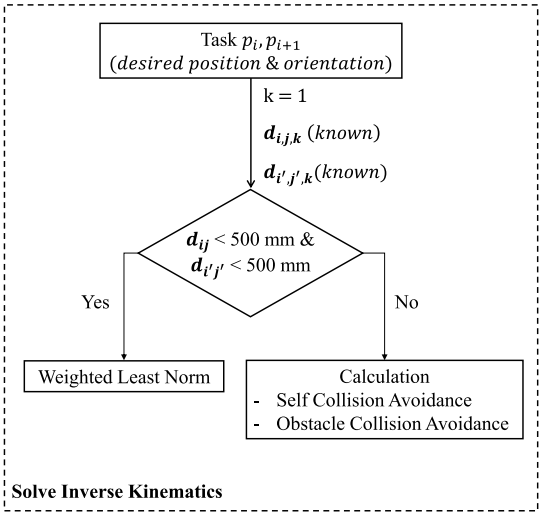


FIGURE 2. Flowchart of the proposed Inverse Kinematics algorithm.

positive, i.e., the joint is moving toward its upper or lower limit, we introduced weights to restrict the movement. If the change in the absolute value of the gradient is zero or negative, i.e., the joint maintained its position, we assigned a weight of 1 to allow unrestricted movement.

The weighted least-norm solution algorithm utilizes weights to control joint movements and introduces a potential function based on joint positions to account for joint range limitations. However, this algorithm does not provide specific collision detection or avoidance methods. Therefore, it should be integrated with other algorithms that address collision detection and avoidance, which is necessary to plan the path of the robotic arm, as shown in Sec. II-B.

2) SELF COLLISION AVOIDANCE

In this section, the concept of a minimum distance between robot links is explained, and an algorithm for estimating and monitoring the distance between adjacent links is proposed. To achieve this, a safety radius larger than its actual radius was set for each link, and the minimum distance formula was used to calculate the shortest distance between links.

When the sensors detect a collision within the safety radius between links, the joint values are determined using a cost function. This cost function was formulated to be inversely proportional to the minimum distance between the joints, meaning that the cost function value increased as the distance between the links decreased. This ensured that the reciprocal values of the minimum distances for all the links decreased, thereby avoiding self-collisions.

$$w_i = \begin{cases} 1 + |\frac{1}{d_{i,j}}|, & \text{if } \Delta|\frac{1}{d_{i,j}}| > 0, \\ 1, & \text{otherwise.} \end{cases} \tag{4}$$

where $d_{i,j}$ represents the minimum distance between the i-th and j-th links, as shown in Fig. 3. And $\frac{1}{d_{i,j}}$ denotes the reciprocal of the minimum distance. As the denominator

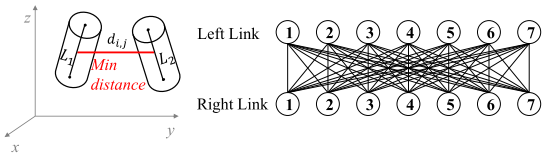


FIGURE 3. Minimum distance between links.

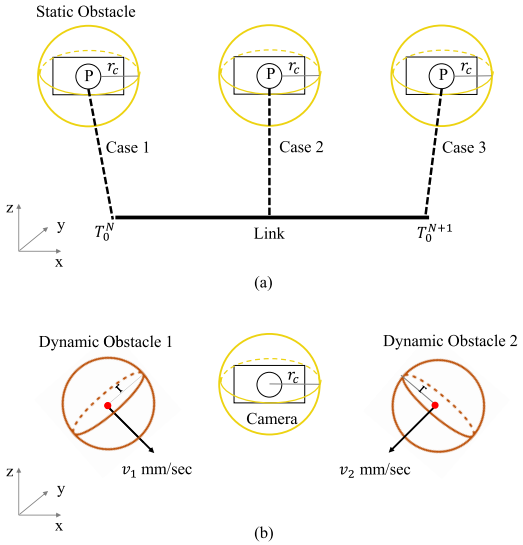


FIGURE 4. (a) Static Sphere Obstacle Collision Check, (b) Dynamic Sphere Obstacle Collision Check.

decreases, the distance between the links decreases and the value of $\frac{1}{d_{i,j}}$ increases. Condition $\Delta|\frac{1}{d_{i,j}}|$ indicates that the rate of change of $\frac{1}{d_{i,j}}$ is positive, indicating a decrease in the distance between the links. Therefore, a cost function h is added when the distance between the links decreases.

3) OBSTACLE COLLISION AVOIDANCE

When a robot collides with an obstacle in its workspace, it can damage itself, the working environment, or surrounding equipment. Therefore, in this study, an algorithm was proposed to avoid collisions by defining the subtask objective function (h) of the robot as the minimum distance between the robot links and the center of the obstacle (camera), as shown in Fig. 4.

To design the algorithm, three cases are considered:

- Case 1: When the center of the obstacle is located outside the endpoint $\mathbf{T}_0^N$ of the link, the distance between $\mathbf{T}_0^N$ and the camera center ($\mathbf{p}$), subtracting the camera's size as the radius (r_c), is computed and used as the cost function.
- Case 2: When the obstacle is located between endpoints $\mathbf{T}_0^N$ and $\mathbf{T}_0^{N+1}$ of the link, the distance between a point and line, after subtracting the camera's radius r, is used as the subtask objective function.
- Case 3: When obstacle p is located outside endpoint $\mathbf{T}_0^{N+1}$ of the link, the distance between $\mathbf{T}_0^{N+1}$ and $\mathbf{p}$, subtracting the camera's radius, is used as the subtask objective function.

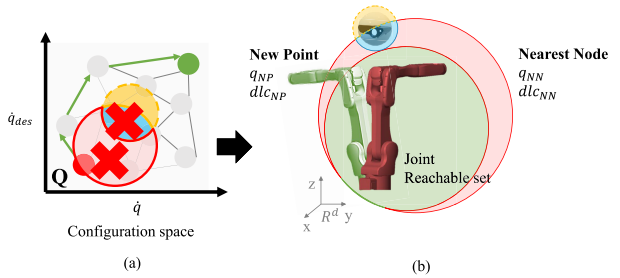


FIGURE 5. (a) Static sphere obstacle collision verification, (b) dynamic sphere obstacle collision verification.

Furthermore, a path-planning algorithm that considers dynamic obstacles is required for a robot to achieve safe movement and task execution in a dynamic environment [17], [18]. In this study, a velocity-based dynamic obstacle method was proposed to address the path-planning problem of a robot.

The algorithm generates smooth paths by considering the current position, dynamic obstacle position, and velocity information during the iterations. It performs collision avoidance and explores safe paths based on the position and velocity of dynamic obstacles. Because this algorithm considers the position and velocity of an obstacle under the assumption of known static obstacle conditions, it is applicable when such information is available

$$d_{i',j'} = \begin{cases} norm(\mathbf{T}_0^N, \mathbf{p}) - r_c & \text{Case 1} \\ norm(cross(\mathbf{T}_0^N \mathbf{T}_0^{N+1}, \mathbf{T}_0^N \mathbf{p}) - r_c & \text{Case 2} \\ norm(\mathbf{T}_0^{N+1}, \mathbf{p}) - r_c & \text{Case 3} \end{cases} \quad (5)$$

B. DUAL-ARM MANIPULATOR PATH PLANNING PROBLEM

Robot path planning aims to minimize the energy consumed by the robot while translating from the starting point to the destination with the shortest possible distance. To achieve this, sampling-based algorithms [12], such as RRT, PRM, and RRT*, are commonly used. These algorithms probabilistically extend trees and efficiently explore obstacle-free regions to generate paths, as shown in Fig. 6.

Previous research has primarily focused on connecting paths by considering only the end effectors of robots, particularly highly articulated manipulators. However, this approach may not fully utilize the robot's DoF, and instead may impose constraints on path generation and collision avoidance if joint movements are not considered. In this study, a method is proposed to consider both the robot's joint space and workspace by representing the local planner values as the motor angles (q_i) of each joint.

$$\mathbf{q} = \begin{bmatrix} q_{i1} & q_{i2} & \cdots & q_{i7} \end{bmatrix}^T \quad (6)$$

Furthermore, in previous studies, cost functions were used to evaluate the cost required to move to an arbitrary point, typically determining the nearest node using the Euclidean distance. However, those methods did not consider the robot's joint values but only the closest distance based on the end effector. Therefore, they may not accurately reflect the cost or feasibility of paths in complex environments involving joint constraints, obstacles, kinematic constraints, or high-dimensional configuration spaces.

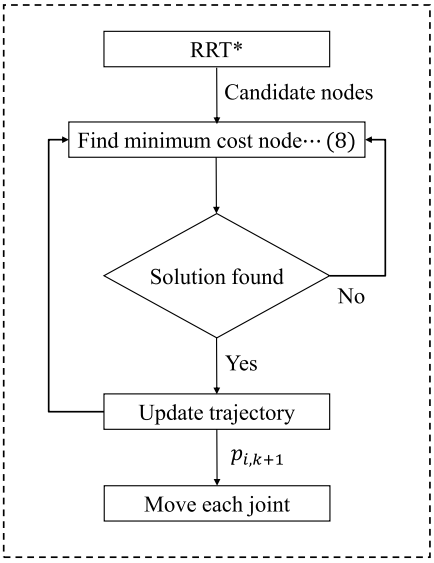


FIGURE 6. Flowchart of the proposed path-planning algorithm.

To address this issue, this study proposed a new cost function ($\mathbf{q}_{NN}$, $\mathbf{q}_{NP}$) that was redefined in the joint space. This cost function combines the rates of change of the joint angles and the distance between the robot and camera, using the nearest node ($\mathbf{q}_{NN}$) and new point ($\mathbf{q}_{NP}$) of the joints as inputs. In other words, as the rate of change increased and the distance between the robot and the camera decreased, the cost function increased.

To calculate the cost function, the method took as inputs the joint angles ($\mathbf{q}_{NN}$) and distance to obstacles (dlc_{NN}) when the robot is at the current node (nearest node) and the joint angles ($\mathbf{q}_{NP}$) and distance to obstacles (dlc_{NP}) at the newly generated node. The cost was computed as the sum of differences between the joint values when moving to the new point $\sum_{i=1}^{7}(\mathbf{q}_{NP} - \mathbf{q}_{NN})$ and the ratio of the difference between the distance to obstacles at the new point and the current obstacle distance. In other words, as the rate of change of the joint angles increased and the distance between the robot and the obstacles decreased, the cost function increased. This process was iteratively updated until a solution was obtained.

$$cost = \frac{\sum_{i=1}^{7}(\mathbf{q}_{NP} - \mathbf{q}_{NN})}{dlc_{NP} - dlc_{NN}}. \quad (7)$$

Through this approach, the cost function verifies the existence of paths that satisfy the constraints between the joint values of each node, thereby enabling path generation by considering the robot joint values and distances.

On the other hand, the RRT* algorithm generates numerous points to represent the state space, which can lead to increased computational complexity and memory

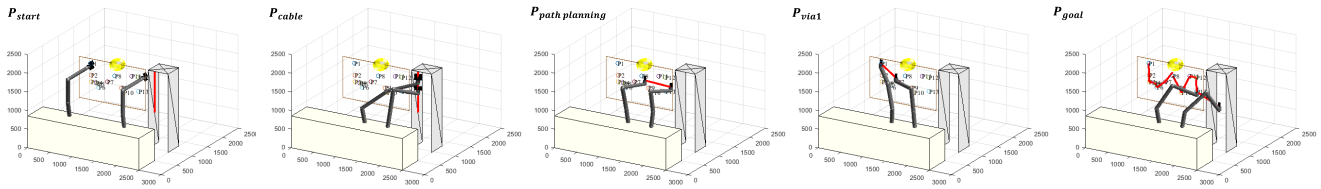


FIGURE 7. Simulation results of the proposed framework for wiring-harness manipulators in Scenario 1.

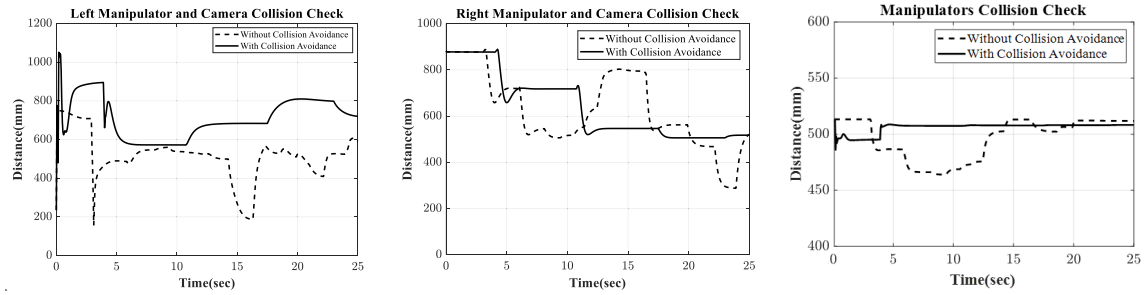


FIGURE 8. (a) Left manipulator and camera collision check, (b) right manipulator and camera collision check, (c) manipulators collision check in Scenario 1.

usage for path calculations. The RDP algorithm reduces unnecessary points while maintaining a close approximation of the original curve in a simpler form. By applying this algorithm, the complexity of curve can be reduced, and the memory usage can be optimized. Therefore, in this study, the Ramer–Douglas–Peucker algorithm was introduced to reduce redundant points in the sampling-based algorithm. The algorithm operates according to the following steps: First, a set of n points was defined on curve C starting from the initial point ($\mathbf{P}_1$). Assuming these points form a line segment, we calculated the distances between each point and the line segment. Among these, the point with the maximum distance that satisfied a predefined threshold condition was identified. This point was referred to as the farthest point. Another line segment was drawn connecting the farthest point and this line segment. Based on the newly drawn segment, the curve was divided into two subcurves. The same process was recursively applied to each subcurve. By repeating this process, we obtained a simpler form of the curve, which approximated the original curve.

III. VALIDATION RESULTS

A. SCENARIO 1: STATIC OBSTACLE

To validate the performance of the proposed algorithm, it was embedded in a simulation to verify whether the two robots, in the presence of static and dynamic obstacles, could evade obstacles and accurately track the target positions while performing wiring harness tasks. Simulation results of the entire robot path in an environment with static obstacles are illustrated in Fig. 7. Sixteen tasks were performed, as follows: (1) starting from the initial point ($\mathbf{P}_{start}$), (2) moving along the cable path ($\mathbf{P}_{cable}$), (3) moving from markers **P1** to **P13**, and (4) verifying the successful execution of tasks, including reaching the goal and direction ($\mathbf{P}_{goal}$). For the left arm, the simulation started from the initial state $\mathbf{P}_{start} = [700, 350, 850]$. $\mathbf{P}_{cable}$ moved along the cable path and subsequently reached $[2200, 950, 1850]$. $\mathbf{P1} - \mathbf{P3}$ involved moving from markers **P1** to **P13**. The goal position $\mathbf{P}_{goal}$, was set to [700, 350, 850]. Similarly, the right arm started from the initial state $\mathbf{P}_{start} = [1930, 350, 850]$. $\mathbf{P}_{cable}$ moved along the cable path, reaching $\mathbf{P}_{cable} = [2200, 950, 1600]$. $\mathbf{P1} - \mathbf{P13}$ involved moving to marker **P1**. The goal position for the right arm was set as the arrival point (in place) at [1930, 350, 850].

The static obstacles were represented as spheres with a radius of 150 mm, whereas the actual size of the robot was 210 mm. Consequently, when the distance between the robot and the obstacle was less than 500 mm, the situation was considered a collision risk. Collision detection was implemented to ensure that the robot could move along a safe path. The tolerance for the position error was set to 50 mm and that for the direction error to 50 rad. The algorithm was designed to terminate when both the position and direction errors were within the specified ranges and the final path was reached. Figure 8 compare the distances between the left arm robot and obstacle, right arm robot and obstacle, and the avoidance algorithm before and after the application, respectively. The distance represents the distance between the joint angles of the seven joints of the robot and the minimum distance of the obstacle.

Figure 8(a) shows that, before deploying the collision avoidance algorithm, the distance between the left-arm robot and obstacle led to a collision after 4 and 16 s of simulation. However, after deploying the collision avoidance algorithm, the minimum distance was maintained at 500 mm

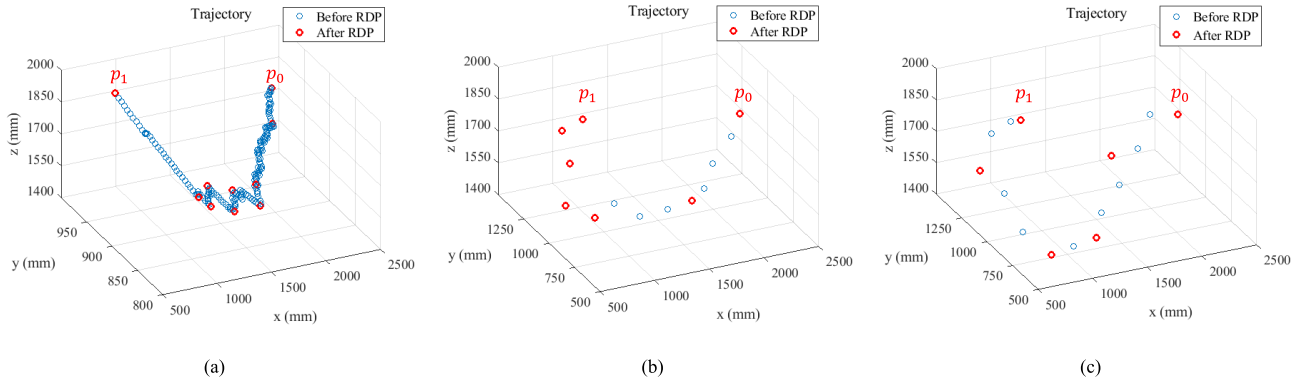


FIGURE 9. Simulation results using Ramer–Douglas–Peucker algorithm: (a) RRT*, (b) 1 static obstacle path planning, (c) 1 static obstacle 2 dynamic obstacle path planning in Scenario 1.

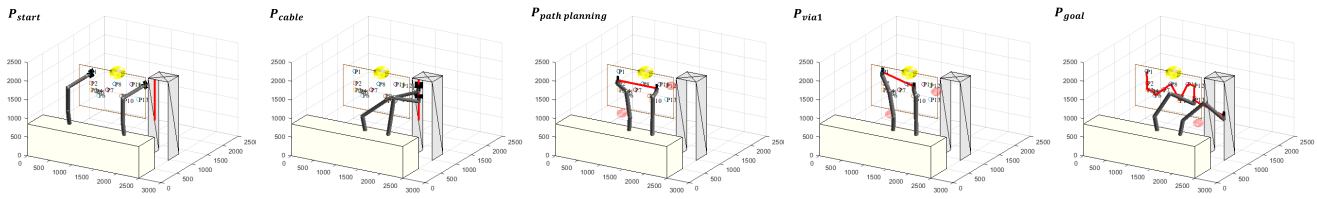


FIGURE 10. Simulation results of the proposed framework for wiring-harness manipulators in Scenario 2.

or more to ensure safe execution of the task. Similarly, Figure 8(b) demonstrates that, before the algorithm was deployed, the distance between the right-arm robot and the obstacle would lead to a collision after 23 s of simulation. However, after deploying it, the minimum distance was maintained at 500 mm or more to ensure safe execution of the task. Figure 8(c) illustrates that, before execution, a collision between two robots was expected between 5 and 10 s in the simulation. However, after execution, a safe path without collisions was generated. According to the experimental results, while moving from $\mathbf{P}_{cable}$ to **P1**, the robot encountered challenges in searching for a narrow path that avoids the obstacles. However, when the proposed algorithm was deployed, it successfully generated paths that allowed the robot to move safely without colliding with any obstacles. These findings demonstrate the effectiveness of the proposed algorithm for robot path planning in complex environments.

To overcome this issue with Sec. II-B, in Fig. 9, the results of the implementation of the RDP algorithm to path optimization are evaluated. This experiment was conducted to assess the performance of the proposed algorithm in path optimization for robot motion planning. An epsilon threshold value of 30 mm was used for the optimization. The experimental results show that, when the existing RRT* algorithm was used to generate paths, as shown in Fig. 9(a), 183 paths were produced, whereas this number was reduced to 10 after the RDP algorithm was applied. This reduction indicated the elimination of unnecessary parts, which resulted in shorter, more efficient paths. Additionally, path simplification reduced the computation time of the robot. Figure 9(b) presents the results in a static obstacle environment, where the initial path consisted of 13 points, which were reduced to six points after optimization. Moreover, by setting the epsilon value to 30 mm, the system achieved appropriate path optimization. This demonstrated the capability of the RDP algorithm in simplifying robot trajectories and generating efficient paths in static obstacle environments. Similarly, Fig. 9(c) illustrates the results in a dynamic obstacle environment, where the initial path consisted of 15 points, which was reduced to six points after optimization. Once again, setting the epsilon value to 30 mm resulted in an optimal path. The RDP algorithm was utilized to optimize the robot's paths by considering dynamic obstacles. This highlights the capability of the RDP algorithm to simplify robot trajectories and generate efficient paths in the presence of mobile obstacles. Consequently, application of the RDP algorithm to the proposed RRT* algorithm made the benefits of path simplification evident. These findings demonstrate the utility and applicability of the algorithm in robot path planning.

B. SCENARIO 2: DYNAMIC OBSTACLE

For Scenario 2, we performed simulations to determine the possibility of a robot arm to safely perform tasks by avoiding collisions with one static and two dynamic obstacles. The radius of the robot arm was set to 210 mm, and both static and dynamic obstacles were represented as spheres. The radii of the static and dynamic obstacles were both set to 90 mm. We considered a collision risk scenario when the distance

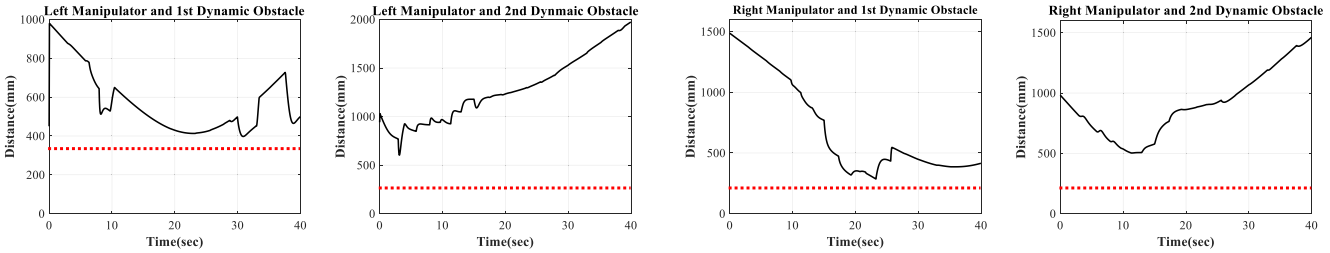


FIGURE 11. (a) Left manipulator and 1st dynamic obstacle collision check, (b) left manipulator and 2nd dynamic obstacle collision check, (c) right manipulator and 1st dynamic obstacle collision check, (d) right manipulator and 2nd dynamic obstacle collision check in Scenario 2.

between the robot and an obstacle was less than or equal to 400 mm and implemented measures to detect collisions and enable the robot to move along a safe path.

The center position of dynamic obstacle1 was [1000, 1200, 1500] mm with velocities of [2.7, −2.7, −2.5] mm/s along the x-, y-, and z-axes, respectively. Dynamic obstacle 2 had a center position of [2000, 1200, 2000] mm and velocities of [−2.5, −2.5, −2.5] mm/s along the x-, y-, and z-axes. The simulations were conducted based on these settings. The range of the position error was set to 50 mm and the direction error to 50 rad. The robot was designed to satisfy both the position error and direction error conditions, and to terminate the algorithm once it ventured on the final path, as shown in Fig. 10 and 11.

IV. CONCLUSION

This study proposed a motion-planning algorithm that used a dual-arm manipulation system for complex tasks in wiring harness operations. The algorithm considered cooperation and communication among robots, joint constraints, and collision avoidance with the aim of reducing operation time and improving efficiency.

The main contributions of the proposed algorithm are as follows:

- To address collision avoidance, a subtask objective function that incorporates self-collision avoidance and collision avoidance between the robots and obstacles was proposed. By leveraging inverse kinematics, the algorithm performs joint constraint tasks. When the distance between the robot and the obstacle is below a certain threshold, subtasks are executed for self-collision and obstacle collision avoidance. However, if the distance condition between the robot and the obstacle is satisfied, the robot uses a sampling-based algorithm to find collision-free paths with the obstacle.
- Unlike previous path-planning methods that focused only on the end effectors of the robot, this algorithm simultaneously considers the target poses and joint values for each task. Furthermore, we propose a method to determine the joint values of the new nodes being added by using inverse kinematics, thereby developing a motion-planning algorithm that considers all joints of the robot.
- To enhance the efficiency of the path-planning results, we adopted the Ramer–Douglas–Peucker algorithm. This algorithm simplifies generated paths and removes unnecessary points, thereby improving their representation. Through simulation validation, the proposed algorithm exhibited superior performance in terms of robot adaptability, task efficiency, and safety compared to existing techniques.

Future research directions include the development of collision-avoidance algorithms in various environments and performance evaluations in real work environments. In addition, the applicability and scalability of the algorithm in actual industrial settings must be explored considering the flexibility of cables.

REFERENCES

[1] B. Siciliano, O. Khatib, and T. Kröger, *Springer Handbook of Robotics*, vol. 200. Berlin, Germany: Springer, 2008.
[2] A. Aristidou, J. Lasenby, Y. Chrysanthou, and A. Shamir, "Inverse kinematics techniques in computer graphics: A survey," *Comput. Graph. Forum*, vol. 37, no. 6, pp. 35–58, Sep. 2018.
[3] J. Wan, H. Wu, R. Ma, and L. Zhang, "A study on avoiding joint limits for inverse kinematics of redundant manipulators using improved clamping weighted least-norm method," *J. Mech. Sci. Technol.*, vol. 32, no. 3, pp. 1367–1378, Mar. 2018.
[4] D. Sanchez, W. Wan, and K. Harada, "Four-arm collaboration: Two dual-arm robots work together to manipulate tethered tools," *IEEE/ASME Trans. Mechatronics*, vol. 27, no. 5, pp. 3286–3296, Oct. 2022.
[5] K. Koo, X. Jiang, A. Konno, and M. Uchiyama, "Development of a wire harness assembly motion planner for redundant multiple manipulators," *J. Robot. Mechatronics*, vol. 23, no. 6, pp. 907–918, Dec. 2011.
[6] D. Sanchez, W. Wan, and K. Harada, "Towards tethered tool manipulation planning with the help of a tool balancer," *Robotics*, vol. 9, no. 1, p. 11, Mar. 2020.
[7] P. Lehner and A. Albu-Schäffer, "Repetition sampling for efficiently planning similar constrained manipulation tasks," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 2851–2856.
[8] F. Hauer and P. Tsiotras, "Deformable rapidly-exploring random trees," in *Proc. Robot., Sci. Syst. XIII*, Jul. 2017, p. 8.
[9] A. Perez, S. Karaman, A. Shkolnik, E. Frazzoli, S. Teller, and M. R. Walter, "Asymptotically-optimal path planning for manipulation using incremental sampling-based algorithms," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2011, pp. 4307–4313.
[10] W. Sun, L. G. Torres, J. Van Den Berg, and R. Alterovitz, "Safe motion planning for imprecise robotic manipulators by minimizing probability of collision," in *Proc. 16th Int. Symp.* Cham, Switzerland: Springer, 2016, pp. 685–701.
[11] R. Shome, K. Solovey, A. Dobson, D. Halperin, and K. E. Bekris, "DRRT*: Scalable and informed asymptotically-optimal multi-robot motion planning," *Auto. Robots*, vol. 44, nos. 3–4, pp. 443–467, Mar. 2020.

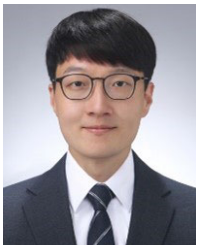
[12] M. Elbanhawi and M. Simic, "Sampling-based robot motion planning: A review," *IEEE Access*, vol. 2, pp. 56–77, 2014.
[13] M. Stilman, "Task constrained motion planning in robot joint space," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Nov. 2007, pp. 3074–3081.
[14] R. Shome and K. E. Bekris, "Anytime multi-arm task and motion planning for pick-and-place of individual objects via handoffs," in *Proc. Int. Symp. Multi-Robot Multi-Agent Syst. (MRS)*, Aug. 2019, pp. 37–43.
[15] T. F. Chan and R. V. Dubey, "A weighted least-norm solution based scheme for avoiding joint limits for redundant joint manipulators," *IEEE Trans. Robot. Autom.*, vol. 11, no. 2, pp. 286–292, Apr. 1995.
[16] A. Colomé and C. Torras, "Closed-loop inverse kinematics for redundant robots: Comparative assessment and two enhancements," *IEEE/ASME Trans. Mechatronics*, vol. 20, no. 2, pp. 944–955, Apr. 2015.
[17] H. Yang, J. Lim, and S.-E. Yoon, "Anytime RRBT for handling uncertainty and dynamic objects," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2016, pp. 4786–4793.
[18] N. Virani and M. Zhu, "Robust adaptive motion planning in the presence of dynamic obstacles," in *Proc. Amer. Control Conf. (ACC)*, Jul. 2016, pp. 2104–2109.

JIYOUNG KIM received the B.S. degree from the Department of Mechanical Engineering, Hanbat National University, in 2021. She is currently pursuing the M.S. degree with the Department of Mechanical Engineering, Kyung Hee University. Her current research interests include collision avoidance, manipulation, and multi-robot motion planning.

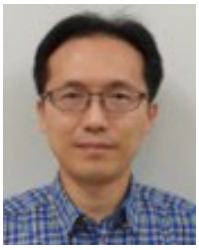
JIN-GYUN KIM was born in South Korea. He received the B.S. and M.S. degrees in civil engineering from Korea University, in 2008 and 2010, respectively, and the Ph.D. degree in ocean systems engineering from the Korea Advanced Institute of Science and Technology (KAIST), in 2014. He was a Senior Researcher with the Korea Institute of Machinery and Materials (KIMM), from 2014 to 2017. He is currently an Associate Professor in mechanical engineering with Kyung Hee University. His research interests include computational dynamics, vibration, and multiphysics modeling and simulation.

JONGWOO PARK was born in South Korea. He received the M.S. and Ph.D. degrees in control and instrumentation engineering from Korea University, Seoul, South Korea, in 2007 and 2016, respectively. He is currently a Senior Researcher with the Department of Robotics and Mechatronics, Korea Institute of Machinery and Materials (KIMM), Daejeon, South Korea. His research interests include robot control and robotic system integration.

BYUNG-KIL HAN was born in South Korea. He received the B.S. degree in mechanical and control engineering from Handong University, Pohang, South Korea, in 2008, and the M.S. and Ph.D. degrees in mechanical engineering from the Korea Advanced Institute of Science and Technology, Daejeon, South Korea, in 2010 and 2019, respectively. He is currently a Senior Researcher with the Korea Institute of Machinery and Materials. His research interests include sequence learning, reinforcement learning-based robot control, and teleoperation.



SANGHYUN KIM (Member, IEEE) received the B.S. degree in mechanical engineering and the Ph.D. degree in intelligent systems from Seoul National University, South Korea, in 2012 and 2020, respectively. He was a Postdoctoral Researcher with The University of Edinburgh, U.K., in 2020, and a Senior Researcher with the Korea Institute of Machinery and Materials, South Korea, from 2020 to 2023. He is currently an Assistant Professor in mechanical engineering with Kyung Hee University. His main research interest includes the optimal control of mobile manipulators.



DONG IL PARK was born in South Korea. He received the B.S., M.S., and Ph.D. degrees in mechanical engineering from the Korea Advanced Institute of Science and Technology (KAIST), in 2000, 2002, and 2006, respectively. He has been researching robotics and mechatronics with the Korea Institute of Machinery and Materials (KIMM), since 2006. His research interests include the design and analysis of mechanical systems, the modeling, control, and application.