Paper:

On-Line Collision Avoidance of Two Command-Based Industrial Robotic Arms Using Advanced Collision Map

Ahmad Yasser Afaghani and Yasumichi Aiyama

Graduate School of Systems and Information Engineering, University of Tsukuba
1-1-1 Tennodai, Tsukuba, Ibaraki 305-8573, Japan
E-mail: a.y.afaghani@ieee.org, aiyama@esys.tsukuba.ac.jp

[Received December 26, 2013; accepted March 17, 2014]

This research aims to build a system for detecting and avoiding collisions between two industrial robotic arms that are controlled using point-to-point commands in on-line mode. Both robots have no prior knowledge of the commands which will be sent after starting the system. For this purpose, a collision map method has been improved to detect potential collisions between the robots and represent them as a collision area on the map. Commonly, industrial robotic arms have a cylindrical or a near-rectangular shape. The links of the robots have been approximated geometrically by using the swept sphere volume which presents tight modelling. Moreover, it is extremely easy to check for collisions and, therefore, feasible for on-line applications. To produce a collision-free trajectory for the robot, scheduling of the command execution time is necessary to avoid any collision areas on the map. The system has been tested on an OpenGL-based simulator to demonstrate the effectiveness of the system.

Keywords: on-line collision avoidance, point-to-point commands, advanced collision map, swept sphere volume, industrial robot arm

1. Introduction

Factory automation has been widely increasing through the use of industrial robots. Most applications rely on many robots operating in the same workspace, rather than having one robot in an ad hoc space. This increases productivity and reduces that amount of space needed in the factory. Thus, collision avoidance should be considered carefully to avoid potential collisions which may occur between the robots. Here, it is worth mentioning that motion planning of robots can be categorized into two types:

1. *Off-line planning*, in which all the commands and motions are decided prior to running the system [1]. This type of planning requires highly-structured environments, which are exhaustive and expensive to construct.
2. *On-line planning*, in which unpredictable commands are sent to the system during operation [2]. In contrast to *off-line* planning, this is better suited to industrial applications that operate in unstructured environments.

Many concepts have been suggested by researchers for collision avoidance in different systems. Lee et al. [3] and Chang et al. [4] suggested a collision map scheme to detect potential collisions between two robots. To avoid collisions, a time scheduling method of travelling speed along a planned trajectory has been proposed. Bien et al. [5] and Lee [6] proposed a method whereby the time-optimal trajectory of each robot is independently planned under the constraints on actuator torques and velocities. Lee et al. [7] describes the safety arc concept for avoiding collision between two 2-degrees-of-freedom planner manipulators. Hwang et al. [8] proposed a collision-free trajectory planning method based on a speed alternation strategy for multi-joint manipulators. Furthermore, a collision-trend index has been used to find an optimal resolution with the proposed method.

On the other hand, many methods have been developed for on-line collision avoidance. Katib [9] presented a unique method of obstacle avoidance for manipulators and mobile robots that relies on an artificial potential field concept. Kalaycioglu et al. [10] suggested a modified impedance control concept for solving the collision problem. Cheng [11] presented a new approach that is based on planned collision-free paths for independent tasks using a 2D geometric model in consideration of the swept region by the robot arms. Cellier et al. [12] proposed the reflex action theory. For example, when the second arm happens to collide with the first arm, the shape of the protection zones is modified. Freund et al. [13] presented a new method called "collision avoidance in real-time environment" (CARE) that aims to avoid collisions by changing the predetermined path of an endangered robot by using the redundant kinematics of that robot. Bosscher et al. [14] proposed an algorithm for collision avoidance by creating a set of linear inequality constraints on the commanded velocity of the robot by formulating joint limits and potential collisions. In tele-operated robots like those used in medical and military applications, on-line collision avoidance has been considered and studied extensively as well [15, 16].

Meanwhile, point-to-point (PTP) command-based control is widely used for industrial robots. The controller of these robots is a black box. Therefore, to implement collision avoidance for such robots, it is not possible to modify the system design of the controller. So, a few number of research papers has tackled that problem in on-line. Zhou et al. [17] proposed a zone-blocking method. When one robot enters a common workspace, a flag is activated to prevent other robots from entering the same space. The method is simple and safe but not very efficient.

This paper addresses on-line collision avoidance for two robot manipulators considering the whole body. The robots are controlled using unpredictable PTP commands that are sent after starting the system. Therefore, the collision map method, which has been developed by Lee et al. [3] for detecting collisions only between the end-effectors (EEFs) of two robot arms, has been improved to include all of the links of the robot body. In this work, an algorithm has been developed assuming that there are no obstacles in the workspace, and that the paths of the robot arms are strictly straight lines.

The rest of this paper is organized as follows: Section 2 presents a system overview and defines the problem. Section 3 explains robot arm modelling using the swept sphere volume. The core of this research is described in Section 4, specifically, the design of an advanced collision map for the purpose of collision detection between two robot arms. Thereafter, Section 5 describes a collision avoidance algorithm using an advanced collision map. Section 6 presents some experiments that have been executed using an OpenGL-based simulator. Finally, Section 7 summarizes the main points of the research.

2. Problem Formulation

Many industrial applications that use robot arms are controlled by PTP commands. Some applications acquire these commands on the fly. This is usually because the arms operate in an unstructured environment e.g., bin picking applications. In consequence, no decision is made in advance for these commands. However, having more than one robot operating in the same workspace introduces the problem of collision between these robots. Therefore, there should be a means of on-line planning for detecting and avoiding potential collisions in a proper way.

The collision problem has been investigated for two industrial robot arms that are controlled by PTP commands. The structure of collision avoidance system consists of three essential modules. The first module is the *Application Module*, which provides PTP commands to robots R1 and R2 in sequence. These commands are acquired by the *Collision Avoidance Module* one after the other. This module is called the planner. It is responsible for checking the possibility of collisions between the two arms on the basis of received commands as well as the current data acquired from the robots. Subsequently, the commands are

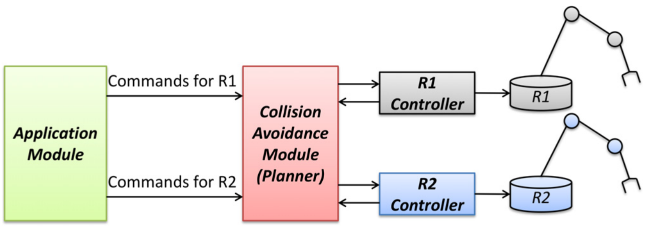


Fig. 1. Overview of collision avoidance system.

reprocessed to avoid any potential collisions before sending the commands to the robot. Finally, the *Robot Controller Modules* are responsible for controlling the robots and providing the planner with the necessary information from both robots, such as their posture, speed, and acceleration in order to perform collision detection. The system modules are illustrated in **Fig. 1**.

3. Geometric Modelling of Robot Manipulator

Modelling of the robot arm plays a significant role in detecting possible collisions between robots. The model should be precise while being based on a simple mathematical representation to minimize the calculation cost. For this to be feasible in an on-line system, many researchers have suggested different modelling techniques, as described below.

In spherical modelling, two parameters are needed, namely, the center and the radius of the sphere. Owing to the rotational invariance of the sphere, the modelling is computationally simple but wasteful of volume. Cylindrical modelling involve an elegant shape whereby each link is covered with a tube, but the mathematical expressions are complicated in 3D space. Polyhedral modelling, which represents each link by a huge number of polygons, provides a very accurate representation but is computationally intensive. Some other modelling methods using the bounding box volume (BBV) concept have been utilized. Discrete-orientation polytopes (DOP) has been utilized for deformed objects but are not useful for rigid bodies such as robot arms. On the other hand, the oriented bounding box (OBB) is suitable for a rigid body and offers very tight modelling and good rotational movement, except that mathematical computation is very intensive.

In this study, the robot arms are modelled using the *Swept Sphere Volume* (SSV), which integrates both tightness and mathematical simplicity. SSV is a volume formed by moving a sphere of a certain radius on a specified primitive such as a point, line, or rectangle. By changing the primitive dimensions and radius of the sphere, it is possible to model any designated object as tight as possible. The SSV modelling method has an advantage of easiness of collision detection due to the rotational invariant characteristic of the sphere. This facilitates the finding of the shortest distance between the primitives and compares it with radii summation of the spheres which sweep on the primitives. Thus, the computational

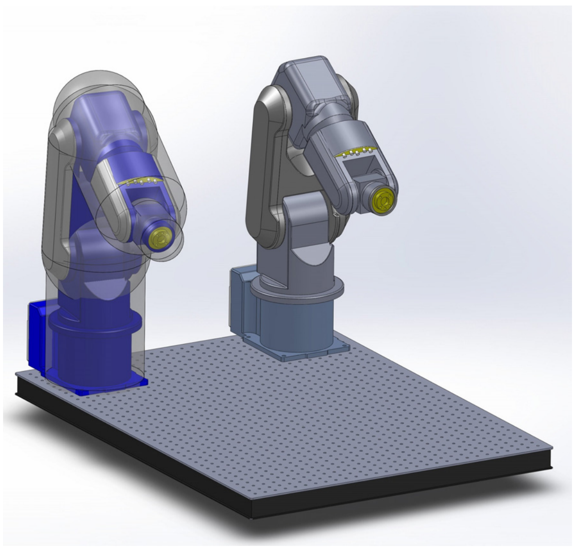


Fig. 2. Modelling of robot arms using swept sphere volume. This graph shows the modelling for one arm (best viewed in color).

cost for this model is low, which makes it appropriate for on-line applications.

Since most industrial robot arms have a near-tube shape, a line primitive is utilized for modelling the whole body of the robots. This research uses two robots, each one is constructed of a base, two links, and an end-effector (EEF). Thus, each robot is modelled using four line segments with appropriate radii of swept spheres. The robots' CAD design is shown in **Fig. 2** with modelling of the left robot using SSV with four line primitives.

4. Collision Detection

Collision avoidance algorithm consists of four essential stages, namely, command acquisition, collision detection, command scheduling, and command execution. Each robot uses the same algorithm which operate in sequence with each other. A brief flow chart of this algorithm is shown in **Fig. 3**.

The detection of collisions between both robots is initiated each time the on-line planner acquires a new PTP command for one of the robots. Thus, that robot's motion is not restricted due to the possibility of managing the command. The other robot becomes a moving or non-moving obstacle for the first one.

Hence, we define two modes for the robot. When the robot executes a PTP command and adheres to its path, the motion is restricted and unchangeable. In this case, the robot is denoted as *Moving Robot* (MR). On the other hand, the robot is denoted as *Standby Robot* (SR) as long as it is not moving. This includes the stages of command acquisition, collision detection, and command scheduling. Here, three different combinations of robot modes can be obtained, MR-MR, SR-SR, and SR-MR. With the MR-MR combination, no collision detection is needed because

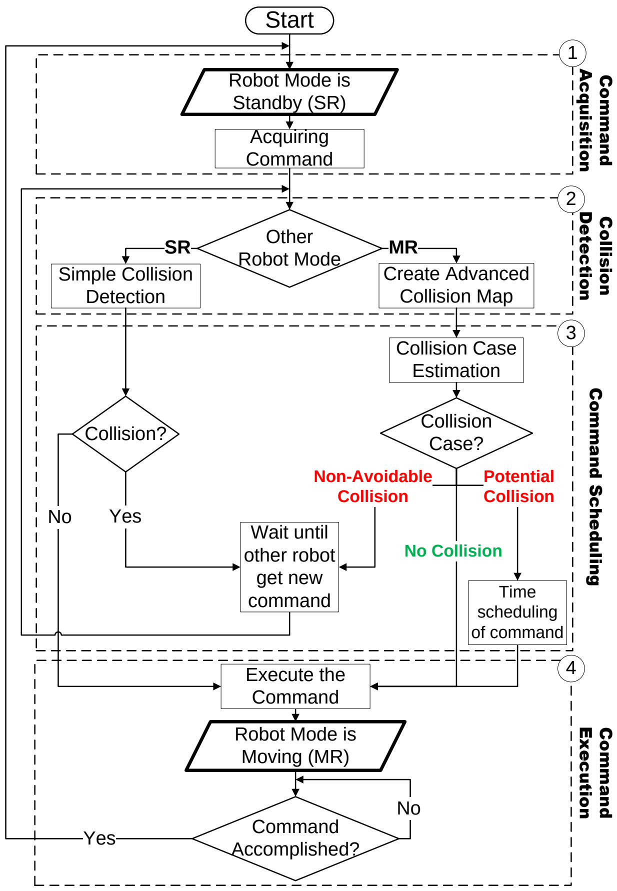


Fig. 3. Flow chart of collision avoidance algorithm.

the algorithm is designed to execute a PTP command for any robot after confirming that there is no risk of collision on the path.

4.1. SR-MR Combination

The collision map method [3] has been developed for representing potential collisions between the EEFs of two robots under their original trajectory information. Collisions in 3D coordinate space are converted into an area in 2D coordinate space which is composed of travelling lengths with the corresponding servo time of the robot that has not yet executed its command. The area is obtained from the path and trajectory information of the two robots.

Since the method is able to reveal potential collisions between two robots so it is appropriate for our system, but it is ineffective when the whole bodies of the robots need to be considered. Therefore, in this study, the collision map method has been improved to overcome this weakness.

Assume there are two robots $R1$ and $R2$ in the workspace. At the moment of acquiring a new PTP command for $R2$ by the planner, robot $R1$ is committed to its path, so it is MR, where $R1$'s EEF is moving in a straight line from $S1$ to $E1$ with path length l_{MR}, and the time needed to reach the target is t_{tar}^{MR}. While $R2$ is SR for which the EEF is supposed to move according to the new

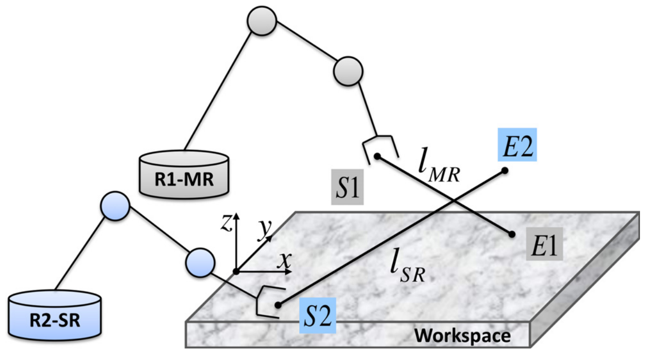


Fig. 4. Paths of robots in shared workspace.

PTP command from $S2$ to $E2$ with path length l_{SR}, and the requisite travelling time t^{SR}_{tar}. A simplified overview of the workspace with two robot arms and paths is shown in **Fig. 4**.

Let us denote the SSV line segments that model $R1$ and $R2$ by seg^{MR}_j and seg^{SR}_i, respectively. Where, $i \in [1,m]$ and $j \in [1,n]$ are indicated by segment number, and n and m are the maximum number of segments for $R1$ and $R2$, respectively. To detect potential collisions between the whole bodies of both robots, it is necessary to check each segment pair of the MR and SR under their original trajectory information. To detect collisions between a pair of segments, first, a minimum distance between both segments is calculated as $d_{min}(seg^{SR}_i, seg^{MR}_j)$. Then, the minimum distance is compared with the summation of the swept spheres radii for both segments. If that distance is equal to or less than the result of the summation, then, there is a collision between two segments.

$$d_{min}\left(seg^{SR}_i, seg^{MR}_j\right) \leq r_i + r_j \quad . \; . \; . \; . \; . \; . \; . \quad (1)$$

where, r_i and r_j are the radii of the spheres which sweep on seg^{SR}_i and seg^{MR}_j, respectively.

Here, we define a new collision mapping concept. It relates the servo time of the MR with the travelling time of the SR. It assumes that the SR starts to move as soon as the planner acquires the command, so, the SR will move from initial position to target position within a time period of t^{SR}_{tar}. To design the aforementioned map, a collision is tested every sampling time dt_s along servo time of the MR $t \in [0, t^{MR}_{tar}]$. Thus, at time t, the MR's segment seg^{MR}_j has a specific posture which can be acquired by the inverse kinematic. This segment posture is checked with the SR's segment seg^{SR}_i for all postures every time sample along $\tau \in [0, t^{SR}_{tar}]$. As a result, ranges of collision timings are obtained as $T_{ij}(t,1), T_{ij}(t,2), \ldots, T_{ij}(t, rng_{ij}(t))$. Where, $rng_{ij}(t)$ is the maximum number of ranges at time t for a checked pair (ij). The range at time t for pair (ij) is referred to as a continuous collision and starts from $\tau^{str}_{ij}(t,k)$ and ends at $\tau^{end}_{ij}(t,k)$ whereas $k \in [1, rng_{ij}(t)]$ refers to the range number. After finishing all collision checks for every pair of segments, $m \times n$ collision range results are acquired, as listed in **Table 1**. An illustrative graph of the new collision mapping of pair (ij) is shown in **Fig. 5**. Afterwards, for determining the potential collisions between the whole bodies of both robots, a union of the ranges every time sample within $t \in [0, t^{MR}_{tar}]$ for all pairs is calculated as follows:

Table 1. Ranges of collision timings obtained between each pair of segments of MR and SR.

	seg^{MR}_1 (Base)	seg^{MR}_2 (Link1)	$\cdots$	seg^{MR}_n (EEF)
seg^{SR}_1 (Base)	$T_{11}(t,k)$	$T_{12}(t,k)$	$\cdots$	$T_{1n}(t,k)$
seg^{SR}_2 (Link1)	$T_{21}(t,k)$	$T_{22}(t,k)$	$\cdots$	$T_{2n}(t,k)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
seg^{SR}_m (EEF)	$T_{m1}(t,k)$	$T_{m2}(t,k)$	$\cdots$	$T_{mn}(t,k)$

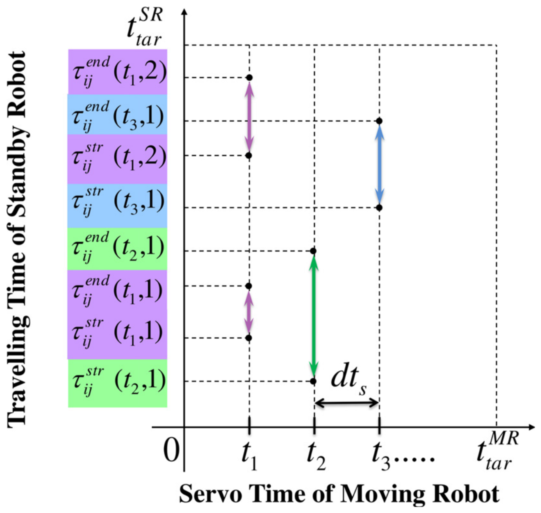


Fig. 5. Map for presenting the ranges of collision timings between segment j of MR and segment i of SR (best viewed in color).

$$\bigcup_{i=0}^{i=m} \bigcup_{j=0}^{j=n} \bigcup_{k=0}^{k=rng_{ij}(t)} T_{ij}(t,k) \longrightarrow \begin{bmatrix} T_{Total}(t,1) \\ T_{Total}(t,2) \\ \vdots \\ T_{Total}(t, rng_{Total}(t)) \end{bmatrix} . \quad (2)$$

The result is a group of total ranges of collision timings $T_{Total}(t,1), T_{Total}(t,2), \ldots, T_{Total}(t, rng_{Total}(t))$. Where, $rng_{Total}(t)$ is the maximum number of total ranges at time t. In addition, each total range has a boundary which starts from $\tau^{str}_{Total}(t, k_{Total})$ and ends at $\tau^{end}_{Total}(t, k_{Total})$ whereas $k_{Total} \in [1, rng_{Total}(t)]$ refers to the total range number.

Since the original collision map design is related to the travelling length of EEF corresponding to the servo time, potential collisions between EEFs can only be illustrated. It is necessary to convert the aforementioned ranges to ranges of collision lengths, i.e., convert from travelling time space to travelling length space. To this end, trajectory information of the SR is used. At every time sample

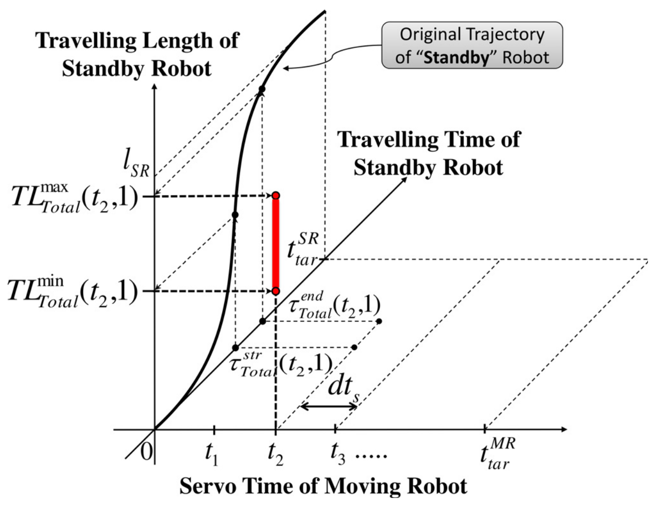

Travelling Length of Standby Robot
Original Trajectory of "Standby" Robot
Travelling Time of Standby Robot
l_{SR}
$TL^{max}_{Total}(t_2,1)$
$TL^{min}_{Total}(t_2,1)$
t^{SR}_{tar}
$\tau^{end}_{Total}(t_2,1)$
$\tau^{str}_{Total}(t_2,1)$
dt_s
0
t_1
t_2
t_3 .....
t^{MR}_{tar}
Servo Time of Moving Robot

Fig. 6. Method for changing from time space to travelling length space using trajectory information of standby robot EEF.

along $t \in [0, t^{MR}_{tar}]$, the range $T_{Total}(t, k_{Total})$ is used to get the travelling length of the EEF's trajectory only at start and end time $\tau^{str}_{Total}(t, k_{Total})$ and $\tau^{end}_{Total}(t, k_{Total})$, respectively:

$$TL^{min}_{Total}(t, k_{Total}) = f_{TL}(\tau^{str}_{Total}(t, k_{Total})) \quad . \; . \; . \quad (3)$$

$$TL^{max}_{Total}(t, k_{Total}) = f_{TL}(\tau^{end}_{Total}(t, k_{Total})) \quad . \; . \; . \quad (4)$$

where, $TL^{min}_{Total}(t, k_{Total})$ and $TL^{max}_{Total}(t, k_{Total})$ are the minimum and maximum travelling lengths of the range at time t, and f_{TL} is the function for calculating the travelling length of the SR at a desired time assuming that $f_{TL}(0) = 0$. A graph which relates the SR's travelling length to the servo time of the MR by means of travelling time of the SR is illustrated in **Fig. 6**. It is shown how to obtain a range of collision lengths at time t, where the SR's trajectory is shown as a curved line and the range is shown with a straight bold line. By drawing all the ranges on that graph, an area called the *Collision Area* will be formed as shown in **Fig. 7**. This figure is a graph of the final *Advanced Collision Map* which relates the SR's travelling length to the servo time of the MR. The final arrival time of the MR t^{MR}_{tar} might be smaller, equal to, or larger than the final arrival time of the SR t^{SR}_{tar}. Here, the graph depicts the original trajectory of the SR passing through the collision area assuming that $t^{MR}_{tar} > t^{SR}_{tar}$. It is notable that the advanced collision map has devolved to resemble the original design of the map [3], and the collision area in the advanced collision map includes all the information on potential collisions between the whole bodies of the robots while the original map is limited to information on collisions between the EEFs. The avoidance method of that area, is elucidated in the next section.

4.2. SR-SR Combination

Let us assume that both robots are SRs and are waiting for commands. If the planner acquires a PTP command

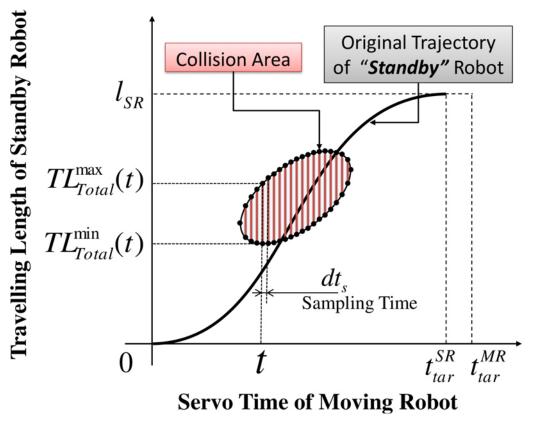


Fig. 7. Advanced collision map showing collision area with original trajectory of standby robot assuming that $t^{MR}_{tar} > t^{SR}_{tar}$.

for robot $R1$, then the other robot $R2$ becomes a static object in front of $R1$. Therefore, it is necessary to check collisions between both robots under the original trajectory of $R1$ only.

In this case, there is no need to create a collision map, given that the collision detection algorithm of the SR-MR combination is sufficient. However, the detection is only performed between the current posture of $R2$ and all postures of $R1$ for all of the travelling time samples under the original trajectory. As a result, if any collision is detected, it will be classified as an inevitable collision which cannot be avoided if $R1$ executes the command. Therefore, for determining that the path of $R2$ is not free of collisions, i.e., $R1$ becomes an obstacle to $R2$, it is sufficient to detect at least one collision between any pair of segments at any time.

5. Collision Avoidance

Collision avoidance methods can be divided into two categories:

- *Time scheduling method*: modifies the trajectory while fixing the geometric path.
- *Path modification method*: modifies the geometric path of the robot.

This work regards PTP commands, which are sent by the application module, as being of high priority and therefore considers that PTP commands should not be changed. Thus, the goal is to avoid any potential collisions between the robots which may occur if those commands are executed, assuming that no robots collide with any stationary objects in the shared workspace. Therefore, time scheduling method can be adopted.

With the SR-MR combination, to avoid the collision area on an advanced collision map, the original trajectory of the SR is shifted by dt_{cf}, which is the necessary delay time required to produce a collision-free path for the SR. Thus, the delay time is calculated as follows: reference to the previous assumption where $R1$ is MR which adheres

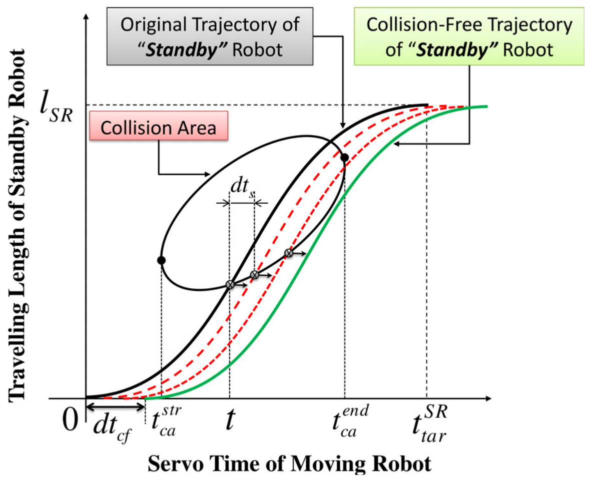


Fig. 8. Description of calculating necessary delay time to avoid the collision area.

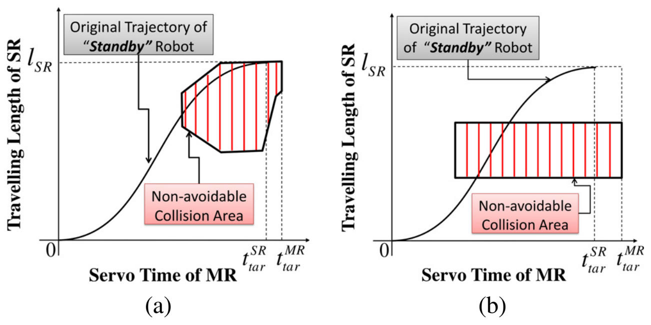


Fig. 9. Examples of non-avoidable collision areas.

to its original trajectory, and $R2$ is SR which must modify its trajectory to avoid the collision area, assuming an initial value of delay time $dt_{cf} = 0$. the method starts by getting the start and end times of the collision area which are t_{ca}^{str} and t_{ca}^{end}, respectively. To check whether the trajectory of $R2$ passes through the area within that period, the travelling length of $R2$'s trajectory by means of the function $f_{TL}(t)$ is calculated at every time sample dt_s:

$$TL(t-dt_{cf}) = f_{TL}(t-dt_{cf}) : t \in [t_{ca}^{str}, t_{ca}^{end}], t_{ca}^{str} \geq dt_{cf} \quad \ldots \ldots (5)$$

Subsequently, $TL(t-dt_{cf})$ is compared with all available ranges of the collision lengths at time t. The range is determined with the minimum and maximum lengths, which are $TL_{Total}^{min}(t, k_{Total})$ and $TL_{Total}^{max}(t, k_{Total})$ as described in Eqs. (3) and (4). Therefore, a collision exists if $TL(t-dt_{cf})$ is within the range defined as follow:

$$TL_{Total}^{min}(t, k_{Total}) \leq TL(t-dt_{cf}) \leq TL_{Total}^{max}(t, k_{Total}) \quad (6)$$

If any collision is detected, the original trajectory of $R2$ will be shifted by one time sample $1 \times dt_s$, and the delay time is increased by $dt_{cf} = dt_{cf} + dt_s$. The next step is a repetition of the last steps but this time the check is done with a new position of travelling length considering that, after each shift, if $t_{ca}^{str} < dt_{cf}$, the time range check is reduced to become $t \in [t_{ca}^{str} + dt_s, t_{ca}^{end}]$. Thus, by continuing to shift the trajectory every time a collision is detected, we get a final collision-free trajectory of $R2$ as illustrated by a curved line in the rightmost of **Fig. 8**.

The output of the aforementioned method is the delay time dt_{cf} for which there are three possible cases:

1) $dt_{cf} = 0$: means that there is no potential collision between the robots.

2) $dt_{cf} > 0$: means that there is a potential collision, and the command execution should be postponed.

3) $dt_{cf} = \infty$: means there will be an inevitable collision if the command is executed directly.

The third case may happen if the original trajectory of $R2$, which is SR, passes through the collision area which has an end time $t_{ca}^{end} = t_{tar}^{MR}$ as shown in the examples in **Fig. 9**. Here, $R2$ is put in a pending situation until $R1$ accomplishes the motion and acquires a new PTP command. Then, we get the SR-SR combination, which was addressed in the last section, with a priority for $R1$ to execute the command.

Since the collision map is built while the system is operating, it is necessary to consider the time of building the map dt_m to calculate the final delay time for obtaining collision-free trajectory of $R2$. To this end, a software timer has been added, and thus, the final delay time required to avoid the collision area in on-line mode is: $dt = dt_{cf} - dt_m$.

With the SR-SR combination, if a potential collision is detected between both robots, it will be unavoidable. Thus, the delay time is set to be infinite $dt_{cf} = \infty$, and the matter is devolved as described before. In this paper, we do not address how to avoid a collision in such a case, and the algorithm is limited to put the robot with a PTP command in a pending situation until the other robot acquires a new PTP command.

6. Simulation Results

The collision avoidance system has been tested and evaluated with two robots using an OpenGL-based simulator. The simulator can emulate the motion of any robot by providing a 3D-CAD model of the robot, in addition to inserting the joint angles of the robot every time sample of motion. In this work, the simulator is used to imitate two Yaskawa Motoman robots (HP3J and UPJ). The robots are fixed on a board, with a distance between the central axis of the robots' bases being 500 mm. The global coordinate system is located in the middle between both robot bases. A snapshot of the simulator is shown in **Fig. 10**.

The experiment is divided into two parts. The first part is performed without a collision avoidance module, so the commands are sent directly to the robot controller, after which the motion is monitored. The second part is performed with a collision avoidance module. Let us assume that $R1$ is the gray color robot and $R2$ is the blue robot, and the maximum velocity and acceleration of each

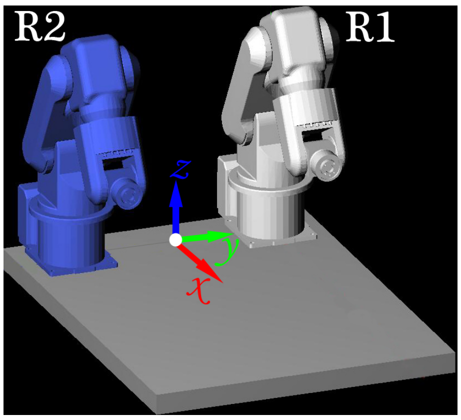
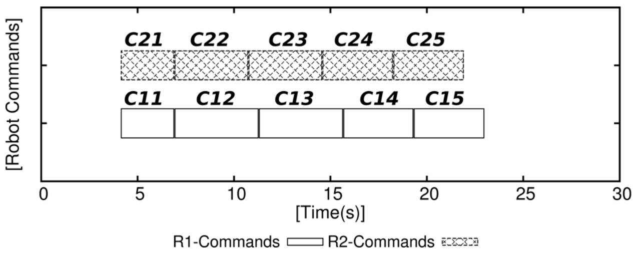


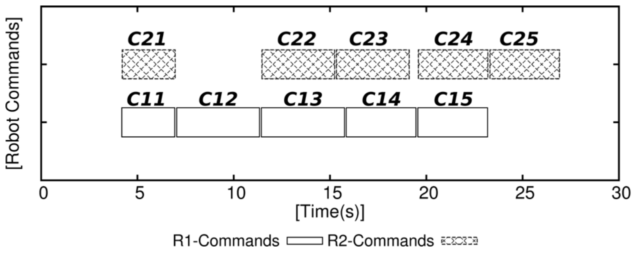
Fig. 10. Snapshot of Open-GL based simulator that shows two robots in their initial posture (best viewed in color).

Table 2. PTP commands that are used in Exp1 and Exp2 for $R1$ and $R2$.

***R*1-Commands** **(*x*, *y*, *z*, roll, pitch, yaw)**	***R*2-Commands** **(*x*, *y*, *z*, roll, pitch, yaw)**
$C11$ $(300,250,300,0,0,0)$	$C21$ $(300,-250,300,0,0,0)$
$C12$ $(500,-50,360,0,-20,0)$	$C22$ $(500,0,250,0,-20,0)$
$C13$ $(300,250,300,0,0,0)$	$C23$ $(300,-250,300,0,0,0)$
$C14$ $(500,15,360,0,0,0)$	$C24$ $(500,-15,300,0,0,0)$
$C15$ $(300,250,300,0,0,0)$	$C25$ $(300,-250,300,0,0,0)$



(a) Exp1. Command chart before applying collision avoidance method.



(b) Exp2. Command chart after applying collision avoidance method.

Fig. 11. Command execution timing of $R1$ and $R2$.

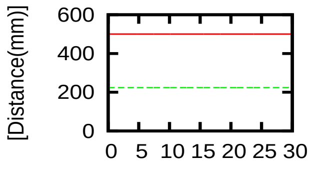
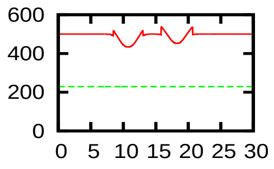
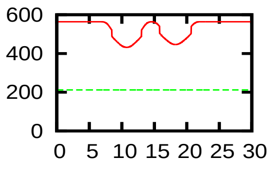
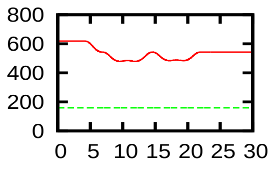
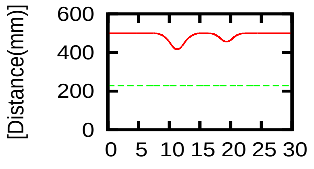
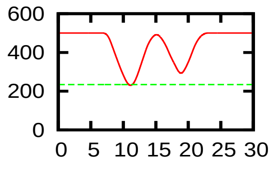
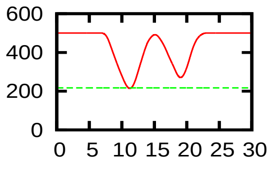
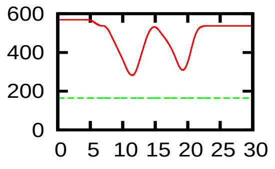
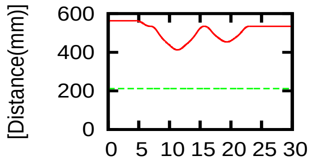
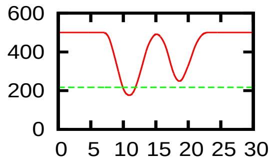
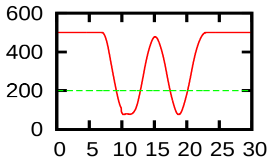
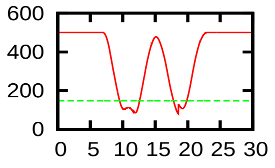
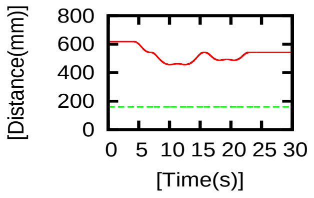
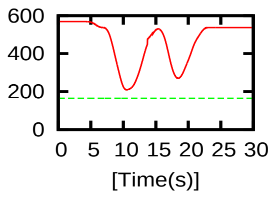
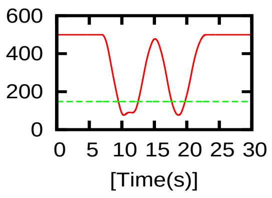
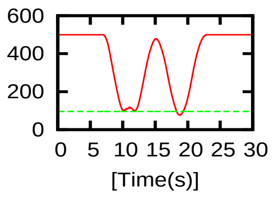
robot EEF are set as $V1_{max} = V2_{max} = 100$ mm/s and $a1_{max} = a2_{max} = 100$ mm/s^2, respectively. Both robots have the same length of links which are $l_{base\text{-}joint1} = 290$ mm, $l_{joint\text{-}joint2} = 260$ mm, $l_{joint2\text{-}joint3} = 270$ mm, and $l_{joint3\text{-}EEF} = 90$ mm. The radii of the spheres which sweep on the line primitives that model the robot are $r_{base} = 112$ mm, $r_{link1} = 117$ mm, $r_{link2} = 100$ mm, and $r_{EEF} = 48$ mm. Thus, PTP commands are sent to the robots as shown in **Table 2**. Each command includes information of EEF posture according to the global coordinate system $[x$ [mm], y [mm], z [mm], roll [deg], pitch [deg], yaw [deg]$]$. The initial posture of each robots is assumed to be $R1_{init}$ $(450,250,300,0,0,0)$ and $R2_{init}$ $(450,-250,300,0,0,0)$.

In the first part of the experiment, the commands are sent directly to the controller. The execution timing of the commands from the startup of the system are illustrated in **Fig. 11(a)**. It is worth mentioning that the period before the execution of the first command is that time required to initialize the system. The minimum distance between each pair of segments of the robots is tracked, and the results for all possibilities of pairs are acquired as shown in **Table 3**. The distance curve is shown as a solid line, and the prohibited distance, which is the radii summation of spheres that model the tracked segments, is shown as a dashed line. We can notice that several curves have passed through the prohibited distance, which means that collisions are inevitable.

The second part of the experiment has been done after including the collision avoidance module. The execution timing of the commands after applying the collision avoidance method are shown in **Fig. 11(b)**. The results of the distance curve of each pair are illustrated in **Table 4**. It is observable that all the curves are navigating away from the prohibited distance. This means that the system can successfully avoid all potential collisions.

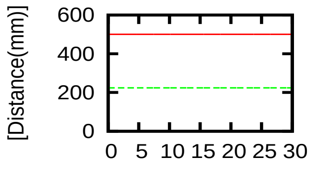
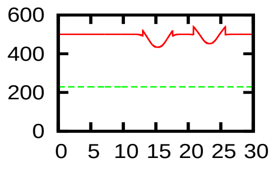
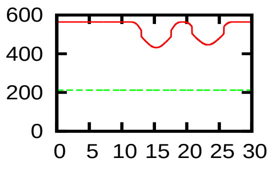
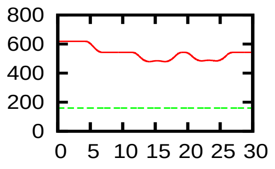
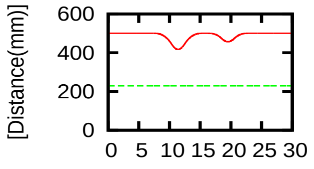
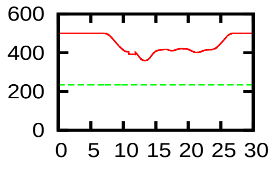
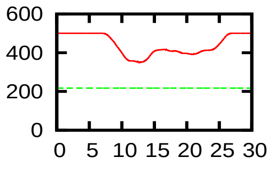
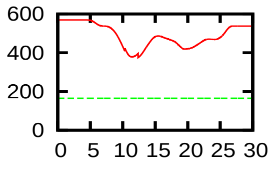
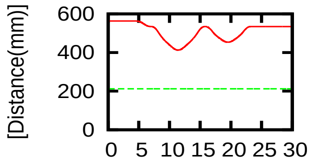
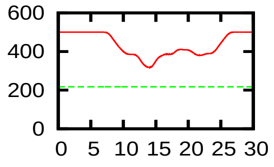
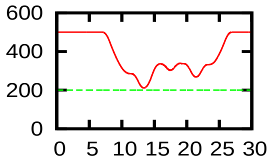
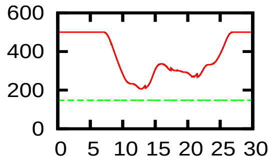
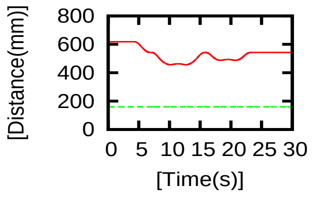
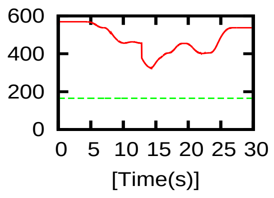
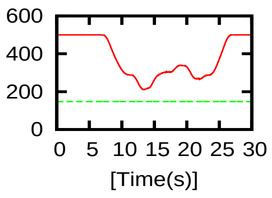
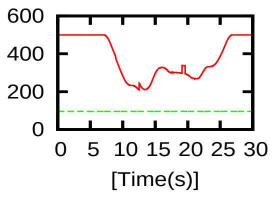
By evaluating our proposed system, it become obvious that the system can overcome the previously discussed (Section 1) limitation of the collision detection method proposed by Lee et al. [3]. The original collision map devised by Lee illustrates the information on potential collisions between only the EEFs of both robots, whereas in the proposed system, the original map has been improved to have an advanced collision map. The new map has the same characteristics as the original map in addition to the ability to show potential collisions between the whole bodies of the robots. A comparison with the work of Zhou et al. [17, 18], a system which has the same problem formulation as our work, shows that the proposed system proves an advantage in terms of time efficiency. The first method proposed by Zhou [17], which is zone-blocking, is simple and safe but large amounts of time are wasted due to the inability of the robot to enter the workspace if another robot is operating in it. The second method proposed by Zhou [18] is more advanced, being based on dividing the workspace into small partitions. At every time sample, the algorithm reserves those partitions that are occupied by the moving robot. Thus, the other robot cannot use those partitions. The experiment described in Zhou's paper [18] has been performed with the same configuration but using the proposed system. Eight PTP commands [18], which differ from those of the previous experiments, are sent to the system. The execution timing of the commands for every method are

Table 3. Exp1. Tracking results of minimum distance between the links of both robots before applying collision avoidance system (best viewed in color).

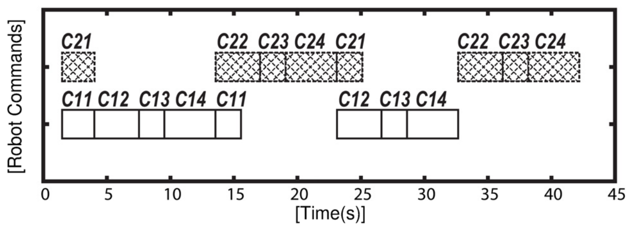
	seg_0^{MR} (base)	seg_1^{MR} (Link1)	seg_2^{MR} (Link2)	seg_3^{MR} (EEF)
seg_0^{SR} (Base)				
seg_1^{SR} (Link1)				
seg_2^{SR} (Link2)				
seg_3^{SR} (EEF)				

Minimum distance between two segments ——— Prohibited distance - - - - - -

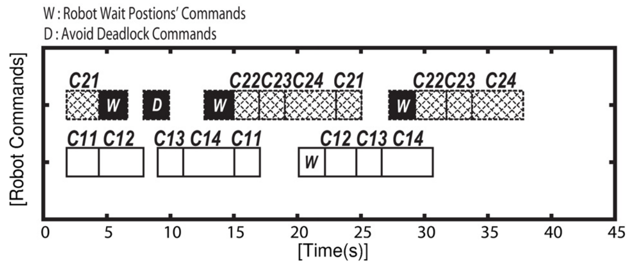
Table 4. Exp2. Tracking results of minimum distance between the links of both robots after applying collision avoidance system (best viewed in color).

	seg_0^{MR} (base)	seg_1^{MR} (Link1)	seg_2^{MR} (Link2)	seg_3^{MR} (EEF)
seg_0^{SR} (base)				
seg_1^{SR} (Link1)				
seg_2^{SR} (Link2)				
seg_3^{SR} (EEF)				

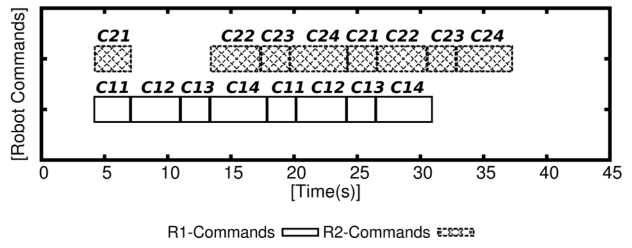
Minimum distance between two segments ——— Prohibited distance - - - - - -



(a) Command chart by applying previous first method [18]



(b) Command chart by applying previous second method [18]



(c) Command chart by applying the proposed method.

Fig. 12. Exp3. Command execution timing of $R1$ and $R2$.

Table 5. Comparison between current and previous methods.

	Previous First Method	**Previous Second Method**	**Current Method**
Method	Zone-blocking	Workspace-partitioning	Collision map
Modelling	None	Spheres	SSV
Time spent to execute commands in Exp3	40.7 s	35.9 s	33.1 s

illustrated in **Fig. 12**. A comparison between the proposed method and Zhou's methods is shown in **Table 5**.

7. Conclusion and Future Works

This paper has proposed a new solution for collision avoidance between two n-link robot manipulators in an on-line system. The robots are controlled using PTP commands and have no prior knowledge of commands to be sent. Thus, the collision map method has been improved for detecting potential collisions between the whole robot bodies. For the purpose of collision detection, the robot links are approximated geometrically using SSV with line primitives which is tight modelling for a near-tube robot shape with a low computational cost due to the ease of the intersection test. Collisions are avoided using time scheduling for the execution time of PTP commands in order to avoid the collision area. The simulation results have demonstrated a successful avoidance of the potential collisions as well as an advantage in terms of time efficiency relative to previous methods.

By addressing the optimality issue, the system has been designed to process PTP commands in sequence and then detect and avoid any potential collisions via the proposed algorithm. Therefore, the trajectory is rescheduled in the optimal way to avoid the collision areas on the map. On the other hand, if the condition whereby PTP commands are received has been changed, e.g., receiving more than one command at a time, the proposed method cannot be classified as being optimal.

Now, we are working on improving the current system to cope with the problem of deadlocks that may occur when both robots become obstacles to each other.

References:

[1] R. Zurawski and S. Phang, "Path Planning for Robot Arms Operating in a Common Workspace," Proc. IEEE Int. Conf. on Industrial Electronics, Control, Instrumentation, and Automation, Power Electronics and Motion Control, pp. 618-623, 1992.

[2] L. Tsai-Yen and J.-C. Latombe, "On-Line Manipulation Planning for Two Robot Arms in a Dynamic Environment," Proc. IEEE Conf. on Robotics and Automation, Vol.1, pp. 1048-1055, 1995.

[3] B. H. Lee and C. S. G. Lee, "Collision-Free Motion Planning of Two Robots," IEEE Trans. on Systems, Man, and Cybernetics, Vol.17, No.1, pp. 21-32, 1987.

[4] C. Chang, M. J. Chung, and B. H. Lee, "Collision Avoidance of Two General Robot Manipulators by Minimum Delay Time," IEEE Trans. on Systems, Man, and Cybernetics, Vol.24, No.3, pp. 517-522, 1994.

[5] Z. Bien and J. Lee, "A Minimum-Time Trajectory Planning Method for Two Robots," IEEE Trans. on Robotics and Automation, Vol.8, No.3, pp. 414-418, 1992.

[6] J. Lee, "A Dynamic Programming Approach to Near Minimum-Time Trajectory Planning for Two Robots," IEEE Trans. on Robotics and Automation, Vol.11, No.1, pp. 160-164, 1995.

[7] S. W. Lee, Y. S. Nam, K. D. Lee, and B. H. Lee, "A Safety Arc Based Collision Avoidance Algorithm of a Two-Arm Robot Manipulator," Proc. 35th SICE Annual Conf. Int. Session Papers, pp. 1167-1172, 1996.

[8] K.-S. Hwang, M.-Y. Ju, and Y.-J. Chen, "Speed Alteration Strategy for Multijoint Robots in Co-Working Environment," IEEE Trans. on Industrial Electronics, Vol.50, No.2, pp. 385-393, 2003.

[9] O. Khatib, "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots," Proc. IEEE Int. Conf. on Robotics and Automation, Vol.2, pp. 500-505, 1985.

[10] S. Kalaycioglu, M. Tandirci, and D. S. Nesculescu, "Real-Time Collision Avoidance of Robot Manipulators for Unstructured Environments," Proc. IEEE Int. Conf. on Robotics and Automation, Vol.1, pp. 44-51, 1993.

[11] X. Cheng, "On-Line Collision-Free Path Planning for Service and Assembly Tasks by a Two-Arm Robot," Proc. IEEE Int. Conf. on Robotics and Automation, Vol.2, pp. 1523-1528, 1995.

[12] L. Cellier, P. Dauchez, R. Zapata, and M. Uchiyama, "Collision Avoidance for a Two-Arm Robot by Reflex Actions: Simulations and Experimentations," J. of Intelligent and Robotic Systems, Vol.2, No.14, pp. 219-238, 1995.

[13] E. Freund and J. Rossman, "The Basic Ideas of a Proven Dynamic Collision Avoidance Approach for Multi-Robot Manipulator Systems," Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, Vol.2, pp. 1173-1177, 2003.

[14] P. Bosscher, D. Hendman, and P. Bay, "Real-Time Collision Avoidance Algorithm for Robotic Manipulators," Proc. IEEE Int. Conf. on Technologies for Practical Robot Applications (TePRA 2009), pp. 113-122, 2009.

[15] R. G. Beaumont and R. M. Crowder, "Real-Time Collision Avoidance in Two-Armed Robotic Systems," J. of Computer-Aided Engineering, Vol.8, No.6, pp. 233-240, 1991.

[16] A. Spencer, M. Pryor, C. Kapoor, and D. Tesar, "Collision Avoidance Techniques for Tele-Operated and Autonomous Manipulators in Overlapping Workspaces," Proc. IEEE Int. Conf. on Robotics and Automation, pp. 2910-2915, 2008.

[17] J. Zhou, K. Nagase, S. Kimura, and Y. Aiyama, "Collision Avoidance of Two Manipulators Using RT-Middleware," IEEE/SICE Int. Symp. on System Integration, pp. 1031-1036, 2011.

[18] J. Zhou and Y. Aiyama, "Efficient Collision Avoidance Method of Two Command-Based Manipulators Using Partitioned Workspace," 31th Annual Conf. of the Robotics Society of Japan (RSJ 2013), pp. 2-5, 2013 (in Japanese).

Name:
Ahmad Yasser Afaghani

Affiliation:
Ph.D. Candidate, Graduate School of Systems and Information Engineering, University of Tsukuba

Address:
1-1-1 Tennodai, Tsukuba, Ibaraki 305-8573, Japan

Brief Biographical History:

2005 Received B.Sc. from Faculty of Electrical and Electronic Engineering, University of Aleppo, Syria

2006- Technical Consultant, Tecnogamma Spa, Treviso, Italy

2008- Lecturer, Institute of Mechanical and Electrical Engineering, University of Aleppo, Syria

2009- Research Student, Intelligent Robot Laboratory, University of Tsukuba

2012 Received M.Sc. from Graduate School of Systems and Information Engineering, University of Tsukuba

Main Works:

- A. Y. Afaghani, Y. Yuta, and J. H. Lee, "Jacobian-matrix-based motion control of an omni-directional mobile robot with three active caster," Proc. IEEE/SICE Int. Symp. on System Integration (SII), pp. 627-633, Dec. 20-22, 2011.
- A. Y. Afaghani and Y. Aiyama, "On-line collision avoidance between two robot manipulators using collision map and simple escaping method," Proc. IEEE/SICE Int. Symp. on System Integration, pp. 105-110, Dec. 15-17, 2013.

Membership in Academic Societies:

- The Institute of Electrical and Electronics Engineers (IEEE) Robotics and Automation Society

Name:
Yasumichi Aiyama

Affiliation:
Associate Professor, Faculty of Engineering, Information and Systems, University of Tsukuba

Address:
1-1-1 Tennodai, Tsukuba, Ibaraki 305-8573, Japan

Brief Biographical History:

1995 Received Ph.D. from Graduate School of Engineering, The University of Tokyo

1995- Research Associate, Graduate School of Engineering, The University of Tokyo

1999- Assistant Professor, Graduate School of Systems and Information Engineering, University of Tsukuba

2004- Associate Professor, Graduate School of Systems and Information Engineering, University of Tsukuba

Main Works:

- T. Kyouso and Y. Aiyama, "Program Development Environment for Multiple Robot-Multiple Application System," J. Robotics and Mechatronics, Vol.18, No.5, pp. 572-579, Oct. 2006.
- S. Shindo, S. Tomita, and Y. Aiyama, "Realization of Pressfitting Operation by Impact Manipulation with an Under-Actuated Manipulator," Int. J. of Automation Technology, Vol.2, No.4, pp. 305-311, Jul. 2008.

Membership in Academic Societies:

- The Institute of Electrical and Electronics Engineers (IEEE) Robotics and Automation Society
- The Robotics Society of Japan (RSJ)
- The Japan Society of Mechanical Engineers (JSME)