IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 11 October 2022, accepted 2 November 2022, date of publication 9 November 2022, date of current version 7 December 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3221182

RESEARCH ARTICLE

Manipulator Collision Avoidance System Based on a 3D Potential Field With ISO 15066

YEON KANG[1], **DONGHAN KIM**[2], **AND DONGWON YUN**[1], **(Senior Member, IEEE)**

[1]Department of Robotics and Mechatronics Engineering, Daegu Gyeongbuk Institute of Science and Technology (DGIST), Daegu 42988, South Korea
[2]Department of Electrical Engineering (Age Service-Tech), Kyung Hee University, Seoul 17104, Republic of Korea

Corresponding authors: Dongwon Yun (mech@dgist.ac.kr) and Donghan Kim (donghani@khu.ac.kr)

This work was supported by the National Research Foundation of Korea (NRF) Grant through the Korea Government (MSIT) under Grant 2020R1C1C1012279.

ABSTRACT Recently, various intelligent technological innovations are being applied to smart factories. As manipulators are widely used in smart factories, the manipulator and the workspace of humans overlap, and interest in cooperative robots and human safety has increased. In relation to this, a collision-avoidance control algorithm applicable in three dimensions (3D) and that also meets existing safety standards for humans and robots is required. In this paper, we propose a 3D potential field-based manipulator collision avoidance algorithm that meets the requirements of the ISO 15066 standard. This algorithm applies Speed and Separation Monitoring(SSM) according to the distance between the manipulator and the obstacle and controls the speed of the manipulator slowly as the risk is higher. This allows us to overcome the limitations that existing studies have not been conducted on 3D potential field-based obstacle avoidance and that it is difficult to apply to the field without considering ISO 15066. The proposed system was verified through simulation and experiments, and through comparison with the existing algorithm, we verified that SSM was well applied to the proposed system.

INDEX TERMS Manipulator, obstacle avoidance, artificial potential field, ISO 15066.

I. INTRODUCTION

The combination of manufacturing and Internet Technology (IT) that creates competitiveness accelerates, it is focusing on a digital revolution of robotics, artificial intelligence, nanotechnology, quantum programming, biotechnology, and the internet-of-things (IoT), among other facets. In relation to this, new technological innovations are emerging in a smart factory. This typically consist of a control layer, cloud & intelligence layer, data layer, and physical layer [1], [2]. Manipulators are widely used in the physical layer, i.e., the space where robots and humans work. As a result, the manipulator and the human workspace overlap, and interest in cooperative robots and human safety increases. Starting with the enactment of ISO 10218-1/2 [3], [4], various safety requirements for collaborative robots have been established, and multiple studies are also in progress. Moreover, safety standards stipulated by ISO 15066 [5] are applied at actual industrial sites. Each ISO standard is shown in Fig. 1. ISO 10218-1 is a green area for safety design and protection dialects of industrial robots and safety requirements for information in use. ISO 10218-2 is a blue area for safety requirements as a robot system when installed and combined with industrial robot production line peripherals. Finally, ISO 15066 is a standard for cooperative robot driving in the yellow area that specifies different safety standards by dividing a zone into several areas according to the distance between a robot and a person. It is largely composed of four safety components: safety-rated monitored stop(SMS), hand guiding(HG), speed and separation monitoring(SSM), and power and force limiting (PFL).

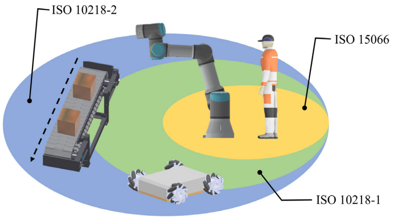


FIGURE 1. Areas covered by ISO 10218-1/2 and ISO 15066 in cooperative robot systems.

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

Research on safety assurance as this concept pertains to cooperative robots can be divided mainly into studies that focus on passive safety assurance methods and those on active safety assurance methods. Passive safety assurance methods are those that reduce the degree of influence through hardware designs of the manipulators used [6] and those that do so through a shock absorption operation [7], [8], [9]. Another approach controls the robot when an collision is detected between a manipulator based on an impact sensor and a person [10], [11], [12], [13]. However, these methods have the drawback of requiring robot-human contact.

Active safety assurance methods are based on an obstacle detection sensor and have the advantage of being safer as they do not necessarily require human contact. As safety barrier function [14] or reinforcement learning [15], [16], various obstacle avoidance algorithms are used to allow the manipulator to work in different environments. Conventionally, inverse kinematics-based obstacle avoidance algorithms were widely used for manipulator control. Using closed form inverse kinematics, there is a study that optimizes the positioning of mobile manipulators with 7 degrees of freedom [17]. Zhao, L proposed algorithm uses inverse mechanics to avoid obstacles while considering the restriction of the manipulator's joint when multiple manipulators avoid complex obstacles in the workspace [18]. In addition, Zhong,J used deep reinforcement learning and inverse kinematics to control the welding manifold to avoid obstacles and move accurately at the same time [19]. Another study presented an algorithm to avoid obstacles in dynamic situations by considering the speed between obstacle-manipulator and inverse kinematics [20]. However, a disadvantage of the inverse kinematics-based obstacle avoidance algorithms are that numerous errors occur and require a lot of computing power during the control process.

To compensate for this, obstacle avoidance control is conducted using a potential field. Tian, Y reduced computational complexity and solved the local minima problem with improved artificial potential field (APF) [21] Zhang, H also established collision distance thresholds and Jacobian-based speed control with a distance between two manipulators in APF to prevent double robotic arm collisions [22], and studies used potential fields and angle limit functions to reduce errors and improve computational speed [23]. However, research is now focused on moving paths in two dimensions (2D).

To overcome this limitation, Wang, C propose a strategy that avoids obstacles that non-holonomic robots move in three dimensions (3D) [24], inspired by sliding mode control [25]. In addition, research has been conducted to reconstruct obstacles based on machine learning to avoid 3D obstacles in the manipulator [26] or programming languages that efficiently generate sample environments composed of complex relationships between three-dimensional objects by combining probabilistic programming techniques and convex computational geometry [27]. Makita, S proposed a system that implements the position of the manipulator and obstacles in augmented reality [28] and virtual reality to guide them to avoid obstacles in 3D, and B established a 3D-collision-force map to study algorithms in which the manipulator avoids obstacles [29]. Regardless, there is a limitation in that it is difficult to apply to a manipulator system that must be path-independent or that the location of obstacles must be accurately known in advance.. Furthermore, it is difficult to avoid when obstacles are located in blind spots where manipulators cover the detection area of sensors or cameras, and they are not consider existing cooperative robot safety standards. To overcome these limitations, we need APF-based obstacle avoidance algorithms built in 3D without blind spots and satisfies the ISO 15066 criterion for application to industrial sites simultaneously.

We propose the method of obstacle avoidance control of the manipulator using a 3D potential field without a lack of perception that meets the ISO 15066 standard. The first contribution point of our system is to integrate the 2D APF generated with each sensor into 3D APF by placing eight sensors at 45-degree intervals on the surface of the manipulator link. Based on this, the system creates an obstacle avoidance path for the manipulator without a blind spot. Conventional research related to APF-based obstacle avoidance mainly focused on minimizing the local minima of the manipulator obstacle avoidance path created by placing one sensor or camera outside the manipulator. In the field of collaborative robots, if an APF is generated through a small number of sensors used outside the robot, there is a limitation that a robot, a person, or an obstacle cannot avoid a collision in a blind spot that occurs covering the sensor detection area. Our first novelty can overcome this limitation. The second contribution point is to satisfy the 15066, SSM, for application to cooperative robot systems when controlling manipulator obstacle avoidance. Current research on APF obstacle avoidance pathways has focused on obstacle avoidance with fast and optimal paths, but no consideration has been made for manipulators working with humans. Therefore, we set up a risk zone according to human stride and applied manipulator controls that apply different speed limits for each area.

Fig. 2 shows a flowchart of the system proposed in this paper. The system calculates the obstacle's position with the sensor's data and creates a 2D potential field for each sensor. Moreover, it integrates this to generate a 3D potential field and speed control the manipulator according to SSM criteria determined by the distance between the obstacle and the manipulator. Through such a system, the sensor detection range in the cooperative robot is not obscured by the manipulator, and obstacles can be avoided. When avoiding an object in space while satisfying the ISO standard, the trajectory of the end effector can pass by avoiding the object. As a result, in this paper, only the position of the end effector was controlled. Accordingly, to reduce the calculation speed, control was performed by simulating with a simplified three-axis manipulator, except for the three axes mainly used for orientation control of the manipulator.

The contents of the paper are organized as follows. Section 2 presents the overall composition of the control

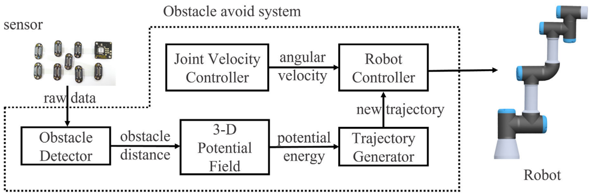


FIGURE 2. Flowchart of a 3D potential field-based manipulator obstacle avoidance system.

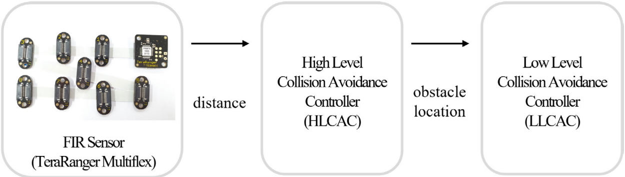

FIR Sensor (TeraRanger Multiflex)
distance
High Level Collision Avoidance Controller (HLCAC)
obstacle location
Low Level Collision Avoidance Controller (LLCAC)

FIGURE 3. Structure of the 3D potential field-based obstacle avoidance system.

TABLE 1. D-H parameters of three-axis rotary manipulators.

Joint	a_i [m]	α_i [°]	d [m]	θ_i [°]
1	0	90	$L_1 = 0.15$	θ_1
2	$L_2 = -0.24$	0	0	θ_2
3	$L_3 = -0.21$	0	0	θ_3

system, the dynamic modeling used in each controller, the sensor algorithm constituting the obstacle avoidance control system, and the 3D potential field algorithm. In Chapter 3, the proposed control system is applied to the manipulator to perform simulation, and the existing and proposed algorithms are compared and validated. Section 4 covers the conclusions and presents ideas for future work.

II. PROPOSED OBSTACLE AVOIDANCE SYSTEM

Fig. 3 shows the structure of the 3D potential field-based obstacle avoidance system proposed in this paper. First, the distance between the manipulator and the obstacle is measured with eight far infrared (FIR) sensors, and the obstacle detection system is used in a high-level collision controller (HLCAC) to determine how far away the manipulator and the obstacle are and where they are located. Using these results, the low-level collision avoidance controller (LLCAC) determines the current danger level according to ISO 15066, and according to this outcome, speed control and 3D potential field control are performed to allow the manipulator to avoid obstacles.

A. HLCAC

1) KINEMATIC CONFIGURATION

Kinematic modeling of the manipulator is required to obtain the exact location of the obstacle using distance data measured by the FIR sensor in the HLCAC. The three-axis rotary manipulator used in this study involved the locking of the fourth, fifth, and sixth axis of a six-axis manipulator (UR3, Universal Robots) [30], as shown in Fig. 4. The numbers in the figure refer to the numbers of joints.

The three-axis manipulator's Denavit-Hartenberg(D-H) parameters and coordinates are shown in Fig. 5. In addition, specific D-H parameters are presented in Table 1.

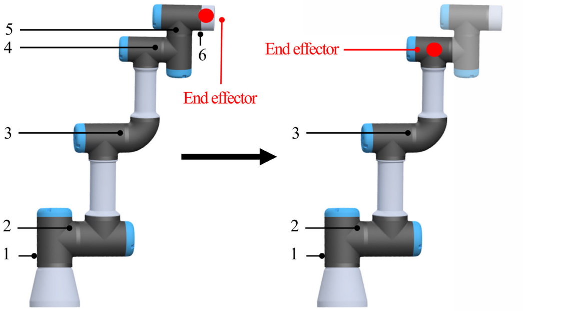


FIGURE 4. Set to three-axis manipulator by fixing fourth, fifth, and sixth joint of six-axis manipulator UR3.

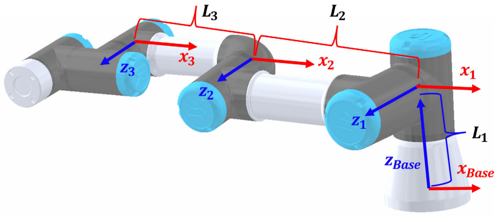


FIGURE 5. DH parameters and coordinate systems of the manipulator.

Based on the above DH parameters, the relationship between the two adjacent links $i-1$ and i may be expressed as (1). For simplicity, $c = cos$ and $s = sin$.

$$ {}^{i-1}_{i}T = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ s\theta_i c\alpha_{i-1} & c\theta_i c\alpha_{i-1} & -s\alpha_{i-1} & -s\alpha_{i-1}L_i \\ s\theta_i s\alpha_{i-1} & c\theta_i s\alpha_{i-1} & c\alpha_{i-1} & c\alpha_{i-1}L_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1} $$

It allows the global coordinate system from the base to the end effector of the robot to display the transformation matrix as shown in (2).

$$ {}^{B}_{E}T = {}^{B}_{1}T\,{}^{1}_{2}T\,{}^{2}_{E}T \tag{2} $$

2) OBSTACLE DETECTION

According to ISO 15066, it is necessary to determine the exact distance and position between the manipulator and the person in order to ascertain the hazardous stage area of the operator. To do this, an obstacle detection sensor must be used to calculate the exact relative position vector between the manipulator and the obstacle. In the system proposed here, eight FIR sensors are attached to the middle of third link of the three-axis rotary manipulator at 45° intervals, as shown in Fig. 6 (a), measuring the distance of the obstacle to obtain the exact position of the obstacle in HLCAC. It shows the sensor attached to the manipulator to meet the conditions set earlier in Gazebo [31], a simulation program by robot operating system (ROS) [32]. Objects marked in yellow refer to the FIR sensors attached to the manipulator link, and the area marked in yellow indicates the detection range of each sensor. The coordinate system of each sensor attached to the third link can be expressed as Fig. 6 (b). The lower part of the

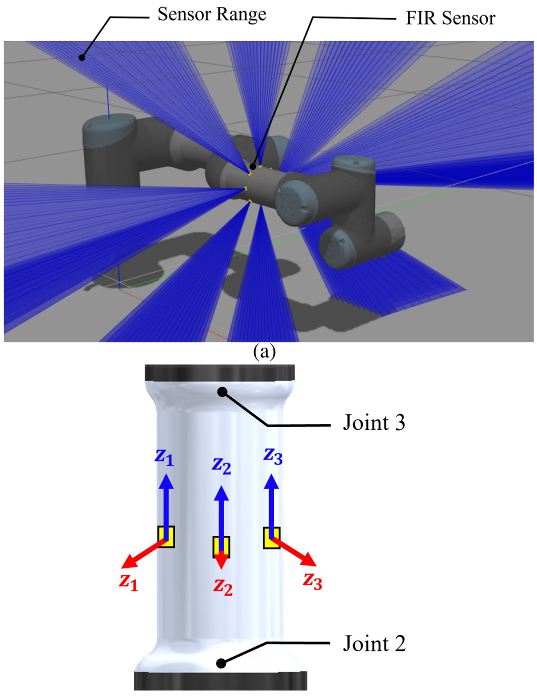


FIGURE 6. Position of the sensors attached to the manipulator (a) Position and detection range of the sensor identified in Gazebo (b) Coordinate system of sensors.

link is the second joint, the upper part is the third joint, and the yellow boxes are the sensors. The z-axis of each sensor is the same direction as the z-axis of third link, and the x-axis is the sensing direction. In order to calculate the exact position of the obstacle detected by each sensor, the transformation matrix from the base joint of the manipulator to each sensor S_i can be expressed as (3) and (4).

$$
{}^{E}_{S_i}T = \begin{bmatrix} c(\frac{\pi}{4} * (i-1)) & -s(\frac{\pi}{4} * (i-1)) & 0 & 0 \\ s(\frac{\pi}{4} * (i-1)) & c(\frac{\pi}{4} * (i-1)) & 0 & 0 \\ 0 \quad 0 & & 1 & -\frac{L_3}{2} \\ 0 \quad 0 & & 0 & 1 \end{bmatrix} \tag{3}
$$

$$
{}^{B}_{S_i}T = {}^{B}_{E}T\,{}^{E}_{S_i}T \tag{4}
$$

In this multi-sensor system, a situation in which an obstacle is detected by two sensors is shown in Fig. 7. The meanings of the variables are shown in Table 2.

In this manner, the distance d_{obs} and angle θ_{obs} between the manipulator and the obstacle can be obtained via (5) and (6), respectively.

$$
\theta_{obs} = \arccos \frac{d_1^2 + (2-\sqrt{2})r^2 - d_2^2}{2d_1(\sqrt{2-\sqrt{2}}r)} \tag{5}
$$

$$
d_{obs} = \sqrt{r^2 + d_1^2 - 2d_1 r \cos\theta_{obs}} - r \tag{6}
$$

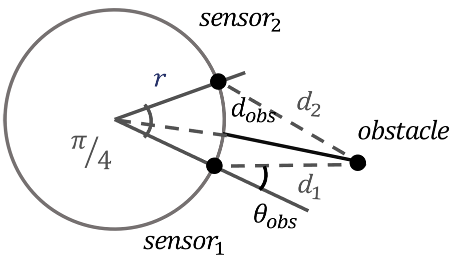


FIGURE 7. One obstacle detected by two FIR sensors.

TABLE 2. Meaning of variables.

Variable	Meaning
r	radius of manipulator link
d_1	distance from $sensor_1$ to obstacles
d_2	distance from $sensor_2$ to obstacles
θ_{obs}	the angle at which the obstacle is located relative to $sensor_1$
d_{obs}	distance from the surface of the manipulator link to the obstacle

B. LLCAC

1) ISO 15066 BASED SAFETY ZONE DEFINITION

The proposed system applied SSM in ISO 15066 based on the distance between persons and obstacles calculated by HLCAC. We set three zones according to the preset unit distance d from the robot's location and each zone's risk levels, as shown in Fig. 8. The unit distance d is 0.1775 m, 10% of the average stride length of people aged 22–30, in [33] who surveyed gait parameters for 164 people aged 5–30. When the robot position was configured as the origin, the area $r \leqq d$ with a radius r was marked in red as the area with a high-risk level. In this area, the joint speed limit of the UR3 manipulator was initialized to $\pm 0.18 rad/s$, which is 10% of $v_{max_{UR}}$. Next, the area where $d < r \leqq 2d$ is where the risk level is intermediate and marked in yellow. In this area, we set the joint speed limit to ± 0.24 rad/s, which is 15% of $v_{max_{UR}}$. Finally, the area $2d < r$ is set as a safe area that can ignore the risk level and is colored in green. In this zone, the manipulator can move without speed limit. We used *Universal Robots ROS driver* provided by Universal Robots to control the joint motor of the UR3, where the speed scaling parameters that control the manipulator's joint speed among input parameters were set to 1, 0.5 and 0.25 from the low-risk order.

2) JACOBIAN MATRIX

To control the speed of a three-axis rotary manipulator in LLCAC, the speed of the manipulator's joint, $\dot{q}$, must be calculated using a jacobian matrix. To find the jacobian matrix, we refer to the coordinate system in Fig. 5 and the D-H parameter in Table 1 to find the regular kinematics of the manipulator by means of (7)-(9).

$$
P_x = c_1[L_3(c_2c_3 + s_2s_3) + L_2c_2] \tag{7}
$$

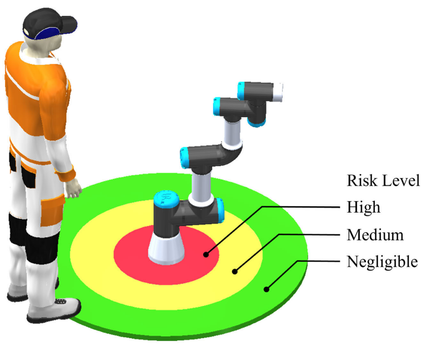


FIGURE 8. Establishment of risk levels according to a distance for human-robot collaboration according to SSM of ISO 15066.

$$P_y = s_1[L_3(c_2c_3 + s_2s_3) + L_2c_2] \tag{8}$$

$$P_z = L_1 + L_2s_2 + L_3(s_2c_3 - c_2s_3) \tag{9}$$

Through differentiation, the velocity on the x, y, and z axis of each joint can be obtained using the jacobian matrix obtained from (10).

$$\dot{X} = \begin{bmatrix} \dot{P}_x \\ \dot{P}_y \\ \dot{P}_z \end{bmatrix} = \begin{bmatrix} \frac{\partial P_x}{\partial \theta_1} & \frac{\partial P_x}{\partial \theta_2} & \frac{\partial P_x}{\partial \theta_3} \\ \frac{\partial P_y}{\partial \theta_1} & \frac{\partial P_y}{\partial \theta_2} & \frac{\partial P_y}{\partial \theta_3} \\ \frac{\partial P_z}{\partial \theta_1} & \frac{\partial P_z}{\partial \theta_2} & \frac{\partial P_z}{\partial \theta_3} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} \tag{10}$$

3) 3-D POTENTIAL FIELD

The basic 2D potential field refers to a robot's obstacle avoidance method derived from the movement of electric charges. This method assumes that the robot is charged with a positive charge on a 2D plane, the obstacle is charged with a positive charge identical to that of the robot, and the destination is a point charged with a negative charge opposite to the robot. This is how the robot finds the route to its destination.

When the coordinate of the robot is $q(x, y)$, the potential energy U(q) acting on the robot as the robot moves to the destination is divided into two components, as shown in (11). $U_{att}(q)$ is the attraction force of the robot to move to the goal, and $U_{rep}(q)$ is the repulsive force of the robot to move away from the obstacle.

$$U(q) = U_{att}(q) + U_{rep}(q) \tag{11}$$

The force, $F(q)$, that moves the robot using the above potential energy is expressed as the sum of the gradient vectors of the potential energy. Additionally, q is the current position of the robot, $\rho_{goal}(q)$ is the distance from the current position q to the destination point as determined by (12), and $\rho(q)$ is the shortest distance from the current position to the obstacle. Also, q_{goal} is the location of the destination point, q_{obs} is the location of the outer point of the obstacle closest to the current location, and ρ_o is a distance constant related to the extent to which the repulsive force potential extends from the obstacle.

$$\rho_{goal}(q) = |q - q_{goal}| \tag{12}$$

U_{att} can be expressed as (13). Here, ξ is a coefficient that determines the size of U_{att} and is directly proportional to the size of U_{att}. The force F_{att} that U_{att} applies to the robot is determined by (14).

$$U_{att}(q) = \frac{1}{2}\xi\rho_{goal}^2(q) \tag{13}$$

$$\begin{aligned} F_{att}(q) &= -\nabla U_{att}(q) \\ &= -\xi\rho_{goal}(q)\nabla\rho_{goal}(q) \end{aligned} \tag{14}$$

Next, U_{rep} can be expressed as (15). At this time, η is a coefficient that determines the size of U_{rep} and is directly proportional to the size of U_{rep}. The force F_{rep} that U_{rep} applies to the robot is expressed as (16).

$$U_{rep}(q) = \begin{cases} \frac{1}{2}\eta(\frac{1}{\rho(q)} - \frac{1}{\rho_o})^2 & \text{if } \rho \le \rho_o \\ 0 & \text{if } \rho > \rho_o \end{cases} \tag{15}$$

$$F_{rep}(q) = \begin{cases} \eta(\frac{1}{\rho(q)} - \frac{1}{\rho_o})\frac{1}{\rho^2(q)}\frac{q - q_{obs}}{\rho(q)} & \text{if } \rho \le \rho_o \\ 0 & \text{if } \rho > \rho_o \end{cases} \tag{16}$$

This potential field-based obstacle avoidance algorithm has a limitation in that it can be applied only in a 2D space. This obviously makes it challenging to apply it as it is to a manipulator operating in a 3D space. To overcome this limitation, the potential energy $U_{xyz}(q)$ in the 3D space is divided into $U_{s_i}(q)$ to regenerate the potential energy on each 2D plane of ith sensor's coordinate. On the existing path of the manipulator accordingly, by combining these potential fields, an obstacle avoidance path in 3D is created. This algorithm is summarized in Algorithm 1.

Algorithm 1 3-D Potential Field System

```
Require initial state($Q_s$), goal state($Q_g$), and distance from
  obstacle($d_{obs}$)
$\overrightarrow{F_{s_i}} = \varnothing$ {Set of a potential field at each 2D cartesian($xz$)
  about $i$th sensor}
$Q_j = \varnothing$ {Set of a $j$th manipulator joint angles}
for $j = 0, 1, \ldots$ do
  if $j = 0$ then
    $Q_j = Q_s$
  else
    while $Q_j = Q_g$ do
      $\overrightarrow{F_{s_i}}(Q_j) = \mathrm{LLCAC}(Q_j)$
      $Q_{j+1} = Q_j + \sum_{i=1}^{8}(s_i^B T \times \delta_{i(x,z)} \frac{\overrightarrow{F_{xz}}(Q_j)}{||\overrightarrow{F_{xz}}(Q_i)||})$
    end while
  end if
  $j = j + 1$
end for
  return $Q_j = 0$
```

III. EXPERIMENT AND RESULT

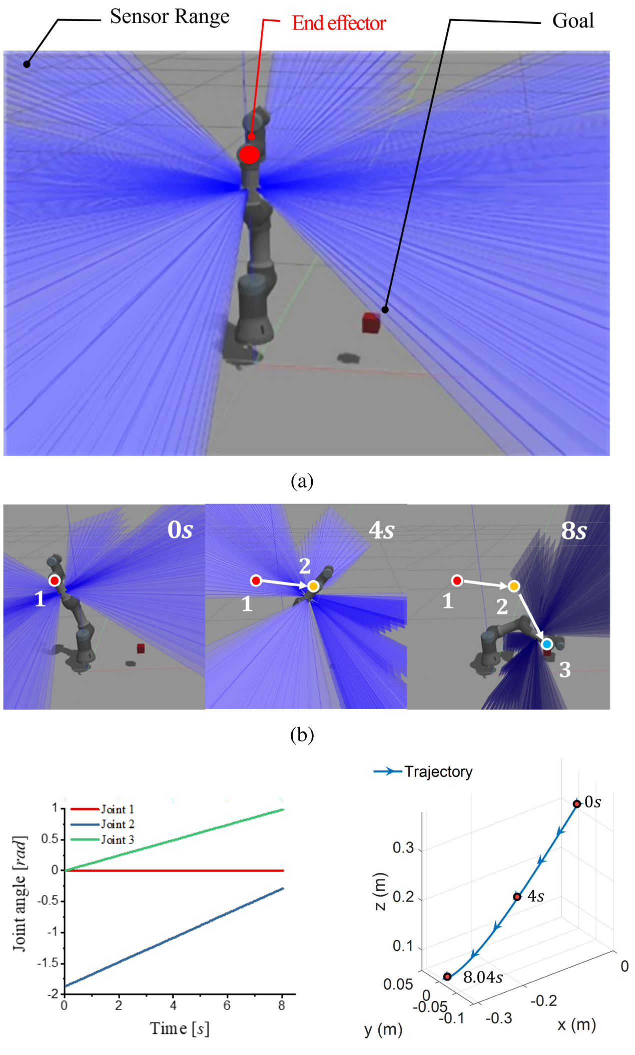
A. SIMULATION

To verify the proposed 3D APF-based obstacle avoidance algorithm in this paper, we conduct experiments to test the ability and time of the manipulator to reach its destination in barrier-free and barrier-free environments. This experiment confirms that the proposed system can construct 3D APF well without blind spots and apply it to obstacle avoidance even if the detection area of the sensor changes as the manipulator moves. To this end, after implementing the UR3 manipulator in the ROS-based Gazebo simulator, eight FIR sensors were attached to the third link of the manipulator at 45° intervals. The FIR sensor used in the experiment is the Terabee Teraranger multiflex type, and the detection range of the sensor is set to fit the sensor's viewing angle of 20°. Fig. 9 (a) shows the simulation environment for the experiment without obstacles. The area marked in yellow is the detection range of the FIR sensor, and the area marked in red is the end-effector of the three-axis rotary manipulator. The goals set in consideration of the working radius of the manipulator are indicated by the red cube.

Fig. 9 (b) shows the generated trajectory when controlling the manipulator's end-effector to move to the goal when there is no obstacle, along with the end-effector's path at four second intervals. In the upper right corner of each scene, the number of seconds is displayed. During this process, the change of each joint angle and the results of analyzing the path of the end-effector as a graph are shown in Fig. 9 (c) and (d), respectively. In Fig. 9 (d), a round marker displays the manipulator at approximately 4-second intervals from where it starts to move to the destination. As a result of the simulation, when there is no obstacle, a time of 8.04s was required for the end-effector of the manipulator to reach the goal, and it could be confirmed that the manipulator moves along the shortest path without a path change.

Next, a simulation with obstacles was conducted. Fig. 10 (a) shows the simulation environment in Gazebo in this case. The obstacle was set as a sphere with a radius of 0.05m, and the green sphere in the figure is the obstacle. In the presence of an obstacle, the result of controlling the manipulator's end-effector to move toward the goal is shown at three second intervals, as shown in Fig. 10 (b). For each scene, the location and path of the end-effector are marked.

As a result of the simulation, it took 16.40s for the manipulator's end-effector to reach the goal in the presence of an obstacle. In addition, it can be confirmed that the manipulator feasibly controls the speed by detecting the obstacle and calculating the danger level through the joint velocity with risk levels of Fig. 10 (c) and the trajectory of the end-effector in Fig. 10 (d). The gray sphere on the graph represents an obstacles. In Fig. 10 (c), we marked the manipulator joint angle over time with a solid line and the joint speed with a dashed line. In the graph, the time when the obstacle is detected and the time when the risk level changes according to the distance from the obstacle are indicated by the pink dashed line, and the risk levels are indicated in green, yellow, and red in the same order as the color in Fig. 8. In addition, Fig. 10 (d) shows the moment the sensor detects an obstacle using a star-shaped marker, and the round marker is used to mark approximately 4-second intervals from where the manipulator starts to move to the destination. It can be confirmed that a new 3D path is regenerated and then utilized by calculating a potential field to avoid an obstacle.



(a)

(b)

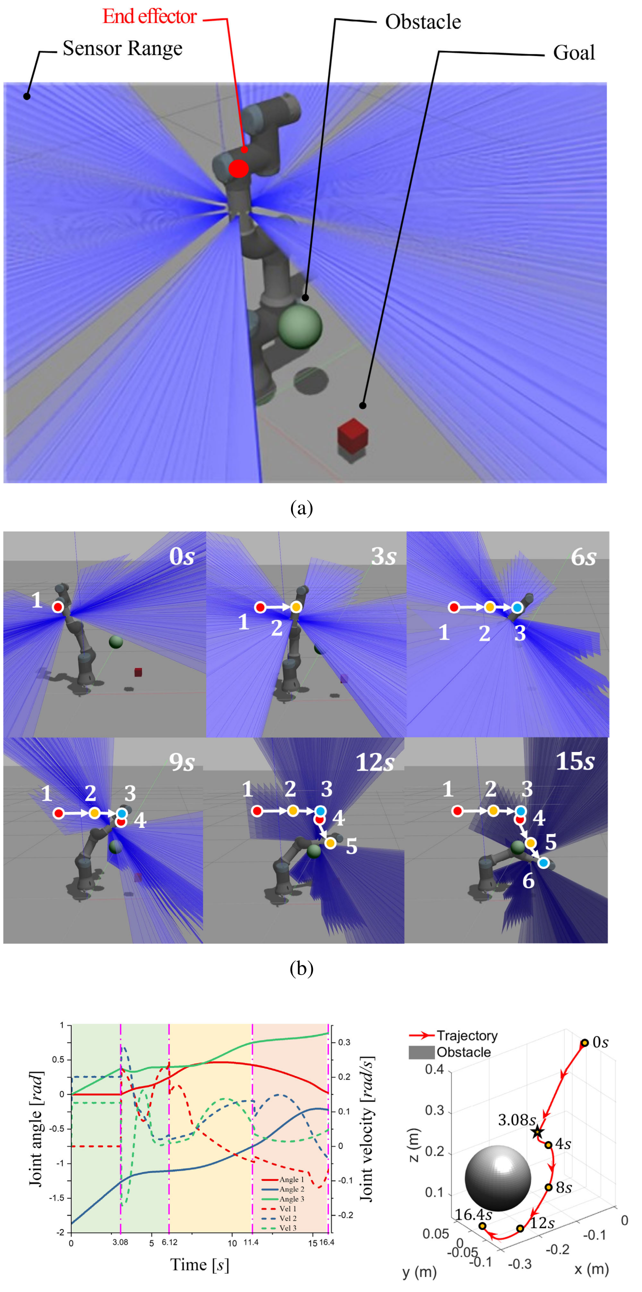
(c)

(d)

FIGURE 9. Simulation without an obstacle (a) Gazebo simulation environment (b) Generated manipulator trajectory (c) The amount of change in each joint angle (d) The trajectory of the end-effector.

B. EXPERIMENT

An experiment was conducted to demonstrate the system by applying the 3D potential field-based obstacle avoidance system to an actual manipulator. To do this, the environment used in the simulation was also used here, as shown in Fig. 11 (a). The third axis of the UR3, indicated by the red circle, was set as the end-effector, and eight FIR sensors were attached



(a)

(b)

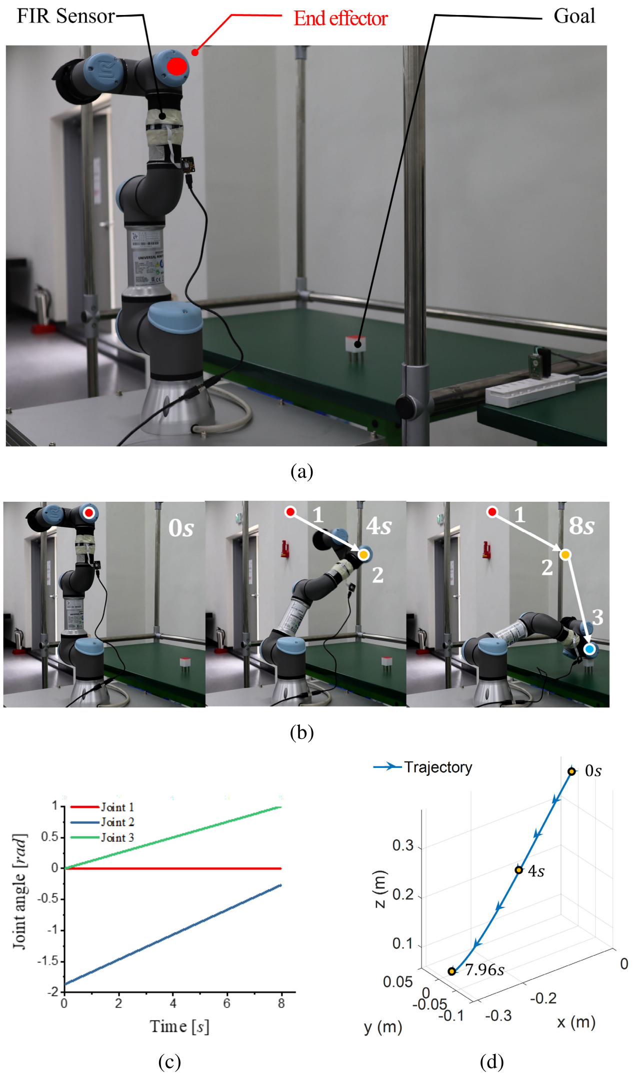
(c)

(d)

FIGURE 10. **Simulation with an obstacle (a) Gazebo simulation environment (b) Generated manipulator trajectory (c) The amount of change in each joint angle and velocity (d) The trajectory of the end-effector.**

to the center of the third link at 45° intervals. In addition, the destination is marked with a white cube.

Fig. 11 (b) shows the result of the manipulator's end-effector motion toward the goal in the absence of an obstacle, along with the end-effector's path at four second intervals. In the upper right corner of each scene, the number of seconds is displayed. At this time, the result of analyzing the path of the end-effector as a graph and the change of each joint angle are shown in Fig. 11 (c) and (d), respectively. In Fig. 11 (d),



(a)

(b)

(c)

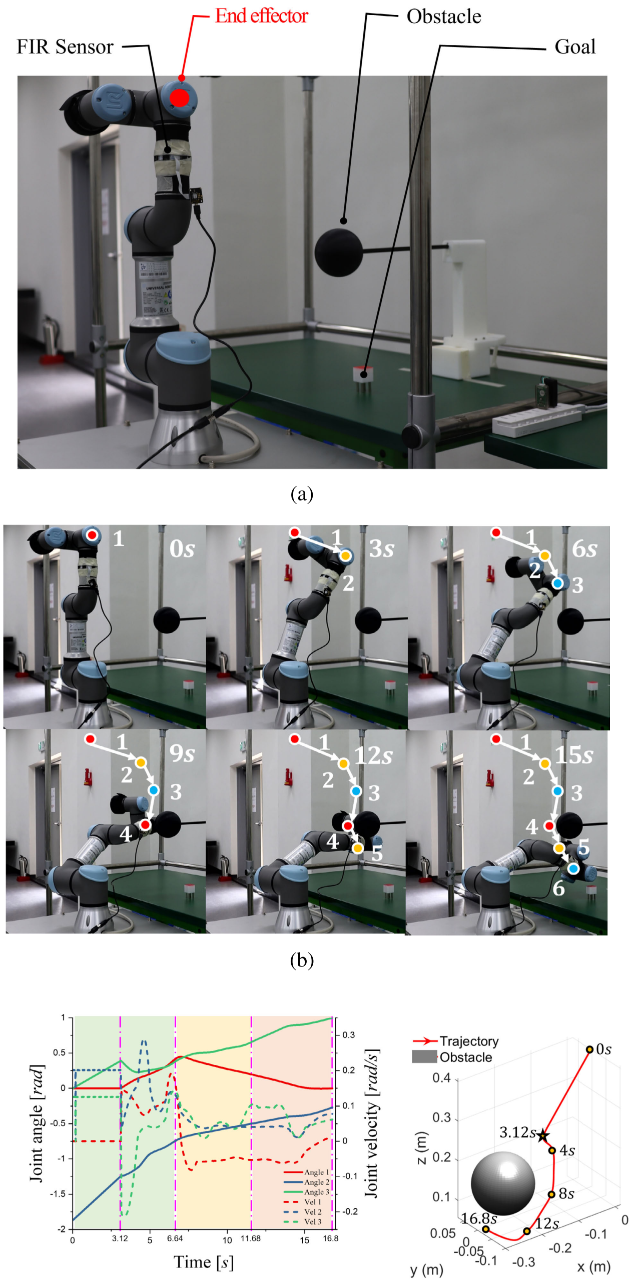
(d)

FIGURE 11. **Experiment without an obstacle (a) Experiment environment (b) Generated manipulator trajectory (c) The amount of change in each joint angle (d) The trajectory of the end-effector.**

a round marker displays the manipulator at approximately 4-second intervals from where it starts to move to the destination. In the actual experiment, in the absence of an obstacle, it took 7.96s for the end-effector of the manipulator to reach the goal. It was also confirmed that it moved along in the shortest path.

Next, the experiment was conducted without obstacles. The experimental environment was configured as shown in Fig. 12 (a), where the obstacle is indicated by the black sphere, and the radius is set to 0.05m with the same approach used in the simulation.

Fig. 12 (b) shows the result of controlling the manipulator's end-effector to move toward the goal in the presence of an obstacle at three second intervals. In each scene, the location and path of the end-effector are marked. In this experiment, it took 16.8s for the end-effector of the manipulator to reach the goal when there was no obstacle. During this process,



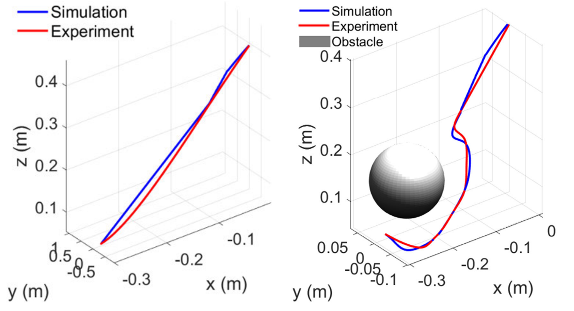
(a)

(b)

(c)

(d)

FIGURE 12. **Experiment with an obstacle (a) Experiment environment (b) Generated manipulator trajectory (c) The amount of change in each joint angle and velocity (d) The trajectory of the end-effector.**



(a)

(b)

FIGURE 13. **Comparison of simulation and experiment results (a) Path when there is no obstacle (b) Path when there is an obstacle.**

it could be confirmed that the manipulator speed was controlled through the calculation of the risk level through the joint velocity with risk levels of Fig. 12 (c). We found that it regenerates and moves a new 3D path to avoid obstacles, as shown in Fig. 12 (d). The gray sphere on the graph represents an obstacles. In Fig. 12 (c), we marked the manipulator joint angle over time with a solid line and the joint speed with a dashed line. In the graph, the time when the obstacle is detected and the time when the risk level changes according to the distance from the obstacle are indicated by the pink washed line, and the risk levels are indicated in green, yellow, and red in the same order as the color in Fig. 8. In addition, Fig. 12 (d) shows the moment the sensor detects an obstacle using a star-shaped marker, and the round marker is used to mark approximately 4-second intervals from where the manipulator starts to move to the destination. The difference in time required between the simulation and experiment was 0.98% in the case without an obstacle and 2.38% with an obstacle. Additionally, the error of the generated path was measured at the point with the most considerable path difference, as shown in Fig. 13 (a) without obstacle and (b) with an obstacle. The gray spheres represent obstacles in these figures. The corresponding errors between the simulation and experiment are $0.011m$ and $0.027m$. The path difference was calculated by the Levenberg-Marquardt method [34].

C. COMPARISON WITH EXISTING WORK

Simulations were conducted to compare the conventional potential field-based obstacle avoidance algorithm with the proposed algorithm. The simulation environment is the same as the obstacle avoidance simulation conducted above, and parameters ξ and η, which determine the attraction and repulsion of the potential field, are set the same as both the conventional and proposed algorithms.

When obstacle avoidance was performed, the avoidance path was generated at 0.01 s intervals using Linear Segment Parabolic Blends (LSPB) [35], and the velocity of all joints of the manipulator was measured at each step. Next, the absolute value of the measured joint velocity was classified and compared according to the distance between the obstacle and the manipulator set in LLCAC as shown in Fig. 14. We performed the same obstacle avoidance and obtained the results that the potential field parameters of the conventional means and the proposed means were the same, but the joint speed of the manipulator was different. The reason is that configuring the potential field is different and is affected by sensor noise. The conventional algorithm was marked in blue box, the proposed algorithm was marked in red box, and the point where the manipulator was located was assumed to be the origin O, and the area with high risk according

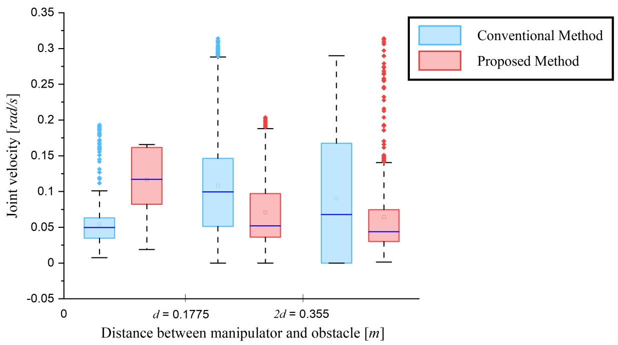


FIGURE 14. Comparison of average manipulator joint speed by distance between obstacle and manipulator while avoiding an obstacle.

to the distance $r \leqq d$, the area $d < r \leqq 2d$, and the area where the risk could be ignored. As a result, existing algorithms moved without joint speed limitations in all zones and did not meet the ISO 15066 SSM criteria. On the other hand, it was verified that the proposed algorithm increases the joint operation speed as the distance between the obstacle and the manipulator increases according to the speed limit preset for each zone. In addition, the time taken to reach the destination while avoiding obstacles was 13.47 s for the existing algorithm, which had no joint speed limit, 17.86% faster than the proposed algorithm's time of 16.40 s.

IV. CONCLUSION AND FUTURE WORK

In this paper, we proposed a 3-D potential field-based obstacle avoidance control system that meets the requirements of ISO 15066 to ensure the safety of workers when people and manipulators exist in the same work space. To confirm the capabilities of the proposed system, simulations and experiments were conducted by applying it to a three-axis rotary manipulator. As a result, in the absence of an obstacle, conventional inverse kinematics-based control was performed. Furthermore, if there is an obstacle, it detects it and generates a three-dimensional potential field accordingly. Based on this, the end effector avoided obstacles along the generated three-axis manipulator obstacle avoidance path while controlling the position. Through this, it was verified that the proposed algorithm successfully avoids obstacles. Next, as a result of comparison with traditional algorithms through simulation, it was affirmed that the proposed algorithm, unlike the existing algorithm, is controlled under a preset joint speed limit according to the distance to the obstacle according to the ISO 15066 SSM criterion.

However, because this system uses a potential field, there is a limitation in that the local-minima may fall when the number of obstacles increases. In addition, as the joint speed limit is applied, the longer work time has increased, which means that there is a limitation of poor work efficiency. In the future, we will solve the problem that potential field based algorithms may fall into local minima when avoiding several obstacles and research how to improve work efficiency while satisfying ISO 15066 SSM criteria.

REFERENCES

[1] S. Wang, D. Li, J. Wan, and C. Zhang, "Implementing smart factory of Industrie 4.0: An outlook," *Int. J. Distrib. Sensor Netw.*, vol. 12, no. 1, Jan. 2016, Art. no. 3159805.

[2] R. Heidel, M. Hankel, U. Döbrich, and M. Hoffmeister, *Basiswissen RAMI 4.0: Referenzarchitekturmodell und Industrie 4.0-Komponente Industrie 4.0*. Beuth Verlag, 2017.

[3] *Robotic Devices–Safety Requirements for Industrial Robots—Part 1: Robots*, I. Robots, ISO(Int. Org. Standardization), 2011.

[4] *Robots and Robotic Devices–Safety Requirements for Industrial Robots—Part 2: Robot Systems and Integration*, Standard 10218-2: 2011, I. ISO, Geneva, Switzerland, 2011.

[5] M. J. Rosenstrauch and J. Krüger, "Safe human-robot-collaboration-introduction and experiment using ISO/TS 15066," in *Proc. 3rd Int. Conf. Control, Autom. Robot. (ICCAR)*, Apr. 2017, pp. 740–744.

[6] N. Tanaka, S. Kobayashi, H. Jin, B. Zhang, and H.-O. Lim, "New passive collision force suppression mechanism for human-friendly robot," in *Proc. Int. Conf. Natural Comput., Fuzzy Syst. Knowl. Discovery*. Springer, 2019, pp. 917–922.

[7] D. Feliu-Talegon, V. Feliu-Batlle, I. Tejado, B. M. Vinagre, and S. H. HosseinNia, "Stable force control and contact transition of a single link flexible robot using a fractional-order controller," *ISA Trans.*, vol. 89, pp. 139–157, Jun. 2019.

[8] Z. Liu, H. Jin, H. Zhang, Y. Liu, Y. Long, X. Liu, and J. Zhao, "A variable stiffness actuator based on second-order lever mechanism and its manipulator integration," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 6999–7005.

[9] G. O. Hoskins, J. Padayachee, and G. Bright, "Human-robot interaction: The safety challenge (an inegrated frame work for human safety)," in *Proc. Southern Afr. Universities Power Eng. Conf./Robot. Mechatronics/Pattern Recognit. Assoc. South Afr. (SAUPEC/RobMech/PRASA)*, Jan. 2019, pp. 74–79.

[10] M. Kong and Y. Bai, "An efficient collision detection algorithm for the dual-robot coordination system," in *Proc. IEEE 3rd Adv. Inf. Technol., Electron. Autom. Control Conf. (IAEAC)*, Oct. 2018, pp. 1533–1537.

[11] S. A. B. Birjandi, J. Kuhn, and S. Haddadin, "Observer-extended direct method for collision monitoring in robot manipulators using proprioception and IMU sensing," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 954–961, Apr. 2020.

[12] S. A. B. Birjandi and S. Haddadin, "Model-adaptive high-speed collision detection for serial-chain robot manipulators," *IEEE Robot. Autom. Lett.*, vol. 5, no. 4, pp. 6544–6551, Oct. 2020.

[13] A. Kabanov and D. Tokarev, "Collision detection and avoidance method for two cooperative robot manipulators," in *Proc. IOP Conf. Mater. Sci. Eng.*, vol. 709, no. 4, Sep. 2020, Art. no. 044021.

[14] F. Ferraguti, C. T. Landi, S. Costi, M. Bonfè, S. Farsoni, C. Secchi, and C. Fantuzzi, "Safety barrier functions and multi-camera tracking for human–robot shared environment," *Robot. Auto. Syst.*, vol. 124, Feb. 2020, Art. no. 103388.

[15] J. Kindle, F. Furrer, T. Novkovic, J. J. Chung, R. Siegwart, and J. Nieto, "Whole-body control of a mobile manipulator using end-to-end reinforcement learning," 2020, *arXiv:2003.02637*.

[16] Z. H. Ang, C. K. Ang, W. H. Lim, L. J. Yu, and M. I. Solihin, "Development of an artificial intelligent approach in adapting the characteristic of polynomial trajectory planning for robot manipulator," *Int. J. Mech. Eng. Robot. Res.*, vol. 9, pp. 408–414, Mar. 2020.

[17] G. Colucci, L. Baglieri, A. Botta, P. Cavallone, and G. Quaglia, "Optimal positioning of mobile manipulators using closed form inverse kinematics," in *Proc. 31st Int. Conf. Robot. Alpe-Adria Danube Region*. Springer, 2022, pp. 184–191.

[18] L. Zhao, J. Zhao, and H. Liu, "Solving the inverse kinematics problem of multiple redundant manipulators with collision avoidance in dynamic environments," *J. Intell. Robotic Syst.*, vol. 101, no. 2, pp. 1–18, Feb. 2021.

[19] J. Zhong, T. Wang, and L. Cheng, "Collision-free path planning for welding manipulator via hybrid algorithm of deep reinforcement learning and inverse kinematics," *Complex Intell. Syst.*, vol. 8, pp. 1–14, Jun. 2021.

[20] L. Zhao, Z. Jiang, Y. Sun, J. Zhao, and H. Liu, "Collision-free kinematics for hyper-redundant manipulators in dynamic scenes using optimal velocity obstacles," *Int. J. Adv. Robotic Syst.*, vol. 18, no. 1, Feb. 2021, Art. no. 17298814211996148.

[21] Y. Tian, X. Zhu, D. Meng, X. Wang, and B. Liang, "An overall configuration planning method of continuum hyper-redundant manipulators based on improved artificial potential field method," *IEEE Robot. Autom. Lett.*, vol. 6, no. 3, pp. 4867–4874, Jul. 2021.

[22] H. Zhang, Y. Zhu, X. Liu, and X. Xu, "Analysis of obstacle avoidance strategy for dual-arm robot based on speed field with improved artificial potential field algorithm," *Electronics*, vol. 10, no. 15, p. 1850, Jul. 2021.

[23] J. Kim, W. Jie, H. H. Kim, and M. C. Lee, "Modified configuration control with potential field for inverse kinematic solution of redundant manipulator," *IEEE/ASME Trans. Mechatronics*, vol. 26, no. 4, pp. 1782–1790, Aug. 2021.

[24] C. Wang, A. V. Savkin, and M. Garratt, "A strategy for safe 3D navigation of non-holonomic robots among moving obstacles," *Robotica*, vol. 36, no. 2, pp. 275–297, 2018.

[25] V. I. Utkin, "Scope of the theory of sliding modes," in *Sliding Modes in Control and Optimization* (Communications and Control Engineering). Springer, 1992, pp. 1–11.

[26] W. Agnew, C. Xie, A. Walsman, O. Murad, Y. Wang, P. Domingos, and S. Srinivasa, "Amodal 3D reconstruction for robotic manipulation via stability and connectivity," in *Proc. Conf. Robot Learn.*, 2021, pp. 1498–1508.

[27] C. Innes and S. Ramamoorthy, "ProbRobScene: A probabilistic specification language for 3D robotic manipulation environments," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2021, pp. 9446–9452.

[28] S. Makita, T. Sasaki, and T. Urakawa, "Offline direct teaching for a robotic manipulator in the computational space," *Int. J. Autom. Technol.*, vol. 15, no. 2, pp. 197–205, Mar. 2021.

[29] P. Svarny, J. Rozlivek, L. Rustler, and M. Hoffmann, "3D collision-force-map for safe human-robot collaboration," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 3829–3835.

[30] *UR3 Technical Specifications*, A. Universal Robots and S. Energivej, Universal Robots, 2016.

[31] N. Koenig and A. Howard, "Design and use paradigms for Gazebo, an open-source multi-robot simulator," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, vol. 3, Oct. 2004, pp. 2149–2154.

[32] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and Y. A. Ng, "ROS: An open-source robot operating system," in *Proc. ICRA Workshop Open Source Softw.*, Kobe, Japan, vol. 3, 2009, p. 5.

[33] S. Voss, C. Zampieri, A. Biskis, N. Armijo, N. Purcell, B. Ouyang, Y. Liu, E. Berry-Kravis, and J. A. O'Keefe, "Normative database of postural sway measures using inertial sensors in typically developing children and young adults," *Gait Posture*, vol. 90, pp. 112–119, Oct. 2021.

[34] Å. Björck, *Numerical Methods for Least Squares Problems*. Philadelphia, PA, USA: SIAM, 1996.

[35] D. Constantinescu and E. A. Croft, "Smooth and time-optimal trajectory planning for industrial manipulators along specified paths," *J. Robotic Syst.*, vol. 17, no. 5, pp. 233–249, 2000.

YEON KANG received the B.S. and M.S. degrees in electronics engineering from Kyung Hee University, Yongin, South Korea, in 2018 and 2020, respectively. He is currently pursuing the Ph.D. degree in robotics and mechatronics engineering with the Daegu Gyeongbuk Institute of Science and Technology (DGIST), Daegu, South Korea.

His research interests include human–robot interaction and industrial robots.

DONGHAN KIM received the B.S., M.S.E., and Ph.D. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology.

He was a Postdoctoral Fellow at the University of Illinois at Urbana–Champaign, in 2003. He has been working as a Professor with the Department of Electrical Engineering (Age Service-Tech), Kyung Hee University, South Korea, since 2007, where he founded the HRI Laboratory, which performs research in human–robot interaction, service robotics, cooperative robotics, and multi-robot systems. Prior to his position at Kyung Hee University, he worked for three years in industrial and commercial hardware development as a Senior Engineer at Samsung Electronics. During his time there, he developed and led numerous projects on mobile phones. His current research interest includes the area of multi-agent robot systems.

Prof. Kim serves on the boards on International Robot-Soccer Committee, Intelligent Robot Standards Forum, Institute of Control, Robotics and Systems, and International Standard Organization.

DONGWON YUN (Senior Member, IEEE) received the B.S. degree in mechanical engineering from Pusan National University, Busan, South Korea, in 2002, the M.S. degree in mechatronics engineering from the Gwangju Institute of Science and Technology (GIST), Gwangju, South Korea, in 2004, and the Ph.D. degree in mechanical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2013.

He was with the Department of Electrical Engineering and Computer Science, University of California at Berkeley, Berkeley, as a Postdoctoral Researcher, from 2014 to 2015. He was a Researcher at the Agency for Defense Development (ADD), in 2004, and a Senior Researcher with the Korea Institute of Machinery and Materials (KIMM), from 2005 to 2016. He joined the Department of Robotics and Mechatronics Engineering, Daegu Gyeongbuk Institute of Science and Technology (DGIST), in 2016, as an Assistant Professor. His research interests include bio-mimetic robot systems, industrial robot systems and mechatronics, soft robotics, and sensors and actuators.