

DESCRIPTION OF THE INVENTION

Title of Invention: CONTROL METHOD FOR SELF-COLLISION

**AVOIDANCE OF ROBOT HAVING MULTIPLE JOINTS AND ROBOT FOR
PERFORMING SAME**

TECHNICAL FIELD

[01] The present invention relates to a control method for avoiding self-collision of a robot configured of a plurality of joints.

BACKGROUND ART

[02] A robot means a machine that automatically processes or operates a given task using its own ability, and application fields of the robot may be classified into industrial, service, medical, space, and underwater purposes.

[03] Among them, industrial robots are applied to industrial automation, and mean multi-purpose manipulators capable of automatic control and reprogram, which mean fixed or mobile robots capable of programming in three axes or more, and may include hand-guided robots, manipulator parts of mobile robots, collaborating robots, and the like.

[04] Meanwhile, an industrial robot system may be configured by including terminal devices, and all machines, equipment, devices,

auxiliary axes, or sensors needed for the robot to perform tasks, together with the industrial robots described above.

[05] Industrial robot systems are already widely used in machining industries, such as automobile production, to repeat operations corresponding to the work of a human arm, and their utilization is increasing as the labor cost increases.

[06] Meanwhile, since multi-joint robots such as industrial robots or walking robots use a plurality of motors, self-collision may occur according to the driving angle.

[07] Accordingly, in order to control the multi-joint robots, various variables, such as mechanical limit angles of hardware, self-collision avoidance, and the like, should be considered, and to this end, techniques for the robots to find an optimal trajectory for moving while avoiding paths where self-collision may occur are developed.

[08] However, since the workspace of the multi-joint robots may be narrowed when movement of the robots is limited to avoid self-collision, it needs to develop techniques that allow the multi-joint robots to have a workspace as large as possible.

SUMMARY OF THE INVENTION

TECHNICAL PROBLEM

[09] Therefore, the present invention has been made in view of the above problems, and it is an object of the present invention

to provide a control method that allows a multi-joint robot to efficiently avoid self-collision, and a robot that performs the method.

[10] The technical problems of the present invention are not limited to those mentioned above, and unmentioned other technical problems will be clearly understood by those skilled in the art from the following description.

TECHNICAL SOLUTION

[11] To accomplish the above object, according to one aspect of the present invention, there is provided a control method for self-collision avoidance of a robot having multiple joints, the method comprising the steps of: modeling each of collision-prone parts of the robot as a three-dimensional shape having a line segment and a circular cross-section; calculating a shortest distance between a first line segment of a first collision-prone part and a second line segment of a second collision-prone part among the modeled collision-prone parts; calculating a relative Jacobian between a first point on the first line segment and a second point on the second line segment connecting the shortest distance, by applying a Jacobian matrix to the first and second points; and calculating a speed for avoiding collision between the first and second collision-prone parts using a size of the three-dimensional shape, the shortest distance between the first and

second line segments, and the relative Jacobian between the first and second points.

[12] The collision-prone part may include a link portion that moves by a joint provided in the robot, and the three-dimensional shape may be formed using a set of points at a predetermined distance from the line segment.

[13] At least part of the control method for self-collision avoidance of a multi-joint robot may be implemented as a computer-readable recording medium that records a program to be executed on a computer, and may be provided as the program itself.

[14] Meanwhile, the control method for self-collision avoidance of a multi-joint robot may be performed by a robot according to an embodiment of the present invention.

[15] A robot according to an embodiment of the present invention has a collision avoidance function, and to this end, the robot comprises: a processor; a memory for loading a computer program executed by the processor; and a storage for storing the computer program, wherein the computer program includes instructions for executing an operation of modeling each of collision-prone parts of the robot as a three-dimensional shape having a line segment and a circular cross-section, an operation of calculating a shortest distance between a first line segment of a first collision-prone part and a second line segment of a second collision-prone part among the modeled collision-prone parts, an

operation of calculating a relative Jacobian between a first point on the first line segment and a second point on the second line segment connecting a shortest distance, an operation of calculating a speed for avoiding collision between the first and second collision-prone parts using the size of the three-dimensional shape, the shortest distance between the first and second line segments, and the relative Jacobian between the first and second points.

ADVANTAGEOUS EFFECTS

[16] According to an embodiment of the present invention, as a self-collision avoidance function of a multi-joint robot may be applied in real time through relatively simple operations by modeling the collision-prone parts of the robot as line segments and three-dimensional shapes and calculating a speed for avoiding collision between the collision-prone parts using the size of the three-dimensional shape, the shortest distance between the line segments, and a relative Jacobian, self-collision can be prevented while expanding the workspace of the robot.

[17] The technical effects of the present invention are not limited to those mentioned above, and unmentioned other technical effects will be clearly understood by those skilled in the art from the following description.

BRIEF DESCRIPTION OF THE DRAWINGS

[18] FIG. 1 is a perspective view showing an example of the configuration of a multi-joint robot.

[19] FIG. 2 is a flowchart illustrating a control method for self-collision avoidance of a multi-joint robot according to an embodiment of the present invention.

[20] FIG. 3 is a view conceptually showing the configuration of a multi-joint robot according to an embodiment of the present invention.

[21] FIGS. 4 and 5 are views for explaining embodiments of a method of modeling a collision-prone part of a robot.

[22] FIGS. 6 through 11 are views for explaining embodiments of a method of calculating a speed for avoiding collision.

[23] FIG. 12 is a block diagram showing the configuration of a computing device in which a method according to an embodiment of the present invention is implemented.

DETAILED DESCRIPTION OF THE INVENTION

[24] Hereinafter, only the principles of the present invention will be exemplified. Therefore, although not clearly described or shown in this specification, those skilled in the art will be able to implement the principles of the present invention and invent various devices included in the spirit and scope of the present invention. In addition, it should be understood that all

conditional terms and embodiments listed in this specification are, in principle, clearly intended only for the purpose of understanding the concept of present invention, and not limited to the embodiments and states specially listed as such.

[25] In addition, it should be understood that all detailed descriptions listing specific embodiments, as well as the principles, aspects, and embodiments of the present invention, are intended to include structural and functional equivalents of such matters. In addition, it should be understood that such equivalents include equivalents that will be developed in the future, as well as currently known equivalents, i.e., all devices invented to perform the same function regardless of the structure.

[26] Accordingly, for example, the block diagrams in the specification should be understood as expressing the conceptual viewpoints of exemplar circuits that embody the principles of the present invention. Similarly, all flowcharts, state transition diagrams, pseudo code, and the like may be practically embodied on computer-readable media, and it should be understood that regardless of whether or not a computer or a processor is explicitly shown, they show various processes performed by the computer or processor.

[27] The functions of various elements shown in the drawings, including functional blocks indicated as a processor or a similar concept, may be provided by the use of hardware capable of executing

software in connection with appropriate software, as well as dedicated hardware. When provided by the processor, the functions may be provided by a single dedicated processor, a single shared processor, or a plurality of individual processors, and some of them may be shared.

[28] In addition, explicit use of the terms presented as a processor, control, or concepts similar thereto should not be interpreted by exclusively quoting hardware having an ability of executing software, and should be understood to implicitly include, without limitation, digital signal processor (DSP) hardware, and ROM, RAM and non-volatile memory for storing software. Other known common hardware may also be included.

[29] In the claims of this specification, a component expressed as a means for performing a function described in the detailed description is intended to include all methods that perform functions including, for example, a combination of circuit elements performing the functions or all types of software including firmware/microcode, and is combined with appropriate circuits for executing the software to perform the functions. Since the present invention defined by these claims is combined with the functions provided by the means arranged in various ways, and combined with the methods required by the claims, it should be understood that any means capable of providing the functions is equivalent to those identified from this specification.

[30] The above objects, features, and advantages will become more apparent through the following detailed description related to the accompanying drawings, and accordingly, those skilled in the art may easily implement the technical spirit of the present invention. In addition, when it is determined in describing the present invention that the detailed description of a known technique related to the present invention may unnecessarily obscure the gist of the present invention, the detailed description will be omitted.

[31] Hereinafter, although a collaborative robot is described as an example in the embodiments of the present invention, the present invention is not limited thereto and may be applicable to industrial robots and various other robots.

[32] A collaborative robot is a robot designed to directly interact with humans within a defined collaborative workspace, and a collaborative operation may mean the working state of a robot system and an operator intentionally designed within the collaborative workspace.

[33] In addition, the collaborative workspace is a workspace within a safeguarded space where robots and humans perform tasks together during production operations, and it may mean an area inside a workspace where a robot system (including workpieces) and a person may perform tasks together during a production task.

[34] FIG. 1 is a view showing the configuration of a robot according to an embodiment of the present invention, and the robot 100 may be configured to include a base 110, a plurality of joints 120 to 128, and a tool flange 150.

[35] Referring to FIG. 1, the base 110 is a part for fixing the robot 100, and a cable connection connector between the robot 100 and the control box may be formed on the base 110.

[36] The tool flange 150 is a part for mounting a gripper or a tool on the robot 100, and input/output ports for controlling the gripper or tool and buttons for direct teaching may be arranged at the positions adjacent to the tool flange 150.

[37] Meanwhile, as a base joint 120, a shoulder joint 122, an elbow joint 124, a first wrist joint 126, a second wrist joint 127, and a third wrist joint 128 are provided between the base 110 and the tool flange 150, the robot 100 may perform 6-axis joint operations.

[38] As described with reference to FIG. 1, as the multi-joint robot operates according to rotation of a motor provided in each joint, self-collision may occur according to driving angles. Therefore, the workspace of the robot that should be restricted to prevent self-collision needs to be changed in real time.

[39] According to an embodiment of the present invention, as a self-collision avoidance function of a multi-joint robot may be applied in real time through relatively simple operations by

modeling the collision-prone parts of the robot as line segments and three-dimensional shapes and calculating a speed for avoiding collision between the collision-prone parts using the size of the three-dimensional shape, the shortest distance between the line segments, and a relative Jacobian, self-collision can be prevented while expanding the workspace of the robot.

[40] Hereinafter, the embodiments of a control method for self-collision avoidance of a multi-joint robot according to the present invention will be described in more detail with reference to FIGS. 2 to 11.

[41] Referring to FIG. 2, each of the collision-prone parts of a robot is modeled as a three-dimensional shape having a line segment and a circular cross-section (step S200).

[42] For example, the collision-prone part may be set to include a link portion that moves by a joint provided in the robot.

[43] As shown in FIG. 3, the robot may be configured to include a plurality of joints 301, 302, 303, 304, 305, and 306 and a plurality of links 311, 312, 313, 314, 315, and 316 connecting the joints, and each link may move by the rotation of a motor of a joint to which one end of the link is connected.

[44] In this case, as shown in FIG. 4, the collision-prone parts 410, 420, 430, 440, and 450 of the robot may be set to include joints and links.

[45] Here, each collision-prone part is modeled as a line segment, and a three-dimensional shape may be formed using a set of points at the same predetermined distance from the line segment.

[46] Accordingly, the three-dimensional shape may be configured to include a cylindrical part having a radius corresponding to the predetermined distance, and two hemispherical parts located above and below the cylindrical part.

[47] Referring to FIG. 5, the collision-prone part 410 may be modeled as a three-dimensional shape configured of a line segment 413, a cylindrical part 411 having a predetermined radius (r) around the line segment 413, an upper hemispherical part 412 which is a set of points at a predetermined distance (r) from the upper end of the line segment 413, and a lower hemispherical part 412' which is a set of points at a predetermined distance (r) from the lower end of the line segment 413.

[48] Here, the three-dimensional shape of the collision-prone part described above may represent an inaccessible space in the part, and as the radius (r) increases, the inaccessible space may be expanded, and as the radius (r) decreases, the inaccessible space may be narrowed.

[49] Next, the shortest distance between a first line segment of a first collision-prone part and a second line segment of a second collision-prone part among the collision-prone parts modeled at step S200 is calculated (step S210).

[50] For example, as shown in FIG. 6(a), when the collision-prone parts 410, 420, 430, and 440 of the robot are modeled as line segments and three-dimensional shapes, the shortest distance between the line segment 443 of the first collision-prone part 440 and the line segment 413 of the second collision-prone part 410 may be calculated as shown in FIG. 6(b).

[51] Meanwhile, as shown in FIG. 6(b), when the shortest distance between the line segments 443 and 413 of the two collision-prone parts 440 and 410 is smaller than the sum (r_0+r_1) of the radii (r_0 , r_1) of the two collision-prone parts 440 and 410, it may be determined that the two collision-prone parts 440 and 410 have invaded each other's inaccessible spaces.

[52] Here, the radii (r_0 , r_1) of the two collision-prone parts 440 and 410 may be set to be the same or different, and the larger the radius of the three-dimensional shape, the wider the inaccessible space of the collision-prone part.

[53] Thereafter, a relative Jacobian between the first and second points is calculated by applying a Jacobian matrix to a first point of the first line segment and a second point of the second line segment connecting the shortest distance calculated at step S210 (step S220).

[54] Next, a speed for avoiding collision between the first and second collision-prone parts is calculated using the size of the three-dimensional shape, the shortest distance between the

first and second line segments, and the relative Jacobian between the first and second points (step S230).

[55] The relative Jacobian calculated at step S220 may be a difference between a first Jacobian value, to which a Jacobian matrix is applied at the first point, and a second Jacobian value, to which a Jacobian matrix is applied at the second point, and accordingly, may represent the relative velocity between the first and second collision-prone parts with respect to the first and second points.

[56] For example, as shown in FIG. 7, the locations of a first point (S_0) and a second point (S_1) connecting the shortest distance (d_{shortest}) between a first line segment 443 of the first collision-prone part 440 and a second line segment 413 of the second collision-prone part 410 may be obtained.

[57] The Jacobian matrix is used to convert the motor angular velocity of each joint provided in the robot into linear velocities and angular velocities at a specific point, and as shown below in Equation 1, the linear velocities ($v_{m,x}$, $v_{m,y}$, $v_{m,z}$) in the x, y, and z axis directions at a corresponding point may be calculated by multiplying a (6×1) matrix, configured of angular velocities ($q_{m,0}$, $q_{m,1}$, $q_{m,2}$, $q_{m,3}$, $q_{m,4}$, $q_{m,5}$) of six motors, by a (3×6) Jacobian matrix.

[58] **[Equation 1]**

$$\begin{bmatrix} J_{00} & J_{01} & J_{02} & J_{03} & J_{04} & J_{05} \\ J_{10} & J_{11} & J_{12} & J_{13} & J_{14} & J_{15} \\ J_{20} & J_{21} & J_{22} & J_{23} & J_{24} & J_{25} \end{bmatrix} \begin{bmatrix} \dot{q}_{m,0} \\ \dot{q}_{m,1} \\ \dot{q}_{m,2} \\ \dot{q}_{m,3} \\ \dot{q}_{m,4} \\ \dot{q}_{m,5} \end{bmatrix} = \begin{bmatrix} v_{m,x} \\ v_{m,y} \\ v_{m,z} \end{bmatrix}$$

[59] When a Jacobian matrix is applied to each of the first point (S_0) and the second point (S_1), the velocity (v^{s_0}) at the first point (S_0) and the velocity (v^{s_1}) at the second point (S_1) can be obtained as shown in FIG. 8.

[60] In addition, as shown in FIG. 8, a normal vector ($n^{s_0s_1}$) of magnitude 1 in the direction from the first point (S_0) to the second point (S_1) can be obtained.

[61] In addition, the relative Jacobian between the first and second points (S_0, S_1) is calculated as shown below in Equation 2, and a relative velocity between the first and second collision-prone parts 440 and 410 with respect to the first and second points (S_0, S_1) can be obtained.

[62] 【Equation 2】

$$(J_{S_0} - J_{S_1})\dot{q}$$

[63] As shown in Equation 2, the relative Jacobian between the first and second points (S_0, S_1) may be obtained by subtracting a second Jacobian value obtained by applying a Jacobian matrix (J_{S_1}) to the second point (S_1) from a first Jacobian value obtained by applying a Jacobian matrix (J_{S_0}) to the first point (S_0).

[64] Here, as the Jacobian matrix (J_{S_0}) at the first point (S_0) and the Jacobian matrix (J_{S_1}) at the second point (S_1) are different from each other and calculated as shown below in Equation 3, the relative velocities ($v'_{m,x}$, $v'_{m,y}$, $v'_{m,z}$) in the x, y, and z axis directions may be calculated between the first and second points (S_0 , S_1).

[65] [Equation 3]

$$\begin{bmatrix} J_{00}^{S_0} - J_{00}^{S_1} & J_{01}^{S_0} - J_{01}^{S_1} & J_{02}^{S_0} - J_{02}^{S_1} & J_{03}^{S_0} - J_{03}^{S_1} & J_{04}^{S_0} - J_{04}^{S_1} & J_{05}^{S_0} - J_{05}^{S_1} \\ J_{10}^{S_0} - J_{10}^{S_1} & J_{11}^{S_0} - J_{11}^{S_1} & J_{12}^{S_0} - J_{12}^{S_1} & J_{13}^{S_0} - J_{13}^{S_1} & J_{14}^{S_0} - J_{14}^{S_1} & J_{15}^{S_0} - J_{15}^{S_1} \\ J_{20}^{S_0} - J_{20}^{S_1} & J_{21}^{S_0} - J_{21}^{S_1} & J_{22}^{S_0} - J_{22}^{S_1} & J_{23}^{S_0} - J_{23}^{S_1} & J_{24}^{S_0} - J_{24}^{S_1} & J_{25}^{S_0} - J_{25}^{S_1} \end{bmatrix} \begin{bmatrix} \dot{q}_{m,0} \\ \dot{q}_{m,1} \\ \dot{q}_{m,2} \\ \dot{q}_{m,3} \\ \dot{q}_{m,4} \\ \dot{q}_{m,5} \end{bmatrix} = \begin{bmatrix} v'_{m,x} \\ v'_{m,y} \\ v'_{m,z} \end{bmatrix}$$

[66] Referring to FIG. 9 (a), the relative velocity (v') between the first and second points (S_0 , S_1) can be obtained by subtracting the velocity (v^{s_1}) at the second point (S_1) from the velocity (v^{s_0}) at the first point (S_0).

[67] In addition, the velocity components between the first and second collision-prone parts in a direction from the first point to the second point can be calculated using the relative Jacobian between the first and second points calculated as described above.

[68] For example, as shown below in Equation 4, only the velocity component ($v'^{s_0s_1}$) in the direction from the first point (S_0) to the second point (S_1) can be extracted from the relative velocity between the first and second points (S_0 , S_1) by multiplying

the relative velocity between the first and second points (S_0, S_1) of the (3×1) matrix obtained in Equation 3 by the normal vector component represented as the (1×3) matrix in the direction from the first point (S_0) to the second point (S_1).

[69] 【Equation 4】

$$\begin{bmatrix} \hat{n}_x^{S_0S_1} & \hat{n}_y^{S_0S_1} & \hat{n}_z^{S_0S_1} \end{bmatrix} \begin{bmatrix} v'_{m,x} \\ v'_{m,y} \\ v'_{m,z} \end{bmatrix} = v'^{S_0S_1}$$

[70] Referring to FIG. 9(b), the relative velocity component ($v'^{s_0s_1}$) in the direction from the first point (S_0) to the second point (S_1) can be obtained by performing inner product of the relative velocity (V') between the first and second points (S_0, S_1) and the normal vector ($n^{s_0s_1}$) in the direction from the first point (S_0) to the second point (S_1).

[71] The relative velocity component ($v'^{s_0s_1}$) in the direction from the first point (S_0) to the second point (S_1) obtained as described above may represent the relative velocity magnitude ($v'^{s_0s_1}$) in the shortest distance direction from the first point (S_0) as shown in FIG. 10.

[72] Referring to FIG. 11 (a), when the relative velocity ($v'^{s_0s_1}$) in the shortest distance direction from the first point (S_0) as described above is greater than 0, the first and second collision-prone parts 440 and 410 move closer to each other in the collision direction with respect to the first and second points (S_0, S_1).

[73] Meanwhile, referring to FIG. 11 (a), when the relative velocity ($v'^{s_0s_1}$) in the shortest distance direction from the first point (S_0) as described above is smaller than 0, the first and second collision-prone parts 440 and 410 move in a direction away from each other with respect to the first and second points (S_0 , S_1).

[74] Therefore, when collision is likely to occur between the first and second collision-prone parts 440 and 410, as a speed for reducing the relative velocity ($v'^{s_0s_1}$) in the shortest distance direction from the first point (S_0) to the second point (S_1) to be smaller than 0 is calculated and applied in a direction of moving the first and second collision-prone parts 440 and 410 away from each other with respect to the first and second points (S_0 , S_1), collision between the first and second collision-prone parts 440 and 410 can be avoided.

[75] Here, the speed calculated to avoid collision between the first and second collision-prone parts 440 and 410 may be the motor angular velocity of at least one among the plurality of joints provided in the robot.

[76] For example, the relative velocity ($v'^{s_0s_1}$) in the shortest distance direction from the first point (S_0) to the second point (S_1) can be reduced to be smaller than 0 by changing at least one among the motor angular velocity of the first joint corresponding to the first collision-prone part 440 and the motor

angular velocity of the second joint corresponding to the second collision-prone part 410, and therefore, the first and second collision-prone parts 440 and 410 may move in a direction away from each other with respect to the first and second points (S_0 , S_1), so that collision can be avoided.

[77] According to an embodiment of the present invention, step S230 may be performed to subtract the radius (r_0) of the three-dimensional shape of the first collision-prone part 440 and the radius (r_1) of the three-dimensional shape of the second collision-prone part 410 from the shortest distance ($d_{shortest}$) between the first and second line segments 443 and 413, and calculate a speed for avoiding collision between the first and second collision-prone parts 440 and 410 according to the result of subtraction.

[78] For example, a speed for avoiding collision between the first and second collision-prone parts 440 and 410 can be calculated by performing step S230 only when the value obtained by subtracting the sum of radii (r_0 , r_1) from the shortest distance ($d_{shortest}$) is a negative value smaller than 0 as shown below in Equation 5.

[79] **【Equation 5】**

$$(d_{shortest} - (r_0 + r_1)) < 0$$

[80] Meanwhile, when the value obtained by subtracting the sum of radii (r_0 , r_1) from the shortest distance ($d_{shortest}$) is a positive value greater than 0, the speed for avoiding collision

between the first and second collision-prone parts 440 and 410 is not calculated, and the movement can be continued without changing the speed of the first and second collision-prone parts 440 and 410.

[81] In addition, the speed of at least one among the first and second collision-prone parts 440 and 410, for calculating the velocity component in a direction of moving the first point (S_0) and the second point (S_1) away from each other, may be obtained using the result of subtraction as described above.

[82] For example, a speed for avoiding collision between the first and second collision-prone parts 440 and 410 may be calculated using Quadratic Programming (QP), which is an optimization technique for obtaining an optimal solution satisfying the condition such as Equation 6 shown below.

[83] **[Equation 6]**

$$\vec{n}^{S_0S_1T}(J_{S_0} - J_{S_1})\dot{q} \approx k(d_{shortest} - (r_0 + r_1))$$

[84] Specifically, when the value obtained by subtracting the sum of radii (r_0 , r_1) from the shortest distance ($d_{shortest}$) is a negative value smaller than 0, the motor angular velocity (q) of the joint, which controls the relative velocity ($v'^{s_0s_1}$) in the shortest distance direction from the first point (S_0) to the second point (S_1) to have a negative value smaller than 0, may be calculated.

[85] In this case, when the motor angular velocity (q) calculated to control the relative velocity ($v'^{s_0s_1}$) in the shortest distance direction to have a negative value is applied to the joint, the first and second collision-prone parts 440 and 410 move in a direction away from each other with respect to the first and second points (S_0, S_1), so that collision can be avoided.

[86] Meanwhile, since the absolute value of the relative velocity ($v'^{s_0s_1}$) in the shortest distance direction from the first point (S_0) to the second point (S_1) increases as the absolute value of a value obtained by subtracting the sum of radii (r_0, r_1) from the shortest distance ($d_{shortest}$) increases, the first and second collision-prone parts 440 and 410 may move at a faster speed with respect to the first and second points (S_0, S_1), and collision can be avoided.

[87] In addition, k is a constant in Equation 6, and the larger the value of k , the higher the speed for avoiding collision between the first and second collision-prone parts 440 and 410, and the smaller the value of k , the lower the speed for avoiding collision between the first and second collision-prone parts 440 and 410.

[88] The control method for self-collision avoidance of a multi-joint robot as described with reference to FIGS. 2 to 11 may be performed by a robot according to an embodiment of the present invention.

[89] Meanwhile, although it has been described in the embodiments of the present invention as an example that the robot 300 is a multi-joint robot having 6 degrees of freedom, the present invention is not limited thereto, and may be applied to various robots, such as multi-joint robots having 6 degrees or more of freedom or multi-legged walking robots.

[90] Hereinafter, an exemplary computing device 500, in which the methods described in various embodiments of the present invention are implemented, will be described with reference to FIG. 12. For example, the computing device 500 of FIG. 12 may be implemented as at least some components of a robot according to an embodiment of the present invention.

[91] Referring to FIG. 12, the computing device 500 may include one or more processors 510, a bus 550, a communication interface 570, a memory 530 for loading computer programs 591 executed by the processor 510, and a storage 590 for storing the computer programs 591. However, FIG. 12 only shows components related to the embodiments of the present invention. Therefore, those skilled in the art will understand that other general-purpose components may be included, in addition to the components shown in FIG. 12.

[92] The processor 510 controls the general operation of each component of the computing device 500. The processor 510 may be configured to include at least one among a Central Processing Unit

(CPU), a Micro Processor Unit (MPU), a Micro Controller Unit (MCU), a Graphic Processing Unit (GPU), and a processor of any form well known in the art of the present invention. In addition, the processor 510 may perform operations for at least one application or program for executing the methods and/or operations according to various embodiments of the present invention. The computing device 500 may have one or more processors.

[93] The memory 530 stores various types of data, commands and/or information. The memory 530 may load one or more programs 591 from the storage 590 to execute the methods and/or operations according to various embodiments of the present invention. Examples of the memory 530 may be RAM, but are not limited thereto.

[94] The bus 550 provides communication functions between the components of the computing device 500. The bus 550 may be implemented as various forms of buses, such as an address bus, a data bus, and a control bus.

[95] The communication interface 570 supports wired and wireless Internet communication of the computing device 500. The communication interface 570 may also support various communication methods, in addition to the Internet communication. To this end, the communication interface 570 may be configured to include a communication module well known in the technical field of the present invention.

[96] The storage 590 may store one or more computer programs 591 non-temporarily. The storage 590 may be configured to include volatile memory such as Read Only Memory (ROM), Erasable Programmable ROM (EPROM), Electrically Erasable Programmable ROM (EEPROM), flash memory, a hard disk, a removable disk, or any form of computer-readable recording media well known in the art to which the present invention belongs.

[97] The computer program 591 may include one or more instructions, in which the methods and/or operations according to the embodiments of the present invention as described with reference to FIGS. 2 to 11 are implemented.

[98] For example, the computer program 591 may include instructions for executing an operation of modeling each of collision-prone parts of the robot as a three-dimensional shape having a line segment and a circular cross-section, an operation of calculating a shortest distance between a first line segment of a first collision-prone part and a second line segment of a second collision-prone part among the modeled collision-prone parts, an operation of calculating a relative Jacobian between a first point on the first line segment and a second point on the second line segment connecting a shortest distance, and an operation of calculating a speed for avoiding collision between the first and second collision-prone parts using the size of the three-dimensional shape, the shortest distance between the first and

second line segments, and the relative Jacobian between the first and second points.

[99] When the computer program 591 is loaded onto the memory 530, the processor 510 may perform the methods and/or operations according to various embodiments of the present invention by executing the one or more of the instructions.

[100] The methods according to the present invention described above can be manufactured as a program for being executed on a computer and may be stored in a computer-readable recording medium. Examples of the computer-readable recording medium include ROM, RAM, CD-ROM, a magnetic tape, a floppy disk, an optical data storage device, and the like.

[101] The computer-readable recording medium may be distributed to computer systems connected through a network, and computer-readable codes may be stored and executed in a distributed manner. In addition, function programs, codes, and code segments for implementing the method may be easily inferred by programmers in the art.

[102] Although the preferred embodiments of the present invention have been illustrated and described above, the present invention is not limited to the specific embodiments described above, and various modifications may be made by those skilled in the art without departing from the gist of the present invention as claimed in the claims. Furthermore, such modifications should

not be individually understood from the technical spirit or prospect of the present invention.

CLAIMS

[Claim 1] A control method for self-collision avoidance of a robot having multiple joints, the method comprising the steps of:

modeling each of collision-prone parts of the robot as a
5 three-dimensional shape having a line segment and a circular cross-section;

calculating a shortest distance between a first line segment of a first collision-prone part and a second line segment of a second collision-prone part among the modeled collision-prone parts;

10 calculating a relative Jacobian between a first point on the first line segment and a second point on the second line segment connecting the shortest distance, by applying a Jacobian matrix to the first and second points; and

calculating a speed for avoiding collision between the first
15 and second collision-prone parts using a size of the three-dimensional shape, the shortest distance between the first and second line segments, and the relative Jacobian between the first and second points.

20 [Claim 2] The method according to claim 1, wherein the collision-prone part includes a link portion that moves by a joint provided in the robot.

[Claim 3] The method according to claim 1, wherein the three-dimensional shape is formed using a set of points at a predetermined distance from the line segment.

5 [Claim 4] The method according to claim 1, wherein the relative Jacobian is a difference between a first Jacobian value obtained by applying a Jacobian matrix to the first point and a second Jacobian value obtained by applying a Jacobian matrix to the second point.

10

[Claim 5] The method according to claim 4, wherein the relative Jacobian represents a relative velocity between the first and second collision-prone parts with respect to the first and second points.

15

[Claim 6] The method according to claim 1, wherein the step of calculating a speed includes the step of subtracting a radius of the three-dimensional shape of the first collision-prone part and a radius of the three-dimensional shape of the second collision-prone part from the shortest distance between the first and second
20 line segments.

[Claim 7] The method according to claim 6, wherein when a result of the subtraction has a negative value, the control method changes

a speed of at least one among the first and second possible collision parts, and when a result of the subtraction has a positive value, the control method does not change the speed of the first and second possible collision parts.

5

[Claim 8] The method according to claim 6, wherein the step of calculating a speed includes the step of calculating a velocity component between the first and second collision-prone parts in a direction from the first point to the second point using the relative Jacobian between the first and second points.

10

[Claim 9] The method according to claim 7, wherein the step of calculating a speed includes the step of obtaining a speed of at least one among the first and second collision-prone parts, for calculating a velocity component in a direction of moving the first point and the second point away from each other using a result of subtraction.

15

[Claim 10] The method according to claim 1, wherein the calculated speed is a motor angular velocity of at least one among a plurality of joints provided in the robot.

20

[Claim 11] A computer program stored in a computer-readable recording medium in combination with hardware to execute the method of any one of claims 1 to 10.

5 [Claim 12] A robot that performs the method of any one of claims 1 to 10.

[Claim 13] A robot having a self-collision avoidance function, the robot comprising:

10 a processor;
 a memory for loading a computer program executed by the processor; and
 a storage for storing the computer program, wherein
 the computer program includes instructions for executing an
15 operation of modeling each of collision-prone parts of the robot as a three-dimensional shape having a line segment and a circular cross-section, an operation of calculating a shortest distance between a first line segment of a first collision-prone part and a second line segment of a second collision-prone part among the
20 modeled collision-prone parts, an operation of calculating a relative Jacobian between a first point on the first line segment and a second point on the second line segment connecting a shortest distance, and an operation of calculating a speed for avoiding collision between the first and second collision-prone parts using

the size of the three-dimensional shape, the shortest distance between the first and second line segments, and the relative Jacobian between the first and second points.