International Journal of Control, Automation and Systems 18(8) (2020) 2095-2107
http://dx.doi.org/10.1007/s12555-019-0076-7
ISSN:1598-6446 eISSN:2005-4092
http://www.springer.com/12555

Trajectory Planning with Collision Avoidance for Redundant Robots Using Jacobian and Artificial Potential Field-based Real-time Inverse Kinematics

Sun-Oh Park, Min Cheol Lee*, and Jaehyung Kim

Abstract: This study proposes an algorithm for combining the Jacobian-based numerical approach with a modified potential field to solve real-time inverse kinematics and path planning problems for redundant robots in unknown environments. With an increase in the degree of freedom (DOF) of the manipulator, however, the problems in real-time inverse kinematics become more difficult to solve. Although the analytical and geometrical inverse kinematics approach can obtain the exact solution, it is considerably difficult to solve as the DOF increases, and it necessitates recalculations whenever the robot arm DOF or Denavit–Hartenberg (D–H) parameters change. In contrast, the numerical method, particularly the Jacobian-based numerical method, can easily solve inverse kinematics irrespective of the aforementioned changes including those in the robot shape. The latter method, however, is not employed in path planning for collision avoidance, and it presents real-time calculation problems. This study accordingly proposes the Jacobian-based numerical approach with a modified potential field method that can realize real-time calculations of inverse kinematics and path planning with collision avoidance irrespective of whether the case is redundant or non-redundant. To achieve this goal, the use of a judgment matrix is proposed for obstacle condition identification based on the obstacle boundary definition; an approach for avoiding the local minimum is also proposed. After the obstacle avoidance path is generated, a trajectory plan that follows the path and avoids the obstacle is designed. Finally, the proposed method is evaluated by implementing a motion planning simulation of a 7-DOF manipulator, and an experiment is performed on a 7-DOF real robot.

Keywords: Inverse kinematics, Jacobian, obstacle avoidance, path planning, potential field.

1. INTRODUCTION

In numerous fields in the industry where machines are employed, robots are typically guided by human users to their desired positions and paths either directly or remotely. Robot information pertaining to the desired angles and natural path to reach a target position is provided to allow the human operator to directly move the robot arm [1–3].

The position and orientation of a robot must be planned to control it such that it can move towards the target. Accordingly, two cases are considered: each joint angle (θ) of the robot is known, and only the target position and orientation are given. The first case merely adapts the use of forward kinematics but the second necessitates certain analytical approaches [4,5]. The process of finding the angle, θ, when the target point is given is defined as inverse kinematics [6]. This technique also has various solution approaches such as analytical and numerical methods [5–8]. The use of an analytical method offers the following advantages: solutions are always accurate, and calculation times are short. Its disadvantages, however, include the following: it is more difficult to solve inverse kinematics as the degree of freedom (DOF) increases, and recalculations are necessary whenever the Denavit–Hartenberg (D–H) parameters are changed. On the other hand, the numerical method can easily adapt to the system even when the D–H parameters change and the DOF increases [9–14]. The latter method, however, encounters a problem when performing real-time calculation and solution approximation.

Numerical inverse kinematics does not guarantee real-

Manuscript received January 31, 2019; revised October 1, 2019; accepted December 1, 2019. Recommended by Associate Editor Kang-Hyun Jo under the direction of Editor-in-Chief Keum-Shik Hong. This work was supported by the Technology Innovation Program(10073147, Development of Robot Manipulation Technology by Using Artificial Intelligence) funded By the Ministry of Trade, Industry & Energy(MOTIE, Korea). This research was also funded and conducted under "the Competency Development Program for Industry Specialists" of the Korean Ministry of Trade, Industry and Energy (MOTIE), operated by Korea Institute for Advancement of Technology (KIAT) (No. P0008473, The development of high skilled and innovative manpower to lead the Innovation based on Robot).

Sun-Oh Park, Min Cheol Lee, and Jaehyung Kim are with the School of Mechanical Engineering, Pusan National University, 2, Busandaehak-ro 63beon-gil, Geumjeong-gu, Busan, Korea (e-mails: {sunoh425, mclee}@pusan.ac.kr, 11045kjh@naver.com).
* Corresponding author.

time calculation, obstacle avoidance, or path planning [15–17]. To solve real-time problems and implement path planning in unknown environments, the use of the potential field method concept is necessary [18–29]. This method, however, encounters a local minimum problem because it is typically employed only for planning the paths of mobile robots and simply considers attractive and repulsive forces [30–33]. Moreover, the method is difficult to use in a robot with a high DOF.

To solve the foregoing problems, a real-time inverse kinematics approach for redundant robots that uses a modified potential field with the Jacobian-based numerical method is proposed. This numerical method is only employed to calculate the initial and target positions and orientations. Thereafter, the path between the initial and desired points is generated by the modified potential field. The numerical calculation time is inconsequential with respect to computational time. This is because numerical calculation is only employed once to obtain each joint angle at the initial and final positions and orientations of the robot's end effector. The modified potential field concept considers not only attractive and repulsive forces but also the judgment related to obstacle identification and condition, the addition of virtual boundaries, and the recognition of and escape from the local minimum. After calculating the obstacle avoidance path, a trajectory plan is generated to follow this path. Finally, simulations and experiments are conducted to evaluate the proposed algorithm.

The main objective of this research is to determine each of the robot's joint angle and to avoid collisions as the robot moves from the initial position to the target position in real time. In unknown environments, this is achieved using the proposed method instead of human guidance. The proposed approach can derive the desired angle of each joint in real time without the necessity of human manipulation, which is typically implemented among industrial robots.

The remainder of this paper is organized as follows: In Section 2, the proposed concept is explained. Section 3 discusses the numerically based inverse kinematics. Section 4, which contains a substantial portion of the paper, focuses on real-time obstacle avoidance. In Section 5, trajectory planning is considered. In Section 6, the simulations and experiments are presented. Section 7 summarizes the conclusions.

2. CONCEPT OF APPROCHING

This study aims to formulate an approach that can automatically plan a path for the robot under an unknown environment when the desired position is given. First, each joint angle of the 7-DOF robot is calculated using inverse kinematics. Second, a path is planned by the modified potential field method when the initial and final angles of each joint are known. To solve the foregoing path planning problems, the artificial potential field method utilizes the concept of attractive and repulsive forces. The following should also be considered: the relationship between the material points and rigid body to recognize the local minimum; the escape from the local minimum; the boundary condition to prevent collision; the condition for obstacle judgment. Finally, trajectory planning is considered after path planning. The former should include the planned route to avoid obstacles. Accordingly, the apportioning of time from start to finish until the target point is reached should also be considered. Fig. 1 shows the overall concept flowchart of the 7-DOF robot motion planning. The proposed algorithm can also be applied regardless of the DOF and shape.

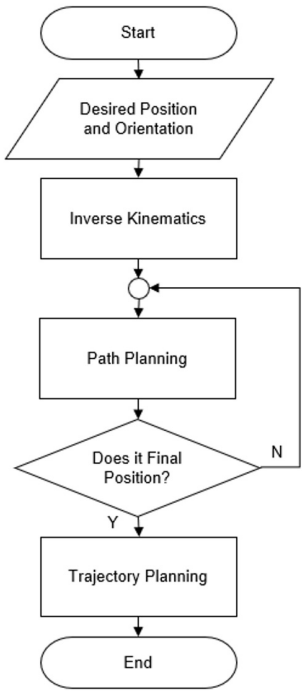

Fig. 1. Flowchart of 7-DOF robot motion planning.

3. INVERSE KINEMATICS

3.1. 7-DOF robot D-H parameter

Fig. 2 shows the real and concept images of the 7-DOF robot. The red circle indicates the origin $(0,0,0)$, and Table 1 summarizes the D–H parameters from this origin. The simulations and experiments are executed by the 7-DOF model.

3.2. Jacobian-based inverse kinematics algorithm

The Jacobian based numerical inverse kinematics method flowchart for obtaining the desired target position angle is shown in Fig. 3. The initial position and orientation are obtained as usual, and each joint angle at the desired position and orientation is calculated.

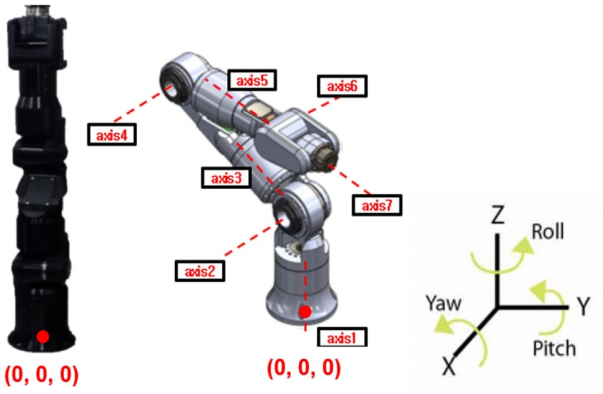

Fig. 2. 7-DOF robot and conceptual diagram.

Table 1. 7-DOF D-H parameters.

Joint	θ (°)	d (mm)	a (mm)	α (°)
1	θ_1*	278	0	-90
2	θ_2*	0	0	90
3	θ_3*	425	0	-90
4	θ_4*	0	0	90
5	θ_5*	425	0	-90
6	θ_6*	0	0	90
7	θ_7*	72	0	0

*: variable rotation angle

In the above, G, P, J, m, n, and B denote the target position, current position, Jacobian, arbitrary constant, DOF, and arbitrary $n \times 1$ value, respectively.

3.3. Jacobian-based numerical method

The Jacobian-based numerical method is typically calculated using the following equations.

$$\theta_{i+1} = \theta_i + d\theta, \tag{1}$$

$$d\theta = J^{-1} dP. \tag{2}$$

In the 7-DOF case, the Jacobian can be expressed as

$$J_{6\times7} = \begin{bmatrix} z_0 \times (o_7 - o_0) & \cdots & z_6 \times (o_7 - o_6) \\ z_0 & \cdots & z_6 \end{bmatrix}, \tag{3}$$

$$z_0 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \ \cdots, \ z_6 = R_5^6 z_5 = R_0^1 \cdots R_5^6 z_0, \tag{4}$$

where P_0^n denotes the coordinates of n in 0; O_n is the distance between the origin and point n.

The Jacobian is not a square matrix, hence a pseudo inverse matrix is utilized to obtain the inverse Jacobian, as follows:

$$J_{6\times7}\left[J_{7\times6}^T \left(J_{6\times7} J_{7\times6}^T\right)^{-1}\right] = J_{6\times7} J_{7\times6}^* = I, \tag{5}$$

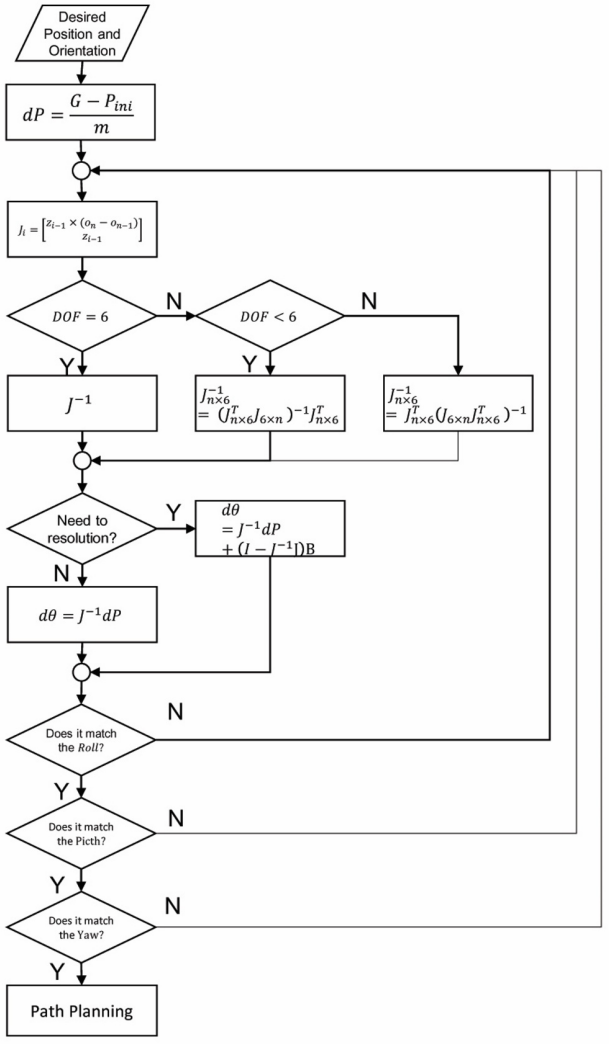

Fig. 3. Flowchart of inverse kinematic algorithm.

where $J_{7\times6}^*$ is the same as $J_{7\times6}^T (J_{6\times7} J_{7\times6}^T)^{-1}$; however pseudo matrix $J_{7\times6}^*$ will be expressed as J^{-1} to appear more intuitive.

Positions x, y, and z do not affect each other but the orientations $\emptyset_x$, $\emptyset_y$, and $\emptyset_z$ yield a different result when the rotation order is changed. The orientation's rotation order must therefore be separated during calculation.

$$\begin{bmatrix} x_s \\ y_s \\ z_s \\ \emptyset_{x,s} \\ \emptyset_{y,s} \\ \emptyset_{z,s} \end{bmatrix} \underset{\rightarrow}{(Roll)} \begin{bmatrix} x_G \\ y_G \\ z_G \\ \emptyset_{x,s} \\ \emptyset_{y,s} \\ \emptyset_{z,G} \end{bmatrix} \underset{\rightarrow}{(Pitch)} \begin{bmatrix} x_G \\ y_G \\ z_G \\ \emptyset_{x,s} \\ \emptyset_{y,G} \\ \emptyset_{z,G} \end{bmatrix} \underset{\rightarrow}{(Yaw)} \begin{bmatrix} x_G \\ y_G \\ z_G \\ \emptyset_{x,G} \\ \emptyset_{y,G} \\ \emptyset_{z,G} \end{bmatrix}. \tag{6}$$

The first step in inverse kinematics is therefore completed and repeated in the (n+1) step as follows:

$$\theta(1) \rightarrow J(1) \rightarrow J^{-1}(1)$$

$$d\theta(1) = J^{-1}(1)dP$$
$$\theta(2) = \theta(1) + d\theta(1)$$
$$\theta(2) \to J(2) \to J^{-1}(2) \to \theta(3) \to \cdots \to \theta(n+1).$$

3.4. Resolution approach

When the 7-DOF robot remains in the null space, the robot can approach the null point from various angles This means that the robot can identify another path to reach the desired position.

The resolution method that uses the Jacobian-based inverse kinematics can derive the various angles of each joint at a desired end effector position and orientation. The numerical method employed is the same in robots with any DOF. To utilize a merit of the redundant robot, the calculation of $d\theta$ in (2) is redefined as follows:

$$d\theta = J^{-1}dP + (I - J^{-1}J)B. \tag{7}$$

When a pose of the robot causes a collision with an obstacle or the rotation angle of the robot exceeds the limiting angle, B is randomly inputted.

4. PATH PLANNING

4.1. Real-time inverse kinematics with potential field

The initial and desired angles are obtained by the Jacobian-based inverse kinematics, as discussed in Section 3. Thereafter, a path is generated by the proposed modified potential field. The numerical inverse kinematics necessitates some computational times which cannot satisfy real-time calculation. The artificial potential field method, however, only requires the calculation of forward kinematics and not inverse kinematics. The robot path computation is thus guaranteed for real-time calculation because the forward kinematics does not require considerable computational time. This means that the angles of each joint of the redundant robot can be calculated in real time by the applied host processing system. The path planning flowchart of obtaining the target position angle is presented in Fig. 4.

In Fig. 4, F_{att}, F_{rep}, and F_{vir} are the attractive, repulsive, and virtual forces, respectively; P and P_i are the current and initial positions, respectively; T, O, and C are the torque, obstacle, and a coefficient, respectively; BC, LMC, and BC_LMC are the boundary condition, local minimum condition, and boundary condition of the local minimum condition, respectively. In the figure, the proposed potential field algorithm is employed. In this algorithm, the given repulsive force varies according to the distance of the boundary from the obstacle surface, and the virtual force is generated at a local minimum state to avoid such the state. Detailed explanations are presented in Sections 4.6 to 4.8.

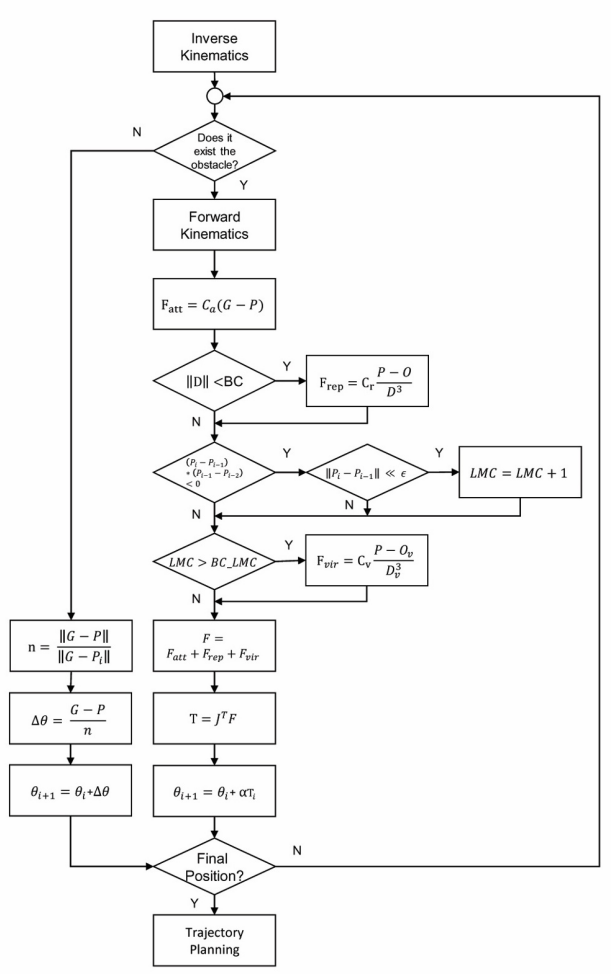

Fig. 4. Flowchart of path planning algorithm using modified potential field.

4.2. Midpoint concept

Although the robot arm is a rigid body, it can be numerically calculated by shaping it using material points spaced at a certain interval. If the spacing of these material points is reduced, the shape approaches that of a rigid body; however, this increases the amount of calculation. If the interval between particle points is overly wide, then it cannot be considered as a rigid body. The number of material points in a rigid body, therefore, varies according to the relationship between obstacle size and rigid body length. The number of material points in a rigid body is expressed as follows:

$$\frac{Length\ of\ Link}{Size\ of\ Obstacle} \le Number\ of\ Mid-points, \tag{8}$$

$$\text{Mid\_Point} = \frac{1}{Number\ of\ Mid-points}. \tag{9}$$

Number of Mid-points in (8) represents the number of material points in each rigid body; Mid_Point in (9) is the relative distance of material points with constant spacing

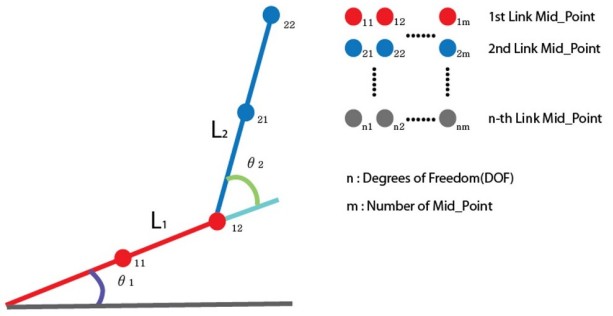

Fig. 5. Concept of Mid_Point on rigid body.

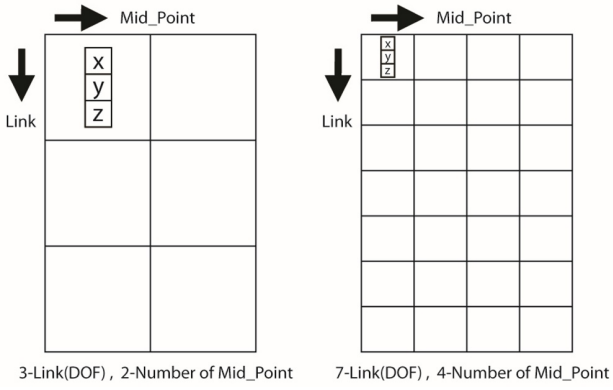

Fig. 6. Concept of Mid_Point structure.

on a rigid body.

For example, if the number of midpoints presented in Fig. 5 is 2, then Mid_Point is [0.5 1], which represents the middle and end points of the rigid body, respectively.

In Fig. 5, a 2-DOF rigid body with 2-Mid_Point numbers are shown. It can be expressed, however, as n-DOF and m-number of midpoints on the rigid body. In Fig. 6, the structures on the left and right sides are 3-DOF with 2-midpoints and 7-DOF with 4-midpoints, respectively.

Each structure space can include an $n \times m$ matrix depending on the contents.

The shape is changed when the rigid body is rotated. Hence the constraint condition is expressed as follows:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_i = Link_{i-1} + Link_i \cdot mid\_point. \tag{10}$$

The position of each material point is part of the top of the rigid link. Therefore, (10) can represent an approximate rigid body shape.

4.3. Obstacle identification condition

In this section, the presence or absence of obstacles in the robot path and the possibility of robot path obstruction are discussed.

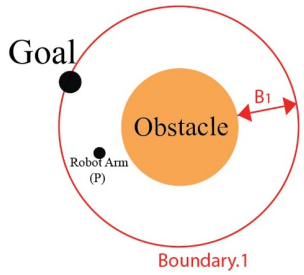

Fig. 7. Target and robot in obstacle boundary.

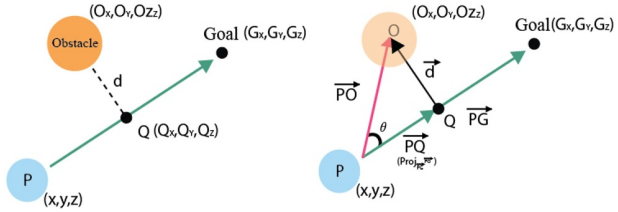

Fig. 8. Concept of obstacle, robot and target position.

Consider the case where an obstacle exists but not on the straight path between the robot arm and target position in Fig. 7. The obstacle does not obstruct the robot path but affects the robot arm because the target position and arm are located within the obstacle boundary. To eliminate the influence of this type of obstacles, the use of a condition-determining matrix that indicates the possibility of obstruction is considered.

In Fig. 8, if the robot's current position, target position, and obstacle position are given, there exists a distance (d) that is the shortest between vector $\overrightarrow{PG}$ and the obstacle. Vector $\overrightarrow{PQ}$ is defined as follows:

$$\overrightarrow{PQ} = \left\|\overrightarrow{PO}\right\| \cos\theta \frac{\overrightarrow{PG}}{\left\|\overrightarrow{PG}\right\|} = \left(\frac{\overrightarrow{PG}\cdot\overrightarrow{PO}}{\left\|\overrightarrow{PG}\right\|\left\|\overrightarrow{PG}\right\|}\right)\overrightarrow{PG}, \tag{11}$$

where $\overrightarrow{PG}\cdot\overrightarrow{PO} = \|\overrightarrow{PG}\|\|\overrightarrow{PO}\|cos\theta$.

The shortest distance (d) from the obstacle to $\overrightarrow{PQ}$ is defined as follows:

$$d = \|\overrightarrow{PO} - \overrightarrow{PQ}\|. \tag{12}$$

The value of parameter t ($0 \le t \le 1$) from P to point Q on vector $\overrightarrow{PQ}$ is defined as

$$t = \frac{\left\|\overrightarrow{PQ}\right\|}{\|PG\|}\frac{\overrightarrow{PG}\cdot\overrightarrow{PQ}}{\|PG\|\,\|PQ\|} = \frac{\overrightarrow{PG}\cdot\overrightarrow{PQ}}{\|PG\|\,\|PG\|}, \tag{13}$$

where $\frac{\left\|\overrightarrow{PQ}\right\|}{\|PG\|}$ is the magnitude of t, and $\frac{\overrightarrow{PG}\cdot\overrightarrow{PQ}}{\|PG\|\|PQ\|}$ is the unit vector of t.

If t is 0, then it represents the robot's current position in the vector. If t is 1, then it represents the target position. If the shortest distance (d) is smaller than the radius of the obstacle (r), then the robot may collide with the obstacle; however, a collision can occur only when the shortest distance is between the robot's current position and obstacle position. Collision is not possible unless these two possibilities are satisfied. The values of these two possibilities can, therefore, be obtained using the vector properties as follows:

$$d_{Judge} = \begin{cases} 1, & (d \le r), \\ 0, & (d > r), \end{cases} \tag{14}$$

$$t_{judge} = \begin{cases} 1, & (0 \le t \le 1), \\ 0, & (t < 0,\ t > 1), \end{cases} \tag{15}$$

$$\lambda = d_{Judge} \cdot t_{judge} = 1. \tag{16}$$

The discrimination matrix is calculated for each mid point, and the obstacle judgment condition is classified according to each part. If λ is 0 at all midpoints, then there is no possibility of collision with any of the rigid bodies in the robot.

4.4. Path planning without obstacle

If it is determined that there is no obstacle based on the discrimination matrix, a path is generated using the current and target positions as follows:

$$\mathrm{n} = \mathrm{C}\frac{\|G-P\|}{\|G-P_i\|}, \tag{17}$$

$$\mathrm{d}\theta = \frac{G-P}{n}, \tag{18}$$

$$\theta_{i+1} = \theta_i + d\theta. \tag{19}$$

The above method is called non-obstacle mode in this paper. Since the presence or absence of obstacles can differ according to the position of the robot, a method of varying the robot position depending on the obstacle position is used rather than keeping $\mathrm{d}\theta$ constant.

4.5. Artificial potential field

If an obstacle exists in the discrimination matrix, an artificial potential field is employed to create an obstacle avoidance path [22–28, 31, 32]. The concept of generating potential and repulsive forces is presented in Fig. 9.

The attractive and repulsive forces are defined as

$$\overrightarrow{F_{att}} = \begin{cases} C_{att}(G-P), & (G-P) \le D_{lim}, \\ F_{max,att}, & (G-P) > D_{lim}, \end{cases} \tag{20}$$

$$\overrightarrow{F_{rep}} = \begin{cases} C_{rep}\dfrac{P-O}{D^3}, & D \le 1, \\ 0, & D > 1, \end{cases} \tag{21}$$

$$\mathrm{D} = \frac{\|P-O\|}{BC}, \tag{22}$$

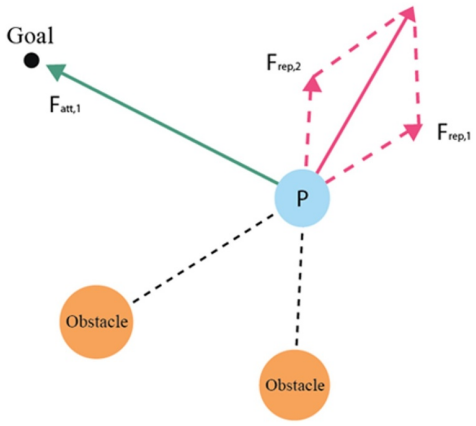

Fig. 9. Potential field with material points.

where D_{lim} is the limited distance, and D is the dimensionless distance.

4.6. Potential field in a rigid body

In Fig. 10, to express each material point as a connected body using the constraint condition, the midpoint concept is applied, and the shape of the body is realized by calculating the material point at a certain distance of each static position.

Considering the attractive and repulsive forces for each m-mid_point of the robot arm with n-DOF, each point considered is represented by the structure presented in Figs. 11 and 12. Each Jacobian (J) has a $6 \times n$ matrix, and each force (F) is a 6×1 matrix; however, rows 4-6 are zero because the material point has no rotational force. Each structure is calculated between the same row and column, and the relationship between the force and torque is obtained using the principle of virtual work in order to convert the force exerted on the material point into torque.

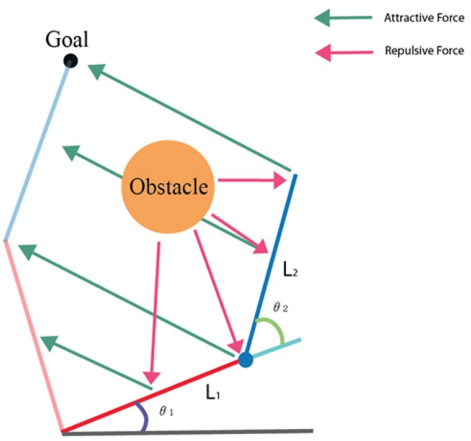

Fig. 10. Potential field with rigid body.

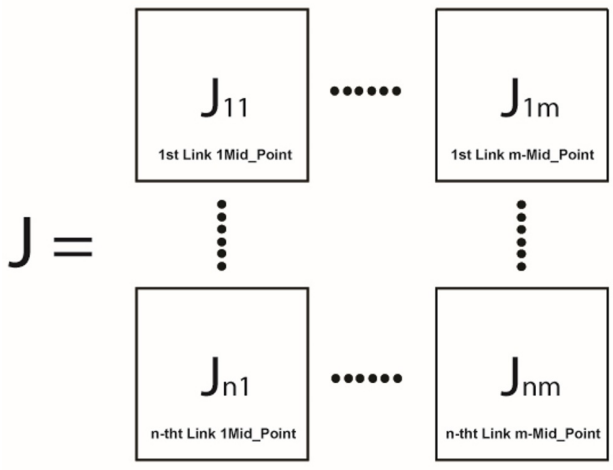

Fig. 11. Concept of Jacobian structure.

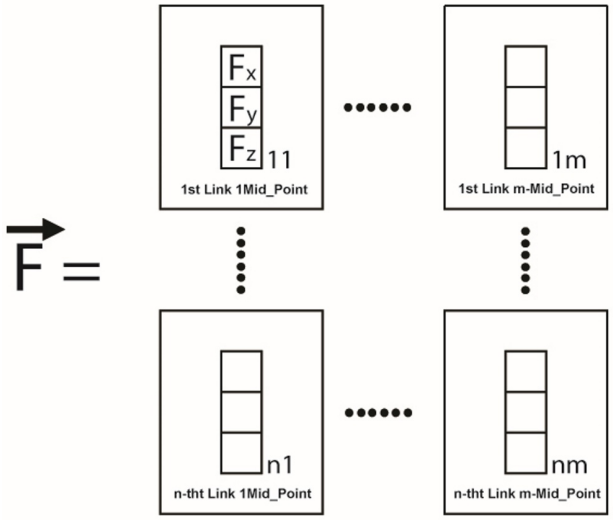

Fig. 12. Concept of force structure.

By applying virtual work at each point, the torque is obtained as follows:

$$\mathrm{T} = \sum_{i=1}^{n}\sum_{j=1}^{m} J_{i,j}^{T} F_{i,j}, \tag{23}$$

$$\theta_{i+1} = \theta_i + C\frac{T}{\|T\|}. \tag{24}$$

Simply put, the force is calculated using the attractive and repulsive forces at each midpoint. It is then converted into torque, and the angle of the next path is subsequently calculated.

4.7. Boundary concept to avoid collision

When an obstacle exists, the following boundaries are necessary—boundary 0: (obstacle radius) to prevent the situation where the obstacle always collides with the robot; boundary 1: to improve collision avoidance stability; boundary 2: to prevent sudden changes in speed and acceleration. By designing all the three boundaries as shown in Fig. 13, the sudden changes in the velocity and acceleration of the robot are precluded and stabilized relative to the obstacle boundary.

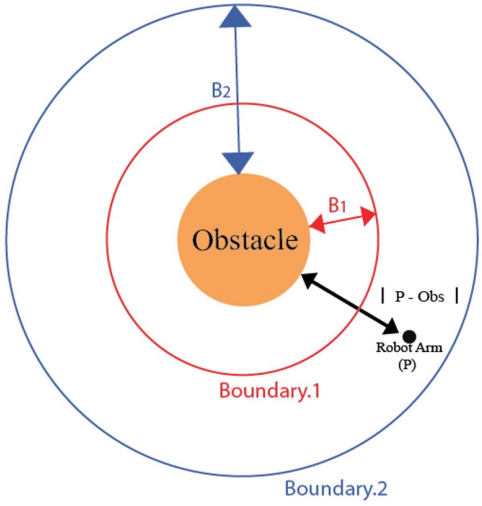

Fig. 13. Concept of boundary 0, 1, and 2.

The repulsive force according to each boundary is as follows:

Boundary. 0

The obstacle radius is equal to the size of the actual obstacle; hence, collision has to be always prevented regardless of the discrimination matrix and generates a strong repulsive force:

$$\overrightarrow{F_{rep,0}} = \begin{cases} C_{rep,0}\dfrac{\overrightarrow{P-O}}{D_0^3}, & D_0 \le 1, \\ 0, & D_0 > 1, \end{cases} \quad \mathrm{D}_0 = \frac{\|P-O\|}{r}, \tag{25}$$

where $C_{rep,0}$ is the arbitrary coefficient in the case of boundary 0 to adjust the repulsive force magnitude..

Boundary. 1

Collision avoidance generates a boundary value greater than the obstacle to increase stability. A strong repulsive force works but not if the discrimination condition (λ) is 0:

$$\overrightarrow{F_{rep,1}} = \begin{cases} C_{rep,1}\dfrac{\overrightarrow{P-O}}{D_1^3}\lambda, & D_1 \le 1, \\ 0, & D_1 > 1, \end{cases} \quad \mathrm{D}_1 = \frac{\|P-O\|}{B_1}, \tag{26}$$

where $C_{rep,1}$ is an arbitrary coefficient in the case of boundary 1 to adjust the repulsive force magnitude.

Boundary. 2

Using the current and boundary positions of the robot, it is found that the coefficient increases as the obstacle is approached, and the repulsive force is generated in proportion to the attractive force. An ideal obstacle avoidance path will avoid obstacles in boundary 2.

$$C_{rep,2} = \frac{B_2 - \|P-O\|}{2(B_2 - B_1)}, \tag{27}$$

$$\overrightarrow{F_{rep,2}} = \begin{cases} C_{rep,2}\,\|F_{att}\|\,\dfrac{\overrightarrow{P-O}}{\|P-O\|}\lambda, & \|P-O\| \le B_2, \\ 0, & \|P-O\| > B_2. \end{cases} \tag{28}$$

4.8. Local minimum and virtual source

The material point of 1-DOF is in the local minimum state. That is, although the robot has not reached the target, the resultant force is zero. It can thus be expressed as follows:

$$F = F_{att} + F_{rep} = 0. \tag{29}$$

In the case of a rigid body with two or more degrees of freedom, the resultant force at one point may be zero but usually not zero at another point. In the case of a multi-degree-of-freedom rigid body, even if the resultant force at one material point is diminished, the resultant force at another material point continues to exist. All rigid bodies therefore continue to move, and the rigid body with zero resultant force also changes. The local minimum in the body therefore becomes a shape that vibrates with respect to a specific position. A special case of the above is when all resultant torques are made zero by each resultant force.

In Fig. 14, the rigid body shown in blue is a special section in which the sum of all torques is zero, and the direction of the torque of the robot is reversed with respect to this position. In the local minimum, there exists a phenomenon where the direction of the vector is reversed because of the action of such attraction and repulsion. If the movement of the robot arm is smaller than the microdisplacement (ε) because of this phenomenon, then it is regarded as a stagnation point of the robot arm.

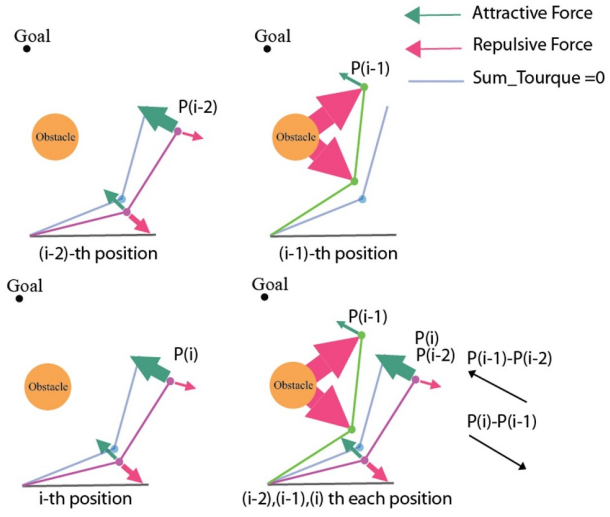

Fig. 14. Rigid body oscillation in a local minimum state.

$$(P_i - P_{i-1}) \cdot (P_{i-1} - P_{i-2}) \le 0, \tag{30}$$

$$\|P_i - P_{i-1}\| \ll \varepsilon. \tag{31}$$

If it is judged to be a local minimum, then a virtual force is generated to escape from it; this is defined as the virtual source [32]. The robot must always move through the space separating the rigid body and the obstacle because a rigid body cannot pass through an obstacle. Moreover, because a rigid body cannot be separated, a stagnation point only occurs if a force is applied in the direction in which the rigid body is separated. To avoid the local minimum, a virtual force is therefore generated at the end-effector point of the body where the stagnation point occurs. By applying a small offset in a direction away from the origin, the robot moves through the space separating it and the obstacle. This is because of the virtual repulsive force defined as

$$P_{V-S} = P_{Ave,E} + \Delta P, \tag{32}$$

where P_{V-S} is the virtual source's position, $P_{Ave,E}$ is the end-effector's average position.

$$F_{\text{vir att}} = -\begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix}, \tag{33}$$

$$\overrightarrow{F_{\text{vir rep}}} = \begin{cases} C_{Vir,rep}\dfrac{P - O_{Vir}}{D^3}, & D \le 1, \\ 0, & D > 1. \end{cases} \tag{34}$$

The virtual force is released when the discriminant matrix (λ) becomes zero. Fig. 15 shows the simulation of a 3DOF robot with an obstacle.

5. TRAJECTORY PLANNING

5.1. Average angle

If there are obstacles in the path, the robot will vibrate at the obstacle boundary because of the attractive and repulsive forces. In order to prevent such a vibration in real time, a path using the average value of motion of each robot is obtained. By taking an average value, a time delay of (i−1) occurs, as shown in Fig. 16.

$$\theta_{i,Ave} = \frac{\theta_i + \theta_{i+1}}{2}. \tag{35}$$

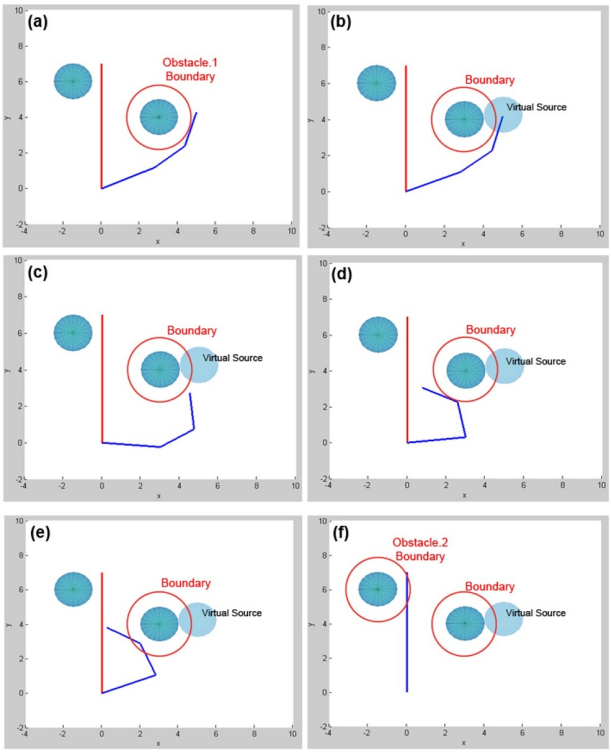

Fig. 15. Virtual source for local minimum avoidance(a) local minimum state; (b) generated virtual source to escape local minimum; (c) virtual mode in progress; (d) point at which discriminant matrix becomes zero; (e) boundary is ignored and moves toward the target point because discriminant matrix is zero; (f) arrival at target position.

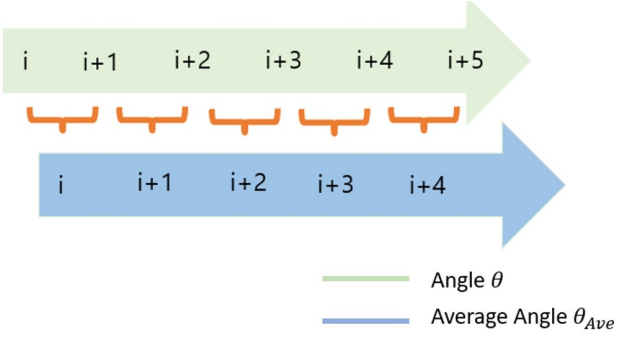

Fig. 16. Average θ timeline.

5.2. Cubic trajectory planning

When a trajectory is created using a cubic trajectory, it can be defined as follows:

$$\theta(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3, \tag{36}$$

$$\dot{\theta}(t) = a_1 + a_2 t + a_3 t^2. \tag{37}$$

The boundary conditions are as follows:

$$\theta(t_i) = \theta_i, \quad \theta(t_e) = \theta_e, \tag{38}$$

$$\dot{\theta}(t_i) = \dot{\theta}_i, \quad \dot{\theta}(t_e) = \dot{\theta}_e. \tag{39}$$

This is a boundary value problem hence, it can be solved using a matrix.

$$\begin{bmatrix} 1 & t_i & t_i^2 & t_i^3 \\ 1 & t_e & t_e^2 & t_e^3 \\ 0 & 1 & 2t_i & t_i^2 \\ 0 & 1 & 2t_e & t_e^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} \theta_i \\ \theta_e \\ \dot{\theta}_i \\ \dot{\theta}_e \end{bmatrix}. \tag{40}$$

Cubic trajectory planning must be designed while maintaining the obstacle avoidance path conceived through path planning; hence, setting the boundary value is of utmost importance. Accordingly, it is necessary to set all boundaries, non-obstacle mode, potential field, and virtual source mode according to the boundary values above.

6. SIMULATION AND EXPERIMENT

The simulations and experiments are separated into two parts: the avoidance of a sphere-shaped obstacle by passing through the z-axis, and the avoidance of a cylinder-shaped obstacle wherein passage through the z-axis is not possible. The initial and desired positions are calculated by Jacobian-based inverse kinematics, and the path is determined by the modified potential field method and non-obstacle mode. After path planning, trajectory planning is run in a simulation, and the result is thereafter evaluated through a real robot arm motion experiment.

6.1. Sphere obstacle

Fig. 17 shows the simulation result of the robot arm's avoidance of a spherical obstacle. The blue and red lines denote the robot's current position and target object position, respectively. This robot calculates the starting and target angles by Jacobian-based numerical inverse kinematics. The path is thereafter generated by the modified potential field or non-obstacle mode method. According to the obstacle judgment condition, the modified potential field method is used from 0 to approximately 5.6 s, and the non-obstacle mode method is used from 5.6 s until the desired position is reached. As the potential field path is passed through the z axis of the obstacle, the obstacle no longer affects the robot path and is therefore inconsequential.

Fig. 18 shows that the angle and angular velocity graph are divided into three sections because of the non-obstacle mode and potential field. The first section of the angle and angular velocity graph from 0 to approximately 2.4 s is the modified potential mode in which only the attractive force is applied. From approximately 2.4 to 5.6 s, the robot and obstacle interact with the attractive and repulsive forces, and the robot generates a path to avoid the obstacle. Finally, the non-obstacle mode is used because the attractive force does not affect any obstacles from 5.6 s until the desired target is reached.

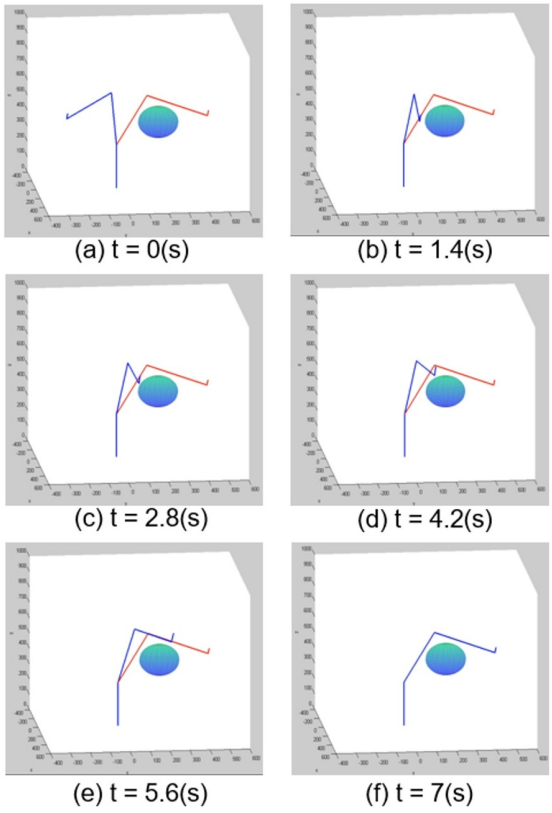

Fig. 17. Spherical obstacle simulation result.

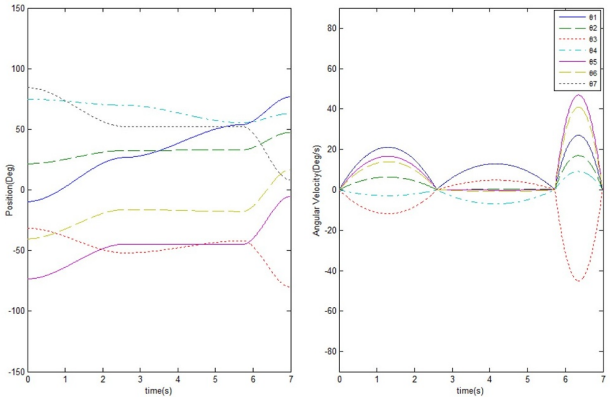

Fig. 18. Angle and angular velocity graph.

Fig. 19 shows the experimental result of the real robot arm. In the simulation, the robot arm avoids the obstacle and reaches the target position.

Table 2 summarizes the details of the spherical obstacle experiment. The simulation, which is repeated 698 times from the initial position to the target position, took 1.3438 s.

6.2. Cylindrical obstacle

Fig. 20 shows the simulation result of the robot arm with the cylindrical obstacle. This obstacle does not allow passage through the z-axis; accordingly, a detour is generated for the robot. Fig. 21 shows that the angle and angular velocity graph are divided into five sections: non-obstacle mode, local minimum, virtual source, potential field, and a gained non-obstacle mode. The second sec-

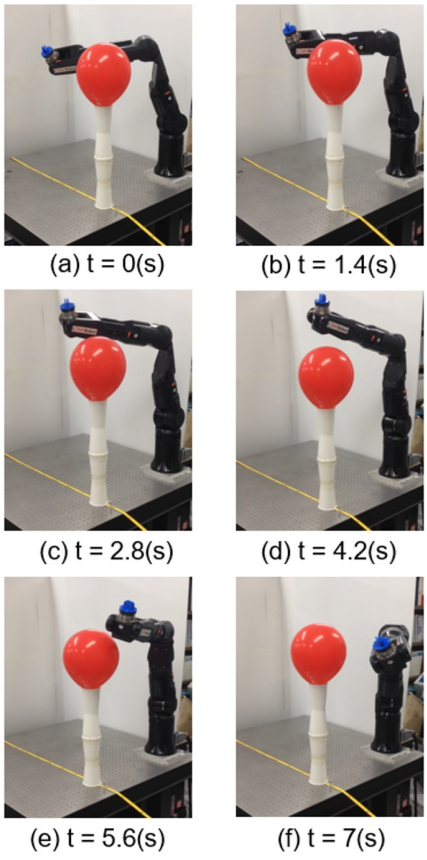

Fig. 19. Spherical obstacle experiment detail.

Table 2. Spherical obstacle experiment details.

Coordinates of the obstacle	(500, 70, 600)
radius of obstacle	(r=100)
Shape of obstacle	sphere
Starting position of end-effector	(550, -200, 650)
Starting orientation of end-effector	(0°, 45°, −45°)
Starting angles	-9.7174, 21.5375, -31.6158, 74.8054, -73.4074, -40.3821, 84.2993
Desired position of end-effector	(550, 300, 650)
Desired orientation of end-effector	(0°, 45°, −45°)
Desired each joint angles	76.6331, 47.4008, -80.0537, 62.8878, -5.4603, 16.6616, 8.2391
Calculation time	1.925 (ms/cycle) (cpu:i7-7700)

unit : Length (mm), angle (°)

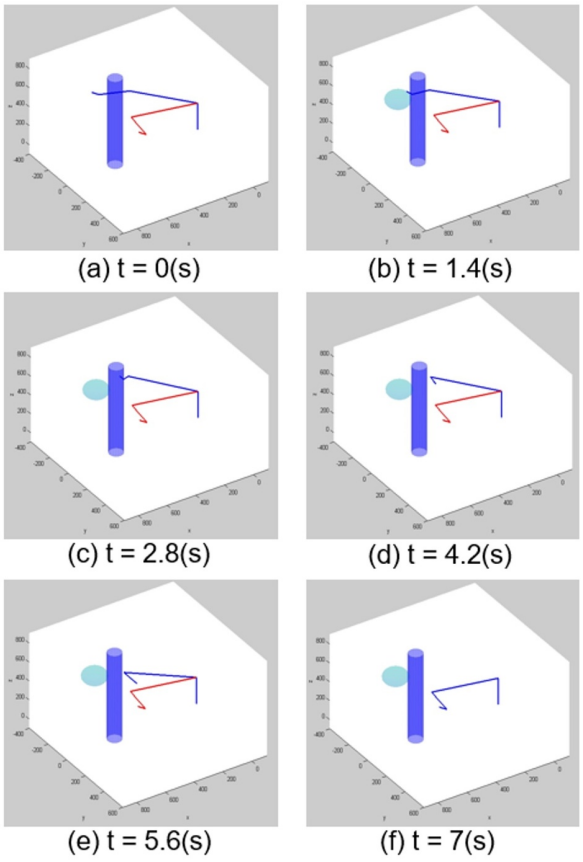

Fig. 20. Cylindrical obstacle simulation result.

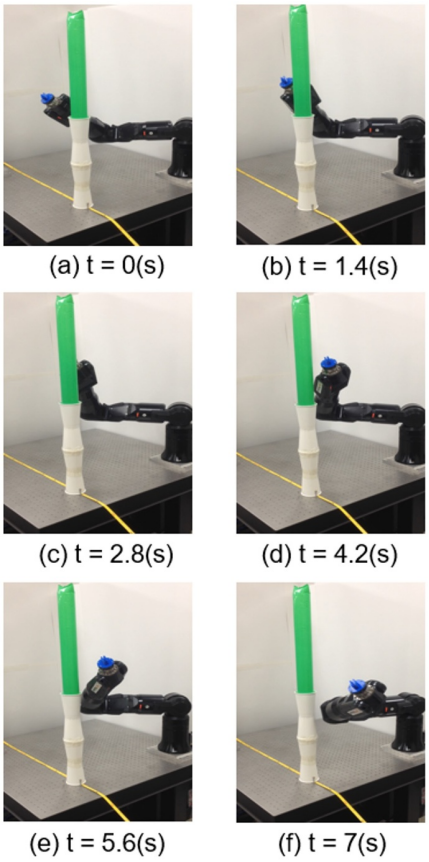

Fig. 22. Cylindrical obstacle experimental result.

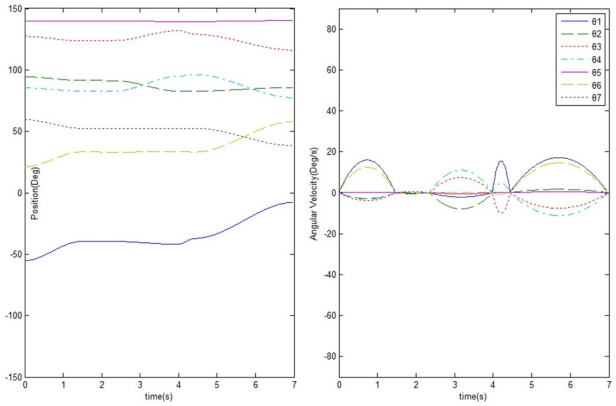

Fig. 21. Angle and angular velocity graph for solving cylindrical obstacle problem.

tion, from approximately 1.2 to 2 s, indicates the local minimum judgment. In view of this, the angle and velocity are terminated. After judging the local minimum, the virtual source is generated at $t = 1.4$ s to escape the local minimum and move to the target position after avoiding the obstacle. Fig. 22 shows the experimental results of the real robot arm in the cylindrical obstacle experiment. Table 3 summarizes the details of this experiment.

Table 3. Cylindrical obstacle experiment details.

Coordinates of the obstacle	(600, -10, ∞)
radius of obstacle	($r = 32$)
Shape of obstacle	cylinder
Starting position of end-effector	(600, -200, 550)
Starting orientation of end-effector	(0°, 45°, −45°)
Starting angles	-55.2217, 94.3661, 127.3170, 85.3764, 139.6585, 21.0791, 59.4845
Desired position of end-effector	(600, 300, 550)
Desired orientation of end-effector	(0°, 45°, −45°)
Desired angles	-7.8782, 85.6572, 116.0870, 77.3493 140.2028, 58.0349, 38.3366
Calculation time	2.5 (ms/cycle) (cpu:i7-7700)

unit : Length (mm), angle (°)

7. CONCLUSION

To solve the inverse kinematics of high-DOF robots and various shapes, the Jacobian-based numerical method is employed. As the DOF increases, however, the numerical

method requires more computational time, making it impossible to guarantee the robot's real-time performance. The Jacobian-based numerical inverse kinematics method is, thus, first used only to calculate the desired position. Other route positions are designed by the modified potential field method to ensure performance in real time. Using the potential field method, the real-time path analysis of the robot becomes possible.

If the obstacle is recognized, the avoidance path of the robot can be designed by the original potential field method. With the original method, however, depending on the positions of the robot and obstacle, it is possible that an ineffective path will be computed, vibration will occur, speed and acceleration will suddenly change, and a local minimum problem will be encountered. To resolve these potential field problems, this study proposed the use of the modified potential field method.

In the numerical method, with the rotation order concept of roll, pitch, and yaw, a pseudoinverse Jacobian concept for a non-square matrix, and a resolution concept when obstacles exist in the calculated robot shape are considered. The modified potential field method is formulated through the inclusion of concepts, such as midpoint definition, obstacle identification condition, judgment matrix, boundary definitions (boundaries 0, 1, and 2), and detection of and escape from the local minimum.

Based on these concepts, inverse kinematics using the Jacobian-based numerical method is calculated. The 7-DOF robot could autonomously move towards the target object position when its desired position and orientation are given. To follow the path of the robot based on the angle calculated by inverse kinematics and the proposed potential field method when obstacles are present, the following are suggested: identify the relationship between force and torque; consider the rigid body and material points; determine the local minimum and the way to escape it. Trajectory planning is performed while the path for avoiding an obstacle is maintained. Simulations and experiments are conducted to verify the validity of the path for avoiding obstacles, and appropriate trajectory planning is achieved considering the robot position and the use of velocity graphs.

Based on the simulations, the numerical method results indicate that the maximum difference between the real robot arm's desired position and the desired position obtained from the simulation is 0.006 mm. The maximum difference between the desired and simulation orientations is 0.5° when dP is divided by 1000. The calculation time using numerical inverse kinematics is approximately 0.8 s; hence, the robot arm moves with real-time performance after this time. The simulation and experimental results indicate that the robot arm position well matches the target position, thereby verifying the robot's real-time performance and ability to avoid obstacles using the proposed method.

REFERENCES

[1] World Robotics Research, 2015, https://ifr.org/worldrobotics/.

[2] SDA20D, Dual-Arm industrial Robot, https://www.motoman.com/industrial-robots/.

[3] S. Nicosia and P. Tomei, "Robot control by using only joint position measurements," *IEEE Trans. Automat Cont.*, vol. 35, no. 9, pp. 1058-1061. 1990.

[4] G. S. Choi and C. S. Kim, "Linear stable systems," *IEEE Trans. on Automatic Control*, vol. 33, no. 3, pp. 1234-1245, Dec. 1993.

[5] K. S. Fu, R. C Gonzalez, and C. S. G. Lee, *ROBOTICS : Control, Sensing, Vision, and Intelligence*, McGraw-Hill, 1987.

[6] S. B Niku, *Introduction to Robotics Analysis, Systems, Application*, PrenticeHall, 2001.

[7] S. Nicosia and P. Tomei, "Robot control by using only joint position measurements," *IEEE Trans. Automat Cont.*, vol. 35, no. 9, pp. 1058-1061. 1990.

[8] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Dynamics and Control*, pp. 33-130, 2004.

[9] S. Jeong, *Robotics Application of Matlab & Simulink*, pp. 130-150, 2015.

[10] P. Corke, *Robotics, Vision and Control*, Springer Publishing, pp. 137-160, 2011.

[11] T. Ho, C. G. Kang, and S. Lee, "Efficient closed-form solution of inverse kinematics for a specific six-dof arm," *International Journal of Control, Automation, and Systems*, vol. 10, no. 3, pp.567-573, 2012.

[12] W. A. Wolvich, *ROBOTICS: Basic Analysis and Design*, CBS College Publishing, 1987.

[13] R. V. Patel and F. Shadpey, *Control of Redundant Robot Manipulators*, Springer, pp.34-78, 2005.

[14] Y. Zhang and L. Jin, *Robot Manipulator Redundancy Resolution*, Wiley, pp. 49-66, 2017.

[15] O. Kanoun, F. Lamiraux, and P. B. Wieber, "Kinematic control of redundant manipulators: Generalizing the task priority framework to inequality task," *IEEE Trans. on Robotics*, vol. 27, no. 4, pp. 785-792, 2011.

[16] M. V. Kirdanski, "Symbolical singular value decomposition for a 7-DOF manipulator and its application to Robot Control," *Proc. of IEEE International Conference on Robotics and Automation*, vol. 3, pp. 895-900, 1993.

[17] L. Sciavicco and B. Siciliano, "A solution algorithm to the inverse kinematic problem for redundant manipulators," *IEEE Journal of Robotics and Automation*, vol. 4, no. 4, pp. 403-410, 1988.

[18] B. R. Munson, A. P. Rothmayer, T. H. Okilshi, and W. W. Huebsch, *Fundamentals of Fluid Mechanics*, Wiley, 2013.

[19] J. Angeles, *Fundamentals of Robotic Mechanical Systems: Theory, Methods, and Algorithms*, Springer-Verlag, 1997.

[20] K. Nakazawa, "Unified environment-adaptive control of accompanying robots using artificial potential field," *Proc. of 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 199-200, 2013.

[21] Y. Kitazawa and J. Mukai, "Robot behaviors connected with utterances and environments via potential fields," *Proc. of SICE Annual Conference*, pp. 614-617, 2007.

[22] F. Fahimi, *Autonomous Robots Modeling, Path Planning and Control*, Springer, pp. 44-60, 2008.

[23] H. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun, *Principles of Robot Motion*, MIT Press, 2007.

[24] C. W. Warren, "Global path planning using artificial potential fields," *Proc. of International Conference on Robotics and Automation*, pp. 316-321, 1989.

[25] N. Zhang, Y. Zhang, C. Ma, and B. Wang, "Path planning of six-DOF serial robots based on improved artificial potential field method," *Proc. of IEEE International Conference on Robotics and Biomimetics*, pp. 617-621, 2017.

[26] Z. Elm and M. O. Efe, "Path planning using model predictive controller based on potential field for autonomous vehicles," *Proc. of 44th Annual Conference of the IEEE industrial Electronics*, pp.2613-2618, 2018.

[27] Y. Peng, Z. Yan, H. Zheng, and J. Guo, "Real-time robot path planning method based on improved artificial potential field method," *Proc. of 37th Chinese Control Conference*, pp. 4814-4820, 2018.

[28] N. Zhang, "Path planning of six-DOF serial robots based on improved artificial potential field method," *Proc. of International Conference on Robotics and Biomimetics*, pp. 617-621, 2017.

[29] Y. Liu, C. Yu, J. Sheng, and T. Zhang, "Self-collision avoidance trajectory planning and robust control of a dual-arm space robot," *International Journal of Control, Automation, and Systems*, vol. 16, no. 6, pp.2896-2905, 2018.

[30] J. M. Gere and B. J. Goodno, *Mechanics of Materials*, Cengage Learning, 2011.

[31] M. G. Park and M. C. Lee, "Real-time path planning in unknown environments using a new potential field approach with a virtual hill," *Proc. of 30th Annual Conference of IEEE Industrial Electronics Society*, 2004.

[32] M. G. Park and M. C. Lee, "A new technique to escape local minimum in artificial potential field based path planning," *Int. Journal of KSME*, vol. 17 no. 12, pp. 1876-1885, 2003.

[33] Q. Xue, A. A. Maciejewski, and P. C. Y. Sheu, "Determining the collision-free joint space graph for two cooperating robot manipulators," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 23, no. 1, pp. 285-294, 1993.

Sun-Oh Park is working towards the completion of his M.S. degree in Mechanical Engineering from Pusan National University, Busan, Korea since 2017. He received his B.S. degree in Mechanical Engineering from Pusan National University, Busan, Korea in 2015. His research interests include intelligent robot control, dynamics, machine learning, machine vision, development path planning. autonomous robots and simulation.

Min Cheol Lee received his Ph.D. degree in Applied Physics from the University of Tsukuba, Tsukuba, Japan in 1991, an M.S. degree in Engineering Science from the University of Tsukuba, Tsukuba, Japan in 1988, and a B.S. degree in Mechanical Engineering from Pusan National University, Busan, Korea in 1983. He was a visiting professor at North Carolina State University from Aug. 2000 to Aug. 2001 and at Purdue University from Aug. 2009 to Aug. 2010. From 1991 to present, he is a professor in the School of Mechanical Engineering from Pusan National University, Korea. His research interests include intelligent robot control, autonomous mobile robot, medical robotics, system identification, sliding mode control, and navigation/localization of mobile robots.

Jaehyung Kim is working towards the completion of his M.S. degree in Mechanical Engineering from Pusan National University, Busan, Korea since 2019. He received his B.S. degree in Mechanical Engineering from Pusan National University, Busan, Korea in 2019. His research interests include intelligent robot control, kinematics, machine learning, machine vision, path planning and autonomous robots and simulation.