Advanced Robotics

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/tadr20

Reactive motion planning using time-layered C-spaces for a collaborative robot PaDY

Hisaka Wada, Jun Kinugawa & Kazuhiro Kosuge

To cite this article: Hisaka Wada, Jun Kinugawa & Kazuhiro Kosuge (2021) Reactive motion planning using time-layered C-spaces for a collaborative robot PaDY, Advanced Robotics, 35:8, 490-503, DOI: 10.1080/01691864.2021.1896381

To link to this article: https://doi.org/10.1080/01691864.2021.1896381

ADVANCED ROBOTICS
2021, VOL. 35, NO. 8, 490–503
https://doi.org/10.1080/01691864.2021.1896381

FULL PAPER

OPEN ACCESS Check for updates

Reactive motion planning using time-layered C-spaces for a collaborative robot PaDY

Hisaka Wada, Jun Kinugawa and Kazuhiro Kosuge

Department of Robotics, Graduate School of Engineering, Tohoku University, Sendai, Japan

ABSTRACT

A reactive motion-planning for collaborative robots using the time-layered C-spaces (TLC-spaces) is proposed in this paper. First, the time-augmented C-space (TAC-space) is introduced. TAC-space is an implementation of the configuration-time space with multiple moving obstacles [Latombe JC. Robot motion planning. Kluwer Academic; 1991. p. 22, 23]. The TAC-space is obtained by stacking the current and predicted future C-spaces along the time axis using predicted motions of the obstacles. Then, TLC-spaces is constructed as the collection of only those C-spaces in the TAC-space that are relevant to the motion planning with moving obstacles. The trajectory that reaches the goal configuration at the specified target time is generated under dynamic constraints including robot velocity and acceleration. We focus on a collaborative robot, PaDY, whose task is to deliver tools and parts to the worker in a factory. Similar to an actual assembly process in an automobile production system, six scenarios are selected for the evaluation of the proposed motion planning method. The simulation results using the real-life motion of workers show that the computation time required for the proposed motion planning using TLC-spaces is shorter than that of our previous method using TAC-space. The experimental results show that the proposed method is applicable to PaDY in human environments.

ARTICLE HISTORY
Received 20 March 2020
Revised 20 January 2021
and 11 September 2020
Accepted 30 January 2021

KEYWORDS
Human–robot collaboration; collision avoidance; sampling-based path planning

1. Introduction

Collaborative robots, which assist human workers, have been developed and used to increase the productivity of the manufacturing processes that cannot be easily automated using industrial robots. The concept of collaborative robots was proposed in a medical application using a passive robot arm to guarantee safe operation of the robot in 1993 [1]. In 1996, a collaborative robot, cobot, was invented by J. Edward Colgate and Michael A. Peshkin based on the passive robotics [2, 3].

Cobots were designed with a view to constraining the motion of robots along a desired trajectory using passive robotics [3]. It is intrinsically safe because its motion is generated by the force/moment applied to it by its operator. Several cobot systems for automobile assembly processes, including a floor-based passive door unloader cobot, were designed and proposed in the late 1990s [4].

In 1999, Y. Yamada et al. proposed Skill-Assist, a robot system to assist human workers in the handling of heavy objects [5]. It has been used in automobile assembly processes. The weight of the object is completely supported by the robot, and the system is operated by the intention force/moment applied to it. Skill-Assist has entered the market and used in many automobile factories [6].

PaDY (in-time *Pa*rts/tools *D*elivery to *Y*ou robot) (see Figure 1), a collaborative robot, was proposed in 2010 [7, 8]. It supports a worker executing assembly-line tasks under a vehicle body by delivering necessary tools and parts to the worker. It predicts the progress of the work being performed by a worker and delivers tools and parts to the worker when he/she needs them [9–12]. It was installed in an automobile assembly line, and its effectiveness for a real-life assembly process was also demonstrated [7].

ISO 10218-1 for the safety requirements of industrial robots has been amended, and ISO 10218-1 for robots and ISO 10218-2 for robot systems and integration were established in 2011 [13, 14]. ISO/TS 15066, the technical specification for collaborative robots, was published in 2016 [15]. These standards have accelerated the commercialization of collaborative robots. Many companies have put collaborative robots into the market, for example, LBR iiwa [16] and LBR iisy [17] by KUKA, YuMi

CONTACT Hisaka Wada hisaka.wada@ieee.org Department of Robotics, Graduate School of Engineering, Tohoku University, 6-6-01 Aoba, Aramaki, Aoba-ku, Sendai 980-8579, Japan

Supplemental data for this article can be accessed here. https://doi.org/10.1080/01691864.2021.1896381

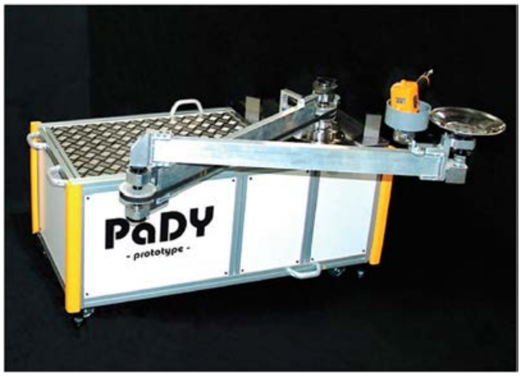


Figure 1. Co-worker robot, named PaDY, in an automobile assembly process. It has a planar arm with two degrees of freedom.

[18] by ABB, UR series by Universal Robots [19], and CR series by FANUC [20].

These collaborative robots have been designed to be safe based on the ISO standards and technical specifications. They stop their motion when a collision with a worker and/or its environment is detected, to guarantee the safety of the workers in the workspace. They can be utilized without being isolated from humans, thereby facilitating their use in human environments. Accordingly, many collaborative robots have been applied to different production processes

Collaborative robots, however, could not be used as they are, in a production system such as the automobile assembly line. In the case of PaDY, which assists a worker involved in an assembly process by delivering tools and parts necessary for each task of the process, the assembly work will be delayed if the collaborative robot stops unexpectedly. Avoiding the halting of the robot motion is thus a serious issue for the collaborative robots.

This paper proposes a reactive motion-planning method for a collaborative robot, PaDY, which is required to arrive at the goal position at a specified time in a workspace shared by multiple workers. The proposed reactive motion-planning method efficiently plans a collision-free robot trajectory and reduces the instances of halts to avoid collisions with workers and/or environments by predicting the motion of the workers.

This paper is organized as follows. Section 2 reviews the configuration-time space ($\mathcal{CT}$-space) [21] and its implementation as Time-augmented C-space (TAC-space) [22]. Section 3 proposes time-layered C-spaces (TLC-spaces) for efficient motion planning in the case of moving obstacles while considering motion constraints of the robot. Section 4 provides the experimental results obtained using PaDY. Section 5 concludes this paper.

2. TAC-space for motion planning with moving obstacles

2.1. Configuration-time space

Considerable research has been performed in the field of robot motion planning. In 1979, Tomás Lozano-Pérez et al. proposed path planning for a manipulator in the configuration space (C-space) to avoid static obstacles [23]. Fujimura et al. proposed a motion planning method in C-space for a robot with moving obstacles, assuming that the shapes of the robot and the moving obstacles are represented by convex polygons [24, 25]. The use of C-space is a well-known method for the motion planning of a robot with multiple degrees of freedom. The concept of the configuration-time space ($\mathcal{CT}$-space), or the path-time space, was proposed as a recursive trajectory-planning method for a manipulator with multiple degrees of freedom by Kamal Kant and Steven W. Zucker in 1986 [26]. Jean-Claude Latombe suggested the use of the $\mathcal{CT}$-space for robot motion planning in an environment with moving obstacles; additionally, they indicated the difficulties associated with the velocity/acceleration constraints on robot motion in 1991 [21]. In 1996, Tsubouchi et al. proposed a motion-planning method for a mobile robot with multiple obstacles moving at constant velocities in the $\mathcal{CT}$-space [27].

Sampling-based path-planning methods in the C-space, including probabilistic roadmap (PRM) [28] proposed in 1996 and rapidly exploring random tree (RRT) [29] proposed in 1998, have been widely used to implement a real-time motion planner for a manipulator. In 2006, J. van den Berg et al. proposed anytime path planning for mobile robots assuming obstacles moving at constant velocities (slanted cylindrical obstacles in the $\mathcal{CT}$-space) based on PRM [30]. In 2007, Sakahara et al. proposed a motion planning method that adopted the RRT algorithm in the $\mathcal{CT}$-space (spatiotemporal space), assuming obstacles moving at constant velocities [31]. In 2009, Tsai et al. proposed bi-directional RRTs in the $\mathcal{CT}$-space, and they searched for a path using RRT in the state-time space to avoid collision with predicted obstacles moving at constant speeds using Kalman filter [32].

In this paper, we propose a $\mathcal{CT}$-space based motion-planning method that efficiently plans the motion in a workspace with multiple workers (Figure 2).

2.2. Time-augmented C-space

The TAC-space was proposed based on predicted motions of multiple obstacles [22]. It is an implementation of the $\mathcal{CT}$-space [21] with multiple moving obstacles. In this subsection, we briefly review the TAC-space, which is a collection of the current and future C-spaces

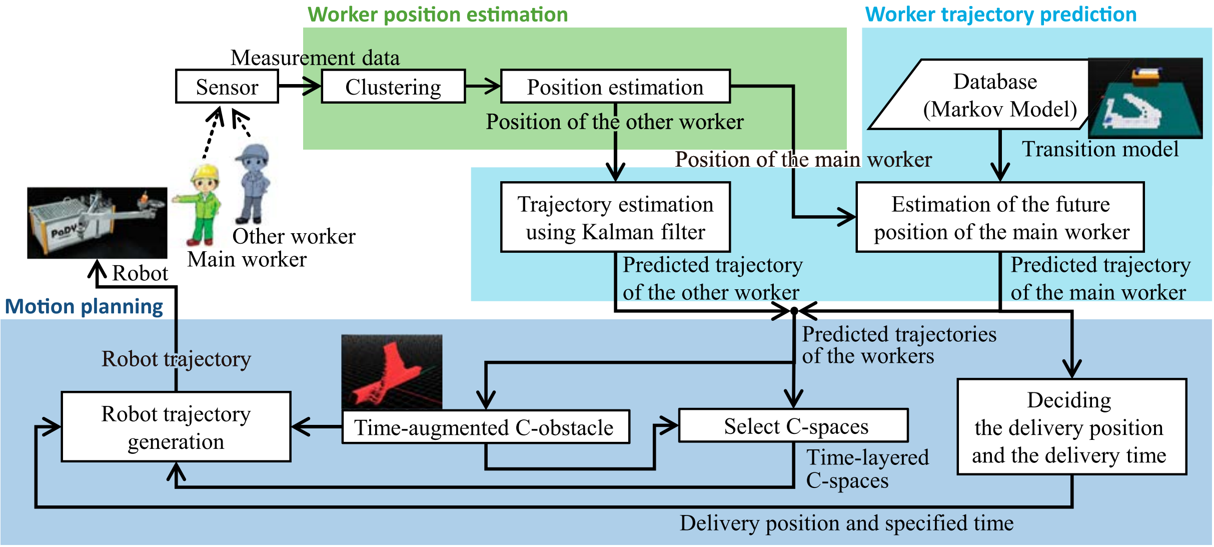


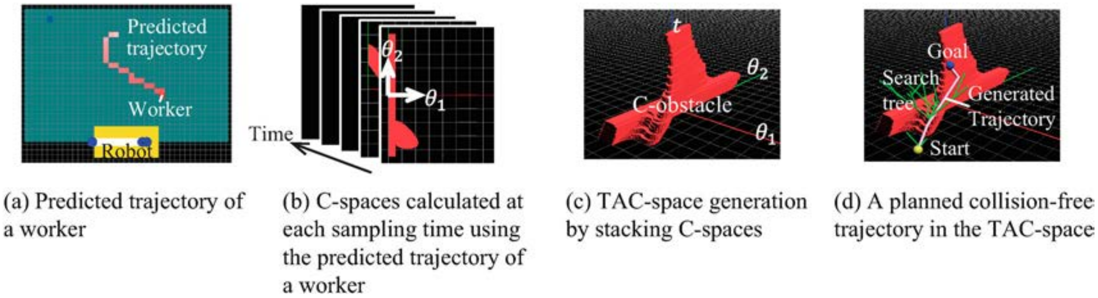
Figure 2. Proposed parts/tools delivery system for assembly tasks.

predicted at each sampling time. Calculation of the C-space with the locations of the predicted obstacles for each sampling time requires a significant computation time.

To reduce the computation time of the C-space and make the real-time implementation possible, we divide the workspace into cells via a two-dimensional grid. In our previous study [22], the workspace was divided into 0.1 m × 0.1 m square cells. The cell where the worker assisted by PaDY exists is predicted using the Markov model, and Kalman filter is used to predict the cells of the other workers in the workspace. The C-obstacle corresponding to each cell is calculated offline. The C-space at any time can be calculated by the union of the C-obstacles that correspond to the cells that are occupied by the obstacles. The TAC-space is generated by collecting the C-spaces with C-obstacles along the time axis.

An example of motion planning in the TAC-space is shown in Figure 3. Consider a case wherein only the PaDY-assisted worker (i.e. main worker) exists in the workspace, and also assume that the main worker and PaDY share the same workspace. First, the motion trajectory of the worker is predicted in the workspace using the Markov model, as shown in Figure 3(a). The corresponding C-space for each sampling time is obtained from the predicted position of the worker and the cells occupied by the worker, as shown in Figure 3(b). The TAC-space is obtained by stacking the obtained C-spaces along the time axis, as shown in Figure 3(c). An example of motion planning performed in the TAC-space by using RRT is shown in Figure 3(d). Because the predicted movement of the worker is reflected in the TAC-space, a collision-free robot motion can be planned in a dynamic environment.

Because the collision check of the planned motion in TAC-space is performed at every sampling time and the motion is re-planned when necessary, the effect of the motion prediction error is eliminated. The motion prediction error of the worker in the vicinity of the current location is sufficiently small for calculating a collision-free trajectory.



(a) Predicted trajectory of a worker

(b) C-spaces calculated at each sampling time using the predicted trajectory of a worker

(c) TAC-space generation by stacking C-spaces

(d) A planned collision-free trajectory in the TAC-space

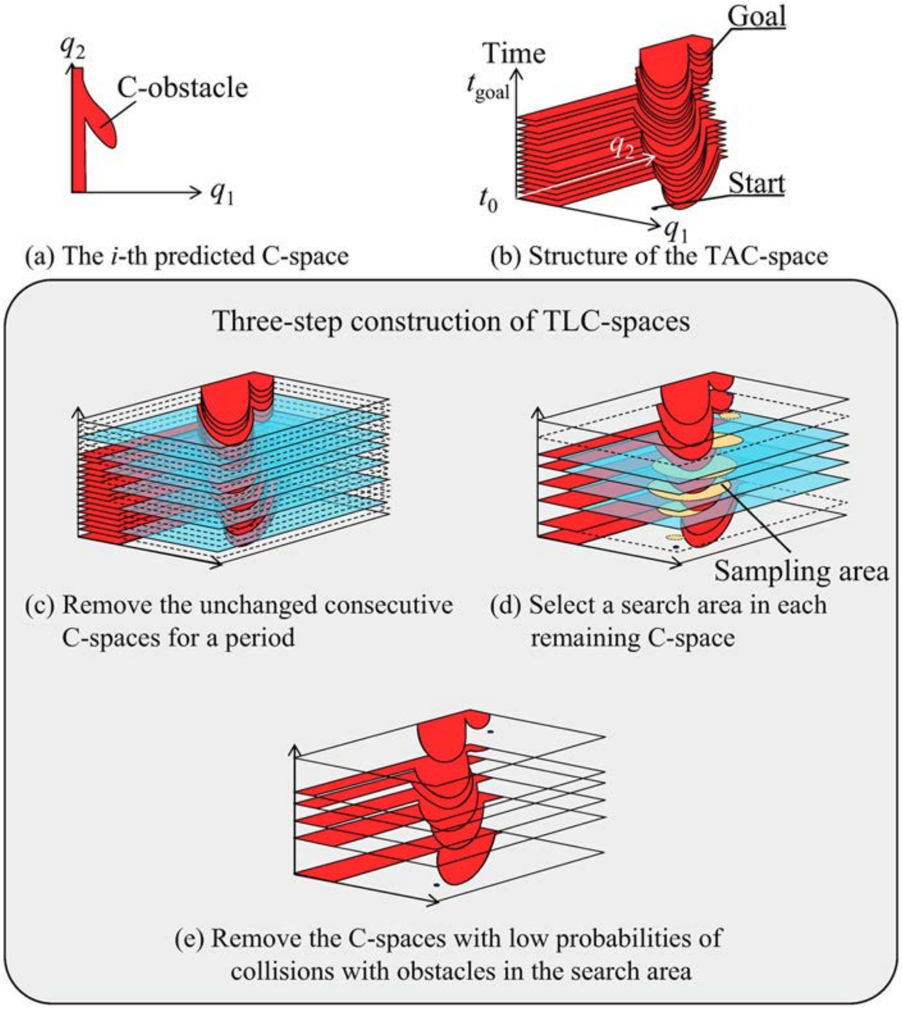
Figure 3. Example of motion planning in the TAC-space.

3. Time-layered C-spaces

When the current robot configuration and the goal configuration with a specified time are given, the current and predicted positions of the workers are updated. C-space at each sampling time until the specified time at the goal configuration is calculated as shown in Figure 4(a). The TAC-space is then obtained by stacking the current and predicted future C-spaces along the time axis, as shown in Figure 4(b). To make the trajectory search efficient in the TAC-space using sampling-based methods, such as RRT and PRM, we propose TLC-spaces in this section. TLC-spaces comprise the C-spaces of the TAC-space selected for trajectory planning. The C-spaces used for trajectory planning are selected in the following three steps.

The first step is to remove similar, consecutive C-spaces from the TAC-space, as shown in Figure 4(c). When all the workers do not move from their cells for a certain period, their C-spaces during that period will be the same. To reduce the search space in the TAC-space, we remove similar C-spaces that do not change for a certain period. The second step is to select a search area in each remaining C-space necessary for trajectory planning. The area reachable by the robot from the current configuration in each C-space is calculated using constraints on robot velocity/acceleration.

Although the TLC-space can be applied to a robot with n degrees of freedom by considering the workspace as a collection of square poles, we consider a planar manipulator with two degrees of freedom for the sake of simplicity. Table 1 shows a list of symbols which is using in math formulas described below. Let the ith C-space be the current or predicted C-space at time t_i. Additionally, let the jth C-space be the predicted C-space at time t_j ($t_i < t_j$). Consider a search area in the jth C-space when we plan a trajectory in the TLC-space from a configuration $\boldsymbol{q}_i \in \mathbb{R}^2$ in the ith C-space. Let $q_{i,m}$ denote the mth element of $\boldsymbol{q}_i \in \mathbb{R}^2$ and $\dot{q}_{i,m}$ denote the time derivative of $q_{i,m}$ at time t_i ($m = 1, 2$). Let $q_{j,m}$ denote the mth element of $\boldsymbol{q}_j \in \mathbb{R}^2$ and $\dot{q}_{j,m}$ denote the time derivative of $q_{j,m}$ at time t_j ($m = 1, 2$). Let the maximum acceleration of q_m



(a) The i-th predicted C-space

(b) Structure of the TAC-space

(c) Remove the unchanged consecutive C-spaces for a period

(d) Select a search area in each remaining C-space

(e) Remove the C-spaces with low probabilities of collisions with obstacles in the search area

Figure 4. Construction of TLC-spaces.

Table 1. List of description of symbols.

Symbol	Description
$\boldsymbol{q}$	robot configuration
t	time
q_m	mth element of robot configuration
$\dot{q}_m$	velocity of q_m
$\ddot{q}_m$	acceleration of q_m
$\dot{q}_m^{\max}$	maximum velocity of q_m
$\ddot{q}_m^{\max}$	maximum acceleration of q_m
$\boldsymbol{q}_i$	arbitrary configuration of the robot in time t_i
$q_{i,m}$	mth element of robot configuration $\boldsymbol{q}_i$
$\dot{q}_{i,m}$	velocity of $q_{i,m}$
$\ddot{q}_{i,m}$	acceleration of $q_{i,m}$
$\boldsymbol{q}_j$	arbitrary configuration of the robot in time t_j
$q_{j,m}$	mth element of robot configuration $\boldsymbol{q}_j$
$\dot{q}_{j,m}$	velocity of $q_{j,m}$
$\ddot{q}_{j,m}$	acceleration of $q_{j,m}$
$\boldsymbol{q}_{\text{goal}}$	robot goal configuration
t_{goal}	target goal time
$q_{\text{goal,m}}$	mth element of robot goal configuration $\boldsymbol{q}_{\text{goal}}$
$\boldsymbol{q}_{\text{start}}$	robot start configuration
$q_{\text{start,m}}$	mth element of robot start configuration $\boldsymbol{q}_{\text{start}}$
$\boldsymbol{\xi}(t)$	robot trajectory in the C-space
$\dot{\boldsymbol{\xi}}(t)$	velocity of robot trajectory in the C-space
$\ddot{\boldsymbol{\xi}}(t)$	acceleration of robot trajectory in the C-space

(the mth element of robot configuration $\boldsymbol{q}$ in its C-space) be $\ddot{q}_m^{\max}$ (a positive constant) and the maximum velocity of q_m be $\dot{q}_m^{\max}$ (a positive constant). Accordingly, we have the following:

$$|\dot{q}_m| \leq \dot{q}_m^{\max}, \tag{1}$$

$$|\ddot{q}_m| \leq \ddot{q}_m^{\max}. \tag{2}$$

The reachable area of configuration $\boldsymbol{q}_j$ in the jth C-space at time t_j satisfies the following inequality:

$$q_{j,m}^{\min} \leq q_{j,m} \leq q_{j,m}^{\max}, \tag{3}$$

where,

$$q_{j,m}^{\min} = \max(\check{q}_{j,m}^{\min}, \hat{q}_{j,m}^{\min}) \tag{4}$$

$$q_{j,m}^{\max} = \min(\check{q}_{j,m}^{\max}, \hat{q}_{j,m}^{\max}), \tag{5}$$

where $\check{q}_{j,m}^{\min}$ and $\check{q}_{j,m}^{\max}$ denote the lower and upper bounds of $q_{j,m}$ reachable from $q_{i,m}$, and $\hat{q}_{j,m}^{\min}$ and $\hat{q}_{j,m}^{\max}$ denote the lower and upper bounds of $q_{j,m}$ reachable to goal configuration q_{goal} at the specified time t_{goal}.

$\check{q}_{j,m}^{\min}$ and $\check{q}_{j,m}^{\max}$ are derived as follows:

$$\check{q}_{j,m}^{\min} = \begin{cases} q_{i,m} + \dot{q}_{i,m}(t_j - t_i) \\ \quad -\dfrac{1}{2}\ddot{q}_m^{\max}(t_j - t_i)^2 & (t_j < t_{\text{acc},i,m}^{-}) \\ q_{i,m} - \dot{q}_m^{\max}(t_j - t_{\text{acc},i,m}^{-}) \\ \quad +\dot{q}_{i,m}(t_{\text{acc},i,m}^{-} - t_i) \\ \quad -\dfrac{1}{2}\ddot{q}_m^{\max}(t_{\text{acc},i,m}^{-} - t_i)^2 & (\text{otherwise}), \end{cases} \tag{6}$$

$$\check{q}_{j,m}^{\max} = \begin{cases} q_{i,m} + \dot{q}_{i,m}(t_j - t_i) \\ \quad +\dfrac{1}{2}\ddot{q}_m^{\max}(t_j - t_i)^2 & (t_j < t_{\text{acc},i,m}^{+}) \\ q_{i,m} + \dot{q}_m^{\max}(t_j - t_{\text{acc},i,m}^{+}) \\ \quad +\dot{q}_{i,m}(t_{\text{acc},i,m}^{+} - t_i) \\ \quad +\dfrac{1}{2}\ddot{q}_m^{\max}(t_{\text{acc},i,m}^{+} - t_i)^2 & (\text{otherwise}), \end{cases} \tag{7}$$

where,

$$t_{\text{acc},i,m}^{-} = \frac{\dot{q}_m^{\max} + \dot{q}_{i,m}}{\ddot{q}_m^{\max}} + t_i, \tag{8}$$

$$t_{\text{acc},i,m}^{+} = \frac{\dot{q}_m^{\max} - \dot{q}_{i,m}}{\ddot{q}_m^{\max}} + t_i. \tag{9}$$

With the minimum and maximum accelerations, $\ddot{q}_m = -\ddot{q}_m^{\max}$ and $\ddot{q}_m = \ddot{q}_m^{\max}$, the velocity of robot configuration $\dot{q}_m$ reaches its minimum and maximum at times $t = t_{\text{acc},i,m}^{-}$ and $t = t_{\text{acc},i,m}^{+}$, as shown in Figure 5(a).

At the goal configuration, we have

$$q_m(t_{\text{goal}}) = q_{\text{goal},m}, \tag{10}$$

and the velocity and acceleration are zero, that is,

$$\dot{q}_m(t_{\text{goal}}) = 0, \tag{11}$$

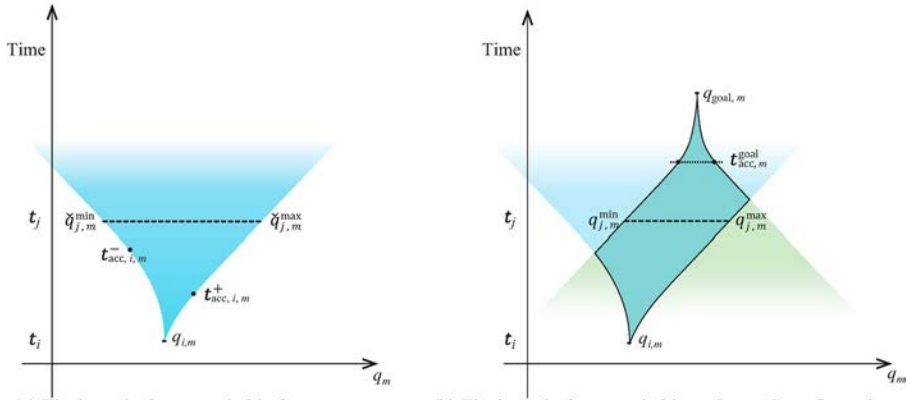
$$\ddot{q}_m(t_{\text{goal}}) = 0. \tag{12}$$

Under these conditions, $\hat{q}_{j,m}^{\min}$ and $\hat{q}_{j,m}^{\max}$ are derived as follows:

$$\hat{q}_{j,m}^{\min} = \begin{cases} q_{\text{goal},m} \\ \quad -\dfrac{1}{2}\ddot{q}_{i,m}(t_{\text{goal}} - t_j)^2 & (t_j > t_{\text{acc},m}^{\text{goal}}) \\ q_{\text{goal},m} - \dot{q}_{i,m}(t_{\text{goal}} - t_j) \\ \quad (t_{\text{acc},m}^{\text{goal}} - t_j) \\ \quad -\dfrac{1}{2}\ddot{q}_{i,m}(t_{\text{goal}} - t_{\text{acc},m}^{\text{goal}})^2 & (\text{otherwise}), \end{cases} \tag{13}$$

$$\hat{q}_{j,m}^{\max} = \begin{cases} q_{\text{goal},m} \\ \quad +\dfrac{1}{2}\ddot{q}_{i,m}(t_{\text{goal}} - t_j)^2 & (t_j > t_{\text{acc},m}^{\text{goal}}) \\ q_{\text{goal},m} + \dot{q}_{i,m}(t_{\text{goal}} - t_j) \\ \quad (t_{\text{acc},m}^{\text{goal}} - t_j) \\ \quad +\frac{1}{2}\ddot{q}_{i,m}(t_{\text{goal}} - t_{\text{acc},m}^{\text{goal}})^2 & (\text{otherwise}), \end{cases} \tag{14}$$

where $q_{\text{goal},m}$ denotes the mth element of $\boldsymbol{q}_{\text{goal}}$ and $t_{\text{acc},m}^{\text{goal}}$ denotes the latest time to start to reduce the velocity from its maximum or to increase the velocity to reach the goal with both the velocity and acceleration as zero, as shown



(a) The bound of $q_{j,m}$ reachable from $q_{i,m}$

(b) The bound of $q_{j,m}$ reachable to the goal configuration $q_{\mathrm{goal},m}$ and final search area for $q_{j,m}$

Figure 5. Reachable configuration from the *i*th C-spaces to the *j*th C-space ($j > i$) under velocity and acceleration constraints.

in Figure 5(b). The time $t^{\mathrm{goal}}_{\mathrm{acc},m}$ is calculated as follows:

$$t^{\mathrm{goal}}_{\mathrm{acc},m} = t_{\mathrm{goal}} - \frac{\dot{q}^{\max}_m}{\ddot{q}^{\max}_m}. \tag{15}$$

The search space is reduced by performing a search in the *j*th C-space only for configurations that satisfy inequality (3).

The third step is to remove the C-spaces with low probabilities of collisions with obstacles. To investigate the probability of collisions, random sampling scheme is applied only to the search area of each selected C-space, as previously mentioned. If the probability of collision in the search space of a C-space is low, then that C-space is not considered for trajectory planning. Conversely, if the number of samples in a C-obstacle area exceeds an empirically obtained threshold, the corresponding C-space is selected for trajectory planning. Note that we use random sampling scheme to perform collision checking when extending a trajectory during the proposed trajectory planning, as will be shown in Section 4. Through these three steps, from the TAC-space, we select the relevant C-spaces used for trajectory planning, following which we construct TLC-spaces.

4. Trajectory planning in TLC-spaces

A collision-free trajectory-planning method in the TLC-space is proposed in this section. The outline of the proposed method is shown in Figure 6. The positions of the workers and the robot configuration are updated, and the TAC-space is constructed for each sampling time. The collision of the current trajectory in the updated TAC-space is checked from the current time. If no collision is detected from the currently planned trajectory in each C-space to the goal, then the currently planned trajectory is used as it is.

If the collision of the current trajectory with C-obstacles in some C-space of the TAC-space is detected, then a TLC-space from the current configuration to the goal configuration is constructed. The search area in each C-space of the TLC-space is determined as described in the previous section. Random sampling scheme is then applied to C-spaces from the initial C-space to the final C-space of the TLC-space along the time axis. The trajectory is planned from the initial configuration in the initial C-space of the TLC-space to the goal configuration, as described in the following.

(1) In the *i*th C-space, randomly select a certain number of configurations from the configurations whose trajectory is connected with the initial configuration. Note that in the beginning, only the initial configuration (current configuration) exists.
(2) Interpolate between one of the selected configurations in the *i*th C-space and the goal configuration, and then generate a trajectory. This step is a kind of a goal-bias often used to plan motion planning using RRT. The trajectory planning process is terminated upon the generation of a feasible trajectory.
(3) If the interpolation between each of the selected configurations in the *i*th C-space and the goal configuration failed, then randomly sample a certain number of configurations inside the search area of the $(i + 1)$th C-space.
(4) Extend the search tree from the configurations in the *i*th C-space to all the randomly sampled configurations in the $(i + 1)$th C-space by interpolation.
(5) Repeat Steps 2–4 until a feasible trajectory is generated or the sampling number exceeds the specified maximum sampling number.

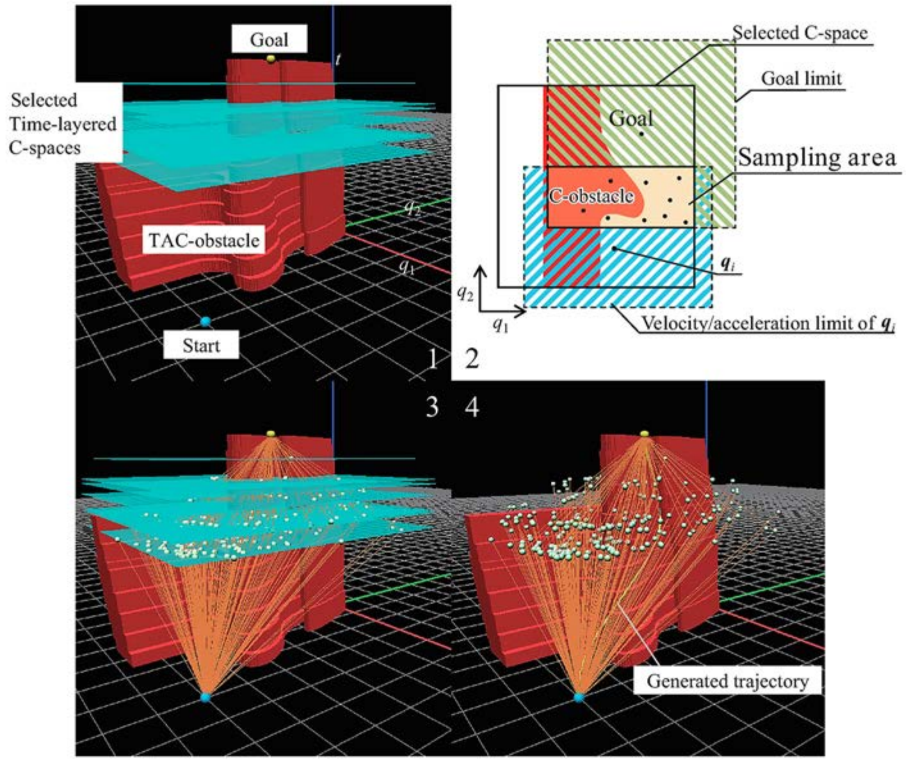


Figure 6. Outline of trajectory planning using the TLC-spaces. (1) Construct TLC-spaces from the TAC-space. (2) Determine the search area for trajectory extension with velocity/acceleration constraints. (3) Sample the path points randomly in each C-space of the TLC-spaces. (4) The trajectory search terminates upon the generation of a feasible trajectory by interpolating path points with a 5th-order polynomial.

The interpolation of the mth elements of the configurations $q_{i,m}$ and $q_{j,m}$ ($j = i+1$ or goal) is performed using the following fifth-order polynomial.

$$\begin{cases} \boldsymbol{\xi}(t) = \boldsymbol{a}_0 + \boldsymbol{a}_1 t + \boldsymbol{a}_2 t^2 + \boldsymbol{a}_3 t^3 + \boldsymbol{a}_4 t^4 + \boldsymbol{a}_5 t^5 \\ \dot{\boldsymbol{\xi}}(t) = \boldsymbol{a}_1 + 2\boldsymbol{a}_2 t + 3\boldsymbol{a}_3 t^2 + 4\boldsymbol{a}_4 t^3 + 5\boldsymbol{a}_5 t^4 \\ \ddot{\boldsymbol{\xi}}(t) = 2\boldsymbol{a}_2 + 6\boldsymbol{a}_3 t + 12\boldsymbol{a}_4 t^2 + 20\boldsymbol{a}_5 t^3, \end{cases} \tag{16}$$

where $\boldsymbol{a}_0, \ldots, \boldsymbol{a}_5$ denote coefficients derived from the initial and terminal conditions as follows:

$$\begin{cases} \xi_m(t_i) = q_{i,m} \\ \dot{\xi}_m(t_i) = \dot{q}_{i,m} \\ \ddot{\xi}_m(t_i) = \ddot{q}_{i,m} \\ \xi_m(t_j) = q_{j,m} \\ \dot{\xi}_m(t_j) = \dot{q}_{j,m} \\ \ddot{\xi}_m(t_j) = \ddot{q}_{j,m}. \end{cases} \tag{17}$$

Note that for the start and goal configuration, i.e. when $t_i = 0$ and $t_j = t_{\mathrm{goal}}$, the following terminal conditions are given:

$$\begin{cases} \xi_m(0) = q_{\mathrm{start},m} \\ \dot{\xi}_m(0) = 0 \\ \ddot{\xi}_m(0) = 0 \\ \xi_m(t_{\mathrm{goal}}) = q_{\mathrm{goal},m} \\ \dot{\xi}_m(t_{\mathrm{goal}}) = 0 \\ \ddot{\xi}_m(t_{\mathrm{goal}}) = 0. \end{cases} \tag{18}$$

Additionally, $\boldsymbol{a}_0, \ldots, \boldsymbol{a}_5$ are derived as follows:

$$\begin{cases} \boldsymbol{a}_0 = \boldsymbol{q}_i \\ \boldsymbol{a}_1 = \dot{\boldsymbol{q}}_i \\ \boldsymbol{a}_2 = \ddot{\boldsymbol{q}}_i/2 \\ \boldsymbol{a}_3 = \dfrac{\begin{array}{c} 20\boldsymbol{q}_j - 20\boldsymbol{q}_i - (8\dot{\boldsymbol{q}}_j + 12\dot{\boldsymbol{q}}_i)\Delta t_{i,j} \\ -(\ddot{\boldsymbol{q}}_j - 3\ddot{\boldsymbol{q}}_i)\Delta t_{i,j}^2 \end{array}}{2\Delta t_{i,j}^3} \\ \boldsymbol{a}_4 = \dfrac{\begin{array}{c} -30\boldsymbol{q}_j + 30\boldsymbol{q}_i + (14\dot{\boldsymbol{q}}_j + 16\dot{\boldsymbol{q}}_i)\Delta t_{i,j} \\ -(2\ddot{\boldsymbol{q}}_j - 3\ddot{\boldsymbol{q}}_i)\Delta t_{i,j}^2 \end{array}}{2\Delta t_{i,j}^4} \\ \boldsymbol{a}_5 = \dfrac{12\boldsymbol{q}_j - 12\boldsymbol{q}_i - (6\dot{\boldsymbol{q}}_j + 6\dot{\boldsymbol{q}}_i)\Delta t_{i,j} + (\ddot{\boldsymbol{q}}_j - \ddot{\boldsymbol{q}}_i)\Delta t_{i,j}^2}{2\Delta t_{i,j}^5}, \end{cases} \tag{19}$$

where $\Delta t_{i,j} = t_j - t_i$.

The terminal conditions except those of the goal configuration are not given a priori. To use the fifth-order polynomial for the interpolation, we must determine $\dot{q}_{j,m}$ and $\ddot{q}_{j,m}$. Let us consider how to determine $\dot{q}_{j,m}$ first. To simplify the implementation, we select $\dot{q}_{j,m}$ as follows:

$$\dot{q}_{j,m} = v_{\mathrm{start},m}(t_j, q_{j,m})T_{\mathrm{start}}(t_j) + v_{\mathrm{goal},m}(t_j, q_{j,m})T_{\mathrm{goal}}(t_j). \tag{20}$$

where,

$$v_{\mathrm{start},m}(t_j, q_{j,m}) = \frac{q_{j,m} - q_{\mathrm{start},m}}{t_j}, \tag{21}$$

$$T_{\mathrm{start}}(t_j) = 1 - \frac{t_j}{t_{\mathrm{goal}}} = \frac{t_{\mathrm{goal}} - t_j}{t_{\mathrm{goal}}}, \tag{22}$$

$$v_{\mathrm{goal},m}(t_j, q_{j,m}) = \frac{q_{\mathrm{goal},m} - q_{j,m}}{t_{\mathrm{goal}} - t_j}, \tag{23}$$

$$T_{\mathrm{goal}}(t_j) = 1 - \frac{t_{\mathrm{goal}} - t_j}{t_{\mathrm{goal}}} = \frac{t_j}{t_{\mathrm{goal}}}. \tag{24}$$

The velocity $\dot{q}_{j,m}$ is the weighted average of the average velocity from the start configuration to sampling point and the average velocity from the sampling point to goal configuration. Substituting Equations (21), (22), (23), and (24) into Equation (20), we have the following:

$$\begin{aligned}\dot{q}_{j,m} &= \frac{(q_{j,m} - q_{\mathrm{start},m})(t_{\mathrm{goal}} - t_j)}{q_{j,m}t_{\mathrm{goal}}} + \frac{(q_{\mathrm{goal},m} - q_{j,m})t_j}{(t_{\mathrm{goal}} - t_j)t_{\mathrm{goal}}} \\ &= \frac{1}{t_{\mathrm{goal}}}\left\{(q_{j,m} - q_{\mathrm{start},m})\frac{t_{\mathrm{goal}} - t_j}{t_j}\right. \\ &\quad \left. + (q_{\mathrm{goal},m} - q_{j,m})\frac{t_j}{t_{\mathrm{goal}} - t_j}\right\}.\end{aligned} \tag{25}$$

The acceleration $\ddot{q}_{j,m}$ is chosen to be zero, i.e. $\ddot{q}_{j,m} = 0$, to avoid excessive change in velocity at the selected configuration. Using these boundary conditions, we can smoothly interpolate two configurations.

5. Experiments

We used PaDY to perform experiments. Figure 7 shows the experimental environment, which consists of PaDY with a tool and parts, a white body of a vehicle, to which the parts are attached, and a laser-range-finder (LRF). The positions of the workers were updated every 30 ms using LRF. The LRF is placed to avoid occlusion for the given experimental tasks. In the actual environment, the possible occlusions could be avoided by using multiple LRFs [7, 10]. Please refer to [7] for more details about the experimental environments.

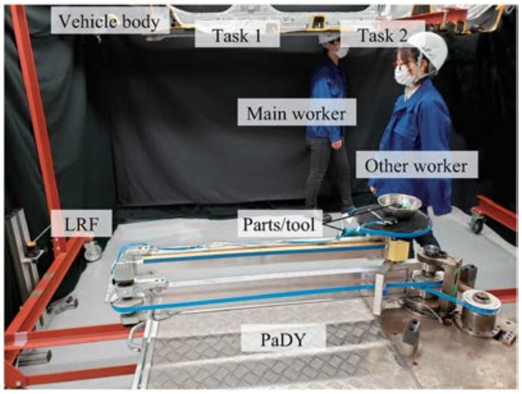


Figure 7. Experimental environment.

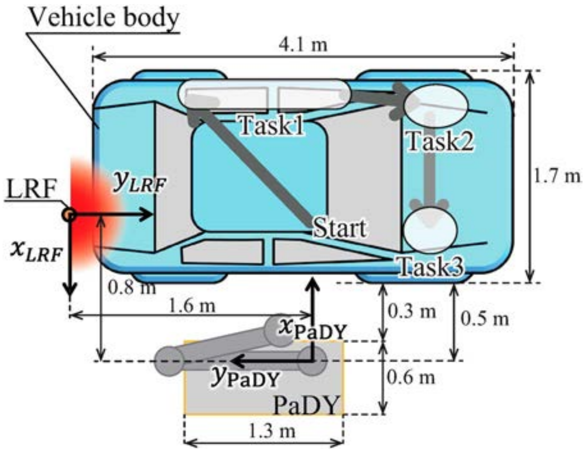


Figure 8. Execution point of three tasks.

Similar to an actual assembly process in an automobile production system, three tasks were performed in an experimental assembly process. Each task was performed for a different part of the vehicle body, as shown in Figure 8. In this process, the main worker was supposed to insert four rubber parts into holes of the body (Task 1), attach four plastic parts to the body (Task 2), and insert and tighten two bolts (Task 3).

PaDY began to move based on the predicted motion of the worker; accordingly, it arrived at the execution point of each task when the worker arrived there. Another worker intentionally cut across the path of PaDY in the experiment. The motion trajectory of the main worker was predicted by using a Markov model, and that of the other worker was predicted by using a Kalman filter, as explained in Section 2.2. PaDY arrived at each position in the workspace with 3.0 s after it started to move from its home position. After the delivery of parts/tools to the worker, it returned to the home position with 3.0 s to avoid possible collisions with the workers. The maximum angular velocity and maximum angular acceleration of each joint of PaDY were 120°/s and 180°/s^2, respectively.

The positions of the workers were updated every 30 ms. When constructing TLC-spaces, collisions with

C-obstacles were checked for each C-space, following which that particular C-space was selected for path planning if more than 10% of the 50 sampling points were located inside the C-obstacles.

The computation time required for planning trajectory using the proposed method was compared with that required for planning trajectory using the TAC-space. Parts assembly tasks were performed using PaDY in the experimental environment, and they were similar to the actual assembly work of an automobile factory. The trajectory planning was performed using a PC with an Intel Core i7-6700K processor, 32 GB RAM, and Windows 10 OS.

The proposed method using TLC-spaces described in Section 4 was compared with the motion planning method using TAC-space. RRT was used for the trajectory planning in the TAC-space. To evaluate the trajectory-planning methods, the following six scenarios were selected.

When the main worker moves from its initial position to Task 1, the following three scenarios are considered. In these scenarios, PaDY starts to move to the delivery position for Task 1, when the main worker is predicted to reach Task 1 position within three seconds.

Scenario 1: Only the main worker moves to Task 1 position in the vicinity of the possible path of PaDY from its initial position to the delivery position. The possibility of collision between the PaDY and the main worker is high.

Scenario 2: The other worker moves at an average velocity of 0.9 m/s in front of PaDY from left to right to intentionally in the vicinity of the possible path of PaDY. Although the possibility of collision between the PaDY and the workers seems high, C-obstacles in the PaDY's TAC space can be easily avoided as shown in Figure 9.

Scenario 3: The other worker moves at an average velocity of 0.9 m/s in front of PaDY from right to left to intentionally in the vicinity of the possible path of PaDY. The possibility of collision between the PaDY and the workers is high.

When the main worker moves from Task 1 to Task 2, the following three scenarios are considered. In these scenarios, PaDY starts to move to the delivery position for Task 2, when the main worker is predicted to reach Task 2 position within three seconds.

Scenario 4: Only the main worker moves to Task. In this case, there is no possibility of a collision between PaDY and the main worker.

Scenario 5: The other worker moves at an average velocity of 0.9 m/s in front of PaDY from left to right to intentionally in the vicinity of the possible path of PaDY. The possibility of collision between the PaDY and the workers is high.

Scenario 6: The other worker moves at an average velocity of 0.9 m/s in front of PaDY from right to left to intentionally in the vicinity of the possible path of PaDY. The possibility of collision between the PaDY and the workers is high.

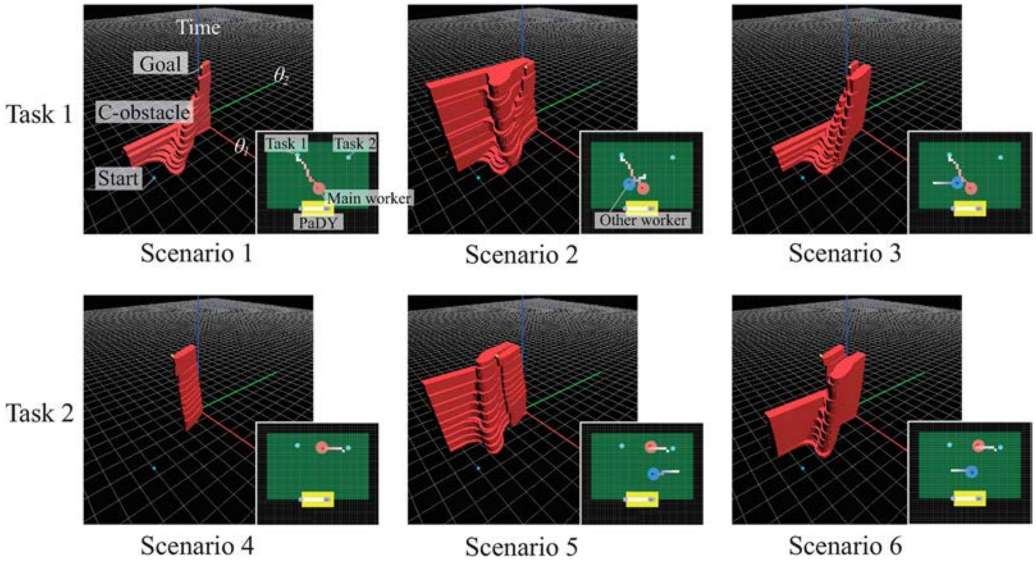
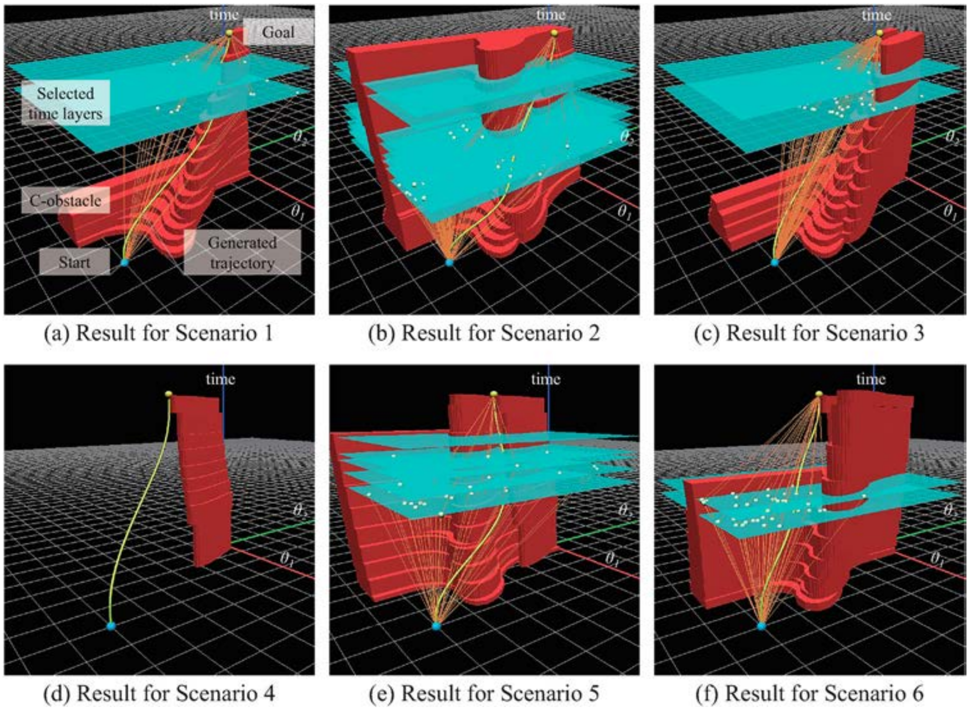


Figure 9. Six TAC-spaces used in the comparative experiments.

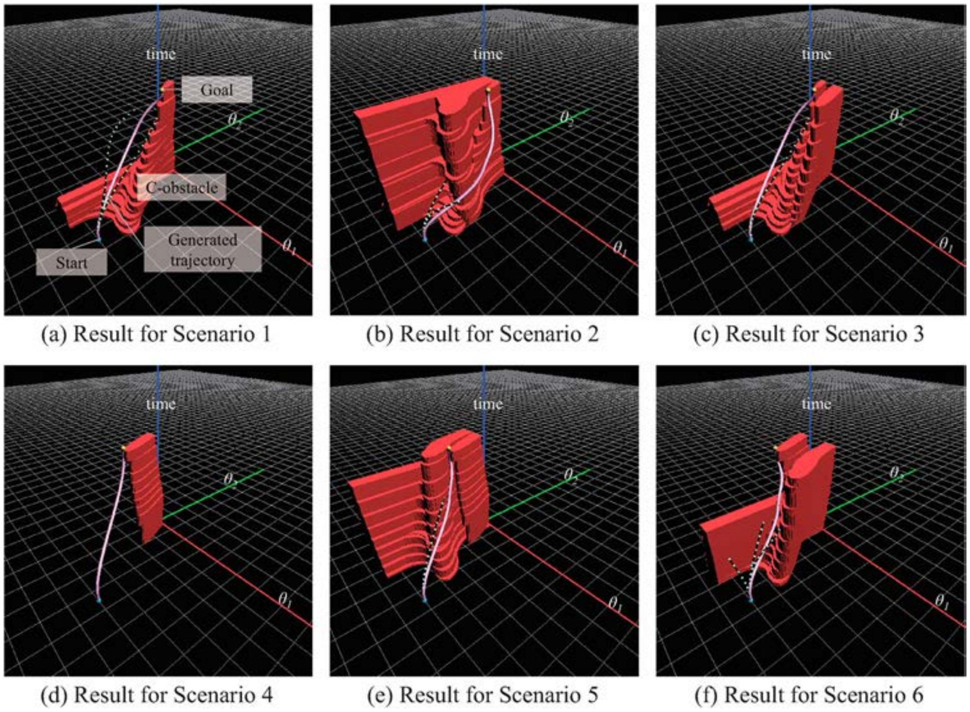
First, we asked the workers to emulate the tasks without the support of PaDY for the above-mentioned six scenarios. The motion of the workers, which was detected by LRF every 30 ms, were recorded for generating the TAC-space for simulation experiments. The TAC-space for each scenario used for the simulation experiments is shown in Figure 9.

An example of trajectories planned using the proposed method for each TAC-space is shown in Figure 10. For comparison, an example of trajectories planned using RRT for each TAC-space is shown in Figure 11. From these figures, the collision-free trajectories seem to have been successfully generated using both methods. This is because the computation time has not been considered



(a) Result for Scenario 1
(b) Result for Scenario 2
(c) Result for Scenario 3
(d) Result for Scenario 4
(e) Result for Scenario 5
(f) Result for Scenario 6

Figure 10. Examples of trajectories generated in the TLC-space.



(a) Result for Scenario 1
(b) Result for Scenario 2
(c) Result for Scenario 3
(d) Result for Scenario 4
(e) Result for Scenario 5
(f) Result for Scenario 6

Figure 11. Examples of trajectories generated in the TAC-space using RRT.

for the results shown in Figures 10 and 11. That is, the trajectory generation has been repeated until the collision-free trajectory is generated.

In our system, we consider it a success if a collision-free trajectory is generated within 20 ms. The success rate of the trajectory planning was calculated for 10,000 different simulations. Table 2 presents the success rate and computation time of the proposed method for each scenario and Table 3 presents the success rate and computation time using RRT for the TAC-space for each scenario.

Table 2. Success rate and computation time of the proposed method for each scenario.

Scenario		1	2	3	4	5	6	Avg.
Success rate (%)		100	100	100	100	100	100	100
Computation time (ms)	Average	1.11	1.10	1.17	0.41	0.65	0.72	0.86
	Variance	1.02	0.84	0.92	0.03	0.13	0.72	–

Table 3. Success rate and computation time of the method that searches for trajectories in the entire TAC-space for each scenario.

Scenario		1	2	3	4	5	6	Avg.
Success rate (%)		97.69	100	99.08	100	100	100	99.46
Computation time (ms)	Average	5.02	0.63	4.13	0.07	0.67	2.02	2.09
	Variance	23.78	0.35	16.86	0.00	0.47	4.01	–

One can see that the success rate of the method using the conventional RRT is not 100% for Scenarios 1 and 3, although the success rate of the proposed method is 100% for all of the scenarios including Scenarios 1 and 3. The results show that the proposed method can generate collision-free trajectories more reliably.

Let us compare the computation time. The computation time of the proposed method for Scenarios 1, 3, 5, and 6 are shorter than those of using RRT for TAC-space, but the computation time for Scenarios 2 and 4 are longer than those of using RRT for TAC-space.

In Scenario 2, the other worker moved from left to right in the workspace when the main worker was moving from the initial position to Task 1 execution point. In this case, the possible interference of the other worker with the path of PaDY was small, and thus planning the trajectory was easy. In Scenario 4, there was no possibility

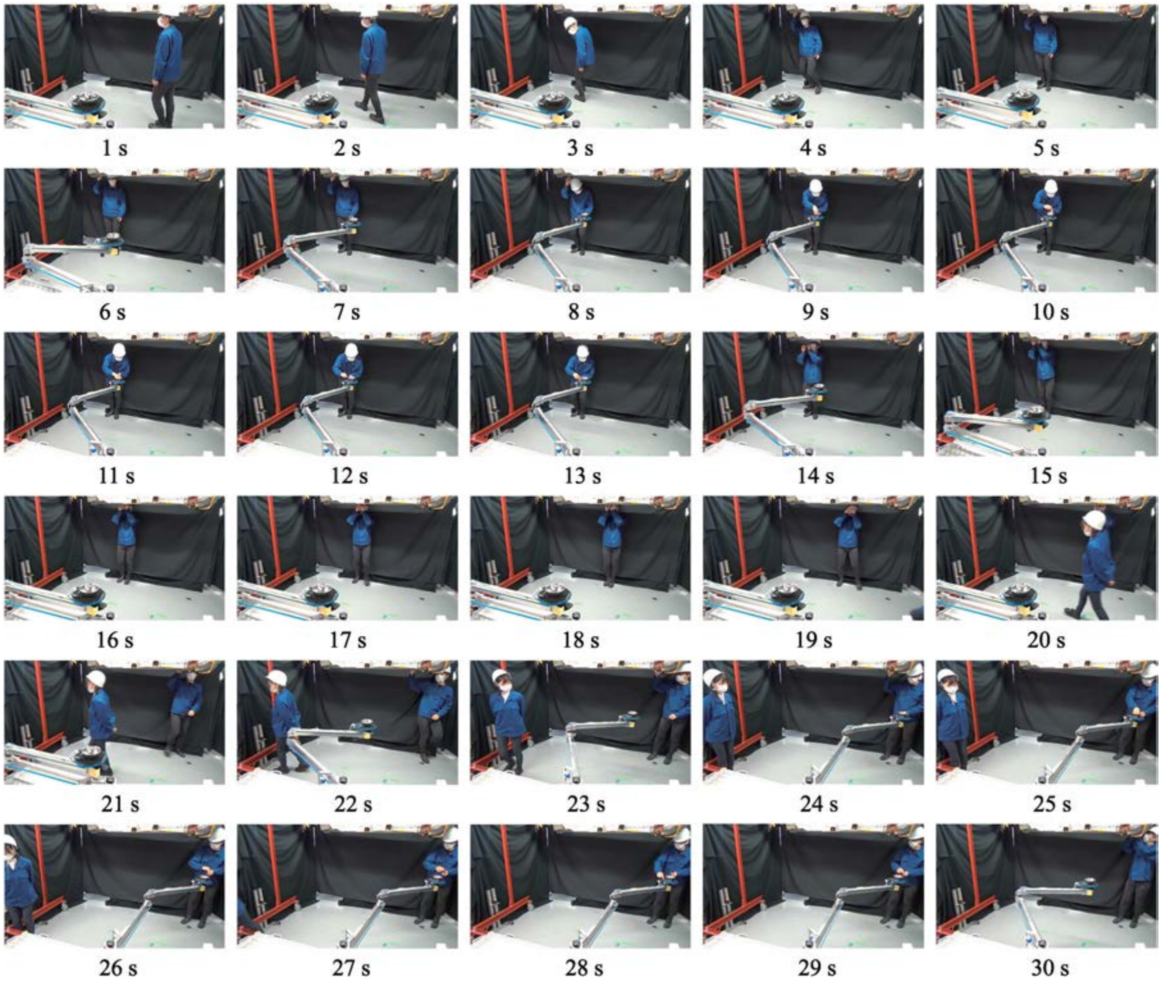
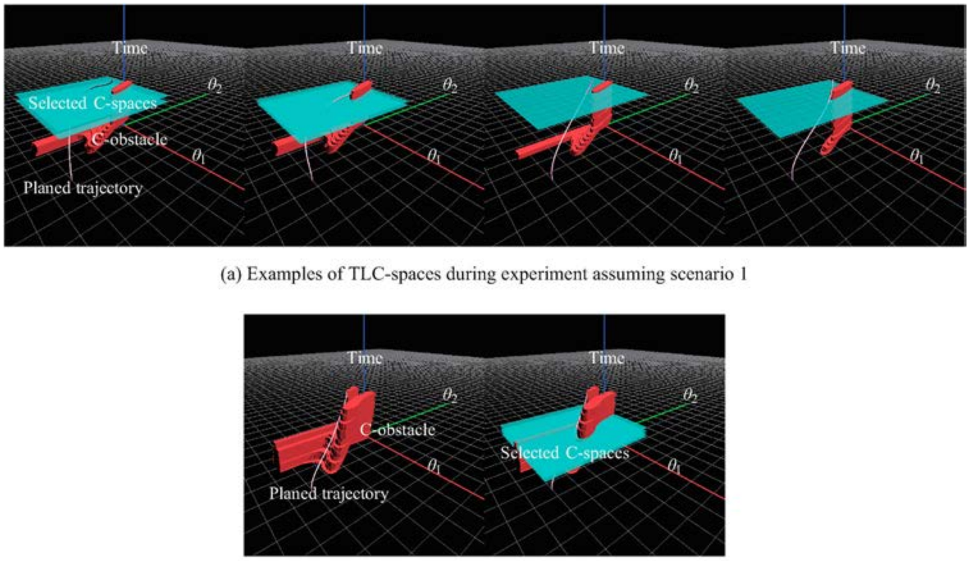


Figure 12. Human–robot collaborative work experiment.



(a) Examples of TLC-spaces during experiment assuming scenario 1

(b) TAC-space and TLC-spaces during experiment assuming scenario 6

Figure 13. Generated TLC-spaces and calculated trajectory for PaDY during the experiment shown in Figure 12.

of collisions between the worker and PaDY because Task 2 was performed far from the trajectory of PaDY. In these Scenarios, the computation time required for constructing the TLC-space made the computation time longer than that of the case using RRT for the TAC-space.

As previously mentioned, the use of the TLC-space is more effective in a complicated situation wherein PaDY might collide with workers. The variance of the computation time for each scenario using the TLC-spaces is considerably smaller than that using the TAC-space. This shows that the proposed method is more reliable and can effectively plan the trajectory even if the environment is crowded for motion planning.

The proposed method is also evaluated in the real environment described at the beginning of this section. Figure 12 shows an example of the experimental results for Scenarios 1 and 6. These scenarios were selected because they were the most difficult ones for motion planning, as determined from previous simulation experiments. In the experiment shown in Figure 12, the worker entered the workspace at around the 20th second and interrupted the path of PaDY from the 20th to 22nd seconds. However, PaDY successfully delivered parts and a tool to the main worker without colliding with the other worker.

Figure 13 shows the generated TLC-spaces and the planned trajectories during the experiment. The TAC-space, which is generated at each sampling time, is used to predict whether a collision would occur between the workers and robot. Only when a collision is predicted in the TAC-space, TLC-spaces are generated to re-plan the robot trajectory. Some of the results of the trajectory planning in Scenarios 1 and 6 are shown in Figure 13(a,b), respectively. The trajectory was re-planned eight times at around the 2nd or 3rd second because the main worker was moving in the workspace and changed his position, as shown in Figure 13(a).

During Scenario 6, after the trajectory was planned at around the 20th second, the trajectory was not further planned. This is because the main worker was moving from Task 1 to Task 2, both of which were far away from PaDY, and also the other worker left the path of PaDY at around the 23rd second. The predicted motion of the other worker at around the 20th second was sufficiently precise, and thus the planned trajectory was used without re-planning.

6. Conclusions

A reactive motion planning using TLC-spaces was proposed. TLC-spaces are the collection of only those C-spaces in the TAC-space that are relevant to the motion planning with moving obstacles (workers). The proposed trajectory-planning method, which searches the trajectory in the TLC-space using a random sampling scheme, reduced the computation time required for the trajectory search by using the entire TAC-space. Although the proposed concept was applied to PaDY, which has two degrees of freedom, the concept of reducing the search space can be easily extended to a collaborative robot with more than two degrees of freedom.

The simulation experiments showed that the average computation time for all the scenarios using the TLC-space was shorter than the computation time required using the TAC-space. The average computation time required for trajectory planning using TLC-spaces for each scenario was less than 2 ms. The success rates of motion planning using the TLC-spaces for the six scenarios were 100%, although those using the TAC-space for Scenarios 1 and 3 were less than 100%. The experimental results obtained using PaDY showed that the proposed method is applicable to a collaborative robot in an environment with two workers.

The PaDY used in this study had only two degrees of freedom and was not an industrial robot as per the ISO standard. However, if a PaDY has more than two degrees of freedom, we must comply with the ISO safety standards. To comply with the ISO safety standards, the maximum robot velocity should be selected according to its distance from workers, although the maximum velocity of each joint was assumed to be constant in this study. This is left for the future work.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This work was supported by Japan Society for the Promotion of Science – JSPS KAKENHI Grant-in-Aid for JSPS fellows Number JP110440.

Notes on contributors

Hisaka Wada received the BS, MS, and PhD degrees in engineering from Tohoku University, Sendai, Japan, in 2014, 2016, and 2020 respectively. Her research interests include human–robot interactions, co-worker robot, and robot motion planning.

Jun Kinugawa received a PhD degree in engineering from Tohoku University, Sendai, Japan, in 2011. He is currently an Assistant Professor with the Department of Robotics, Tohoku University. His research interests are human–robot interactive systems include co-worker robots, robotic hands, and power assistive systems.

Kazuhiro Kosuge received the BS, MS, and PhD degrees in Control Engineering from the Tokyo Institute of Technology in 1978, 1980, and 1988 respectively. He is a Distinguished Professor of Tohoku University in the Department of Robotics. From 1980 through 1982, he was a Research Staff in the Production Engineering Department, Nippon Denso Co., Ltd. (DENSO Co., Ltd.). From 1982 through 1990, he was a Research Associate in the Department of Control Engineering at Tokyo Institute of Technology, and from 1989 to 1990, he was a visiting research scientist at, Department of Mechanical Engineering, Massachusetts Institute of Technology. From 1990 to 1995, he was an Associate Professor at Nagoya University. From 1995, he has been at Tohoku University. He served as Vice President for Technical Activities, IEEE in 2020, Director & Delegate, Division X, Board of Directors, IEEE (2015–2016), President of IEEE Robotics and Automation Society (2010–2011) etc. He served for several domestic organizations in Japan. He served as Science Advisor, Research Promotion Bureau, Ministry of Education, Culture, Sports, Science and Technology, Japan (2010–2014), Senior Program Officer, Japan Society of Promotion of Science (2007–2010), Selected Fellow, Center for Research and Development Strategy, Japan Science and Technology Agency (2005–2012). He is Technical Committee Co-Chair on Robotics Research for Practicality, 2018-Present, and was Editor in Chief, Advanced Robotics (2002–2007). He is IEEE Fellow, SICE Fellow, JSME Fellow, RSJ Fellow and JSAE Fellow.

References

[1] Troccaz J, Lavallee S, Hellion E. A passive arm with dynamic constraints: a solution to safety problems in medical robotics. Proceedings of IEEE Systems Man and Cybernetics Conference-SMC; Vol. 3. IEEE; 1993.

[2] Edward Colgate J, Peshkin MA. Cobots. US patent no. 5,952,796. 1999 Sep 14.

[3] Edward Colgate J, Peshkin MA, Wannasuphoprasit W. Cobots: robots for collaboration with human operators. Proceedings of ASME International Mechanical Engineering Congress and Exposition; 1996. p. 433–439.

[4] Akella P, Peshkin M, Colgate ED, et al. Cobots for the automobile assembly line. Proceedings of IEEE International Conference on Robotics and Automation (ICRA 1999); Vol. 1; 1999. p. 728–733.

[5] Yamada Y, Konosu H, Morizono T, et al. Proposal of skill-assist: a system of assisting human workers by reflecting their skills in positioning tasks. 1999 Proceedings of IEEE International Conference on Systems, Man, and Cybernetics (SMC'99); Vol. 4; 1999. p. 11–16.

[6] ARAKI. Skill assist device (in Japanese). [Cited 2020 Aug 10]. Available from: https://www.araki-mfg.co.jp/products/02/

[7] Kinugawa J, Kawaai Y, Sugahara Y, et al. PaDY: human-friendly/cooperative working support robot for production site. 2010 Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS); 2010. p. 5472–5479.

[8] Kosuge K, Sugahara Y, Kinugawa J, et al. Working support robot system. US patent no. 8,682,482. 2014 Mar 25.

[9] Kinugawa J, Sugahara Y, Kosuge K. Co-worker robot–Pa DY. Acta Polytech Hung. 2016;13(1):209–221.

[10] Tanaka Y, Kinugawa J, Sugahara Y, et al. Motion planning with worker's trajectory prediction for assembly task partner robot. 2012 Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS); 2012. p. 1525–1532.

[11] Kinugawa J, Kanazawa A, Arai S, et al. Adaptive task scheduling for an assembly task co-worker robot based on incremental learning of a human's motion pattern. IEEE Robot Autom Lett. 2017;2(2):856–863.

[12] Kanazawa A, Kinugawa J, Kosuge K. Adaptive motion planning for a collaborative robot based on prediction uncertainty to enhance human safety and work efficiency. IEEE Trans Robot. 2019;35(4):817–832.

[13] ISO 10218-1:2011. Robot for industrial environments – safety requirements – part 1: robot. Geneva: International Organization for Standardization; 2011.

[14] ISO 10218-2:2011. Robot for industrial environments – safety requirements – part 2: robot systems and integration. Geneva: International Organization for Standardization; 2011.

[15] ISO/TS 15066. Robots and robotic devices – collaborative robots. International Organization for Standardization; 2016.

[16] KUKA. LBR iiwa. [cited 2018 Sep 7]. Available from: https://www.kuka.co/en-de/products/robotics-systems/industrial-robots/lbr-iiwa

[17] KUKA. LBR iisy. [cited 2018 Sep 7]. Available from: https://www.kuka.co/en-de/technologies/human-robot-collaboration/lbr-iisy

[18] ABB. YuMi – IRB 14000. [cited 2018 Sep 7]. Available from: https://new.abb.com/products/robotics/industrial-robots/yumi

[19] Universal Robots. Collaborative robots from universal robots. [cited 2018 Sep 7]. Available from: https://www.universal-robots.com/products

[20] FANUC. Collaborative industrial robot FANUC CR-35iA. [cited 2019 Nov 26]. Available from: http://www.fanuc.eu/pt/en/robots/robot-filter-page/collaborative-robots

[21] Latombe JC. Robot motion planning. Kluwer Academic; 1991. p. 22–23.

[22] Wada H, Kanazawa A, Konada K, et al. Dynamic collision avoidance method for co-worker robot using time augmented configuration-space. 2016 Proceedings of IEEE International Conference on Mechatronics and Automation (ICMA); 2016. p. 2564–2569.

[23] Lozano-Pérez T, Wesley MA. An algorithm for planning collision-free paths among polyhedral obstacles. Commun ACM. 1979;22(10):560–570.

[24] Fujimura K, Samet H. Accessibility: a new approach to path planning among moving obstacles. Proceedings CVPR '88: The Computer Society Conference on Computer Vision and Pattern Recognition; 1988. p. 803–807.

[25] Fujimura K, Samet H. Motion planning in a dynamic domain. Proceedings of IEEE International Conference on Robotics and Automation (ICRA); Vol. 1; 1990. p. 324–330.

[26] Kant K, Zucker SW. Toward efficient trajectory planning: the path-velocity decomposition. Int J Rob Res. 1986;5(3):72–89.

[27] Tsubouchi T, Naniwa T, Arimoto S. Planning and navigation by a mobile robot in the presence of multiple moving obstacles and their velocities. J Robot Mechatron. 1996;8(1):58–66.

[28] Kavraki LE, Svestka P, Latombe J-C, et al. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. IEEE Trans Rob Autom. 1996;12(4):566–580.

[29] LaValle SM. Rapidly-exploring random trees: a new tool for path planning. Computer Science Dept. Oct. 1998;98(11).

[30] van den Berg J, Ferguson D, Kuffner J. Anytime path planning and replanning in dynamic environments. 2006 Proceedings of IEEE International Conference on Robotics and Automation (ICRA); 2006. p. 2366–2371.

[31] Sakahara H, Yasuhiro M, Miyazaki F. Real time motion planning in dynamic environment containing moving obstacles using spatiotemporal RRT. Trans Soc Instrum Control Eng. 2007;43(4):277–284.

[32] Tsai YC, Huang HP. Motion planning of a dual-arm mobile robot in the configuration-time space. 2009 Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS); 2009. p. 2458–2463.