The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA

Compact Design of Work Cell with Robot Arm and Positioning Table under a Task Completion Time Constraint

Lounell B. Gueta, Ryosuke Chiba, Tamio Arai, Tsuyoshi Ueyama and Jun Ota

Abstract—**A work cell is generally designed to achieve a high throughput and its size is typically viewed as contingent to component sizes. In this paper, we aim to design a compact work cell (spatial requirement) and to minimize its task completion time (temporal requirement) to a value set as a constraint. By doing so, a work cell occupies a minimal space and achieves its desired throughput. The work cell size is evaluated based on the size and the swept volume of components. This evaluation is important since a robot arm can have a very large swept volume depending on a given task. To satisfy the spatial and temporal requirements, we propose the integration of the base placement optimization, goal rearrangement, and motion coordination between the robot arm and the positioning table. Furthermore, we introduce two motion coordination schemes based on the spatial and temporal requirements. We showed the effectiveness of the proposed method through simulations.**

I. Introduction

ROBOT arm with positioning table is an important system applied in manufacturing work cells such as in inspection and welding. It is widely-used due to its flexibility, reliability, and efficiency in the use of robot arm workspace [1]. Since a work cell is the basic manufacturing unit, this system has to be compact and be able to execute task in minimal time.

In previous studies, the facility layout problem, a nondeterministic polynomial time-complete problem, deals with the placement of several machines. A comprehensive survey of this problem is provided in [2]. In [3] and [4], a layout is designed to minimize the traveled path of a robot arm by determining the relative position and orientation of machines in a work cell. To save space, machines are represented as squares instead of *super*-shapes like circles enclosing machines [3]. Such representation allows the machines to be compactly located in a floor area. Since exact or linear programming methods are computationally inefficient for large number of machines, heuristics are used, e.g., machines with high frequency of interaction are placed near to each other [3]. Commonly, an area is divided into blocks and the path traveled by the robot arm is approximated as a distance between blocks. These distances can be easily calculated since a machine once it is assigned a placement in the layout can be treated as a static goal of the robot arm. In this paper, however, goals can be repositioned by a positioning table making the problem complicated. Some studies provide solutions for component arrangement, motion programming, and layout optimization in the form of interactive software [5], which may require some user skills to generate a good layout. In the above-mentioned studies, the layout is focused on two-dimensional (2D) floor area. A related study dealing with 3D layout of objects is evaluated in [6]. In that study, components are packed into a container with the objective of maximizing packing density. In this paper, however, the work cell components are moving (e. g., the robot arm and the positioning table simultaneously move to execute task).

A problem similar to the facility layout design is the robot arm base placement optimization. Several methods are proposed such as employing a random method for the base placement optimization with a probabilistic roadmap method for the motion planning [7], optimization of some kinematic criteria [8], [9], and the task completion time minimization [10]. In [10], it is shown that in order to reduce the task completion time, a robot arm must be placed afar from its goals to achieve few joint motions; this, however, results into a large occupied floor area.

Several studies focused on the motion planning of robot arms. A comprehensive literature for the motion planning can be found in [11]. In multiple-goal tasks (e.g., inspection and welding), motion planning has been dealt with in combination with goal rearrangement [12]-[15]. In those studies, the emphasis is on minimizing the task completion time.

In this paper, we aim to design a compact work cell consisting of a 6-DOF robot arm and a 1-DOF positioning table, as shown in Figure 1. This study is unique from previous studies in two aspects: 1) the evaluation of a compact work cell and 2) the incorporation of a task completion time constraint in the optimization. We evaluate the compactness of a work cell on the basis of the size and the swept volume of work cell components. This evaluation has two merits: (a) it extends the facility layout problem into 3 dimensions which is practical for real settings that consider both the floor area and the height of a work cell, and (b) it takes into account the motion of the work cell components, which is important since the swept volume can be very dependent on a given task. For instance, a robot arm has a small footprint but can occupy a substantial space owing to its very large workspace. Furthermore, only a portion of this workspace is utilized in a specific task. Minimizing the swept

L. B. Gueta, and J. Ota are with Research into Artifacts, Center for Engineering, at the University of Tokyo, Kashiwanoha 5-1-5, Kashiwa, Chiba, Japan, 277-8568. (emails:{gueta, ota}@robot.t.u-tokyo.ac.jp).

T. Arai is with Department of Precision Engineering, School of Engineering, at the University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo, Japan, 113-8656. (email: arai-tamio@robot.t.u-tokyo.ac.jp).

R. Chiba is with Faculty of System Design, Tokyo Metropolitan University, 6-6, Asahigaoka, Hino-shi, Tokyo, Japan, 191-0065. (email: rchiba@sd.tmu.ac.jp).

T. Ueyama is with DENSO WAVE INCORPORATED, 1, Yoshiike, Kusaki, Agui-cho, Chita-gun, Aichi, Japan, 470-2298.

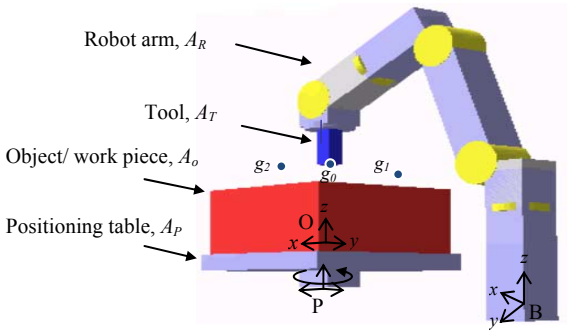

Fig. 1 A system consisting of a robot arm and a positioning table. *P* is the reference coordinate frame of the base of the table, *O* to that of an object or work piece, and *B* to that of the robot arm base.

volume can, therefore, be crucial in designing a work cell.

The incorporation of the task completion time constraint in the optimization is important to achieve a high-throughput target, e.g. number of volume of products in an hour. In this paper, the throughput target is determined by setting a desired task completion time, $t_{desired}$. Conceptually, if $t_{desired}$ is achieved in every work cell, the movement of products from one work cell to another can be planned to avoid bottlenecks in assembly line. In previous studies [12]-[15], the task completion time is minimized as the performance index; in this study, it is minimized and should satisfy the task completion time constraint.

In order to design a compact work cell and satisfy the task completion time constraint, we propose the integration of the base placement optimization, goal rearrangement, and motion coordination between the robot arm and the positioning table. We introduce two motion coordination schemes based on the the task completion time and the work cell size. No study has yet proposed a motion coordination scheme for robot arm and positioning table on the basis of the work cell size.

In Section II, the problem in this paper is described. Section III is an overview of the proposed method and Section IV discusses the details of this method. The simulations, results and discussion are provided in Section V and a conclusion is provided in Section VI.

II. Problem Description and Formulation

Let us consider a work cell with components consisting of a 6-DOF robot arm A_R, a tool A_T, a 1-DOF positioning table A_P, and an object A_O (Figure 1).

A. Input Parameters and Assumptions

The task is defined by several goals located on A_o. The goals are assumed to be given and are denoted as $g_i = (x_i, y_i, z_i, \alpha_i, \beta_i, \gamma_i)$ $i \in (1 \ldots n)$, where g_i is the i^{th} goal in a goal order, n is the number of goals, the position (x_i, y_i, z_i) is referred to the coordinate system of A_O and the tool orientation $(\alpha_i, \beta_i, \gamma_i)$, correspond to yaw, pitch, and roll, respectively. The home position (i. e., the initial and final configuration of A_R and A_P before and after task execution) is denoted as g_0.

We assume that the sizes and shapes of the work cell components are known. Moreover, we assume that A_P can handle objects of various sizes and weights without affecting its speed.

B. Objective Function and Constraints

The configuration of A_R and A_P at goal g_i is denoted as $q_i = \{\theta_i{}^0 \ldots \theta_i{}^6\}$ where $\theta_i{}^0$ is the rotation angle of A_P and $\{\theta_i{}^1 \ldots \theta_i{}^6\}$ are the joint angles of A_R. The task is executed as follows. First, A_R and A_P are positioned at configuration q_0; or conversely A_R is at goal g_0. Then, A_R moves to a goal to achieve the given goal pose; simultaneously, A_P rotates and positions A_O. Afterwards, A_R and A_P moves back to q_0. Based from this task description, the task completion time in moving from and returning to g_0 can be calculated as:

$$t_{actual} = \sum_{i=1}^{n} c(g_{i-1}, g_i) + c(g_n, g_0) \quad . \tag{1}$$

where $c(\cdot)$ is the motion time of A_R and A_P in moving from goals g_{i-1} to g_i. With respect to (1), we emphasize on the following points. First, t_{actual} is the actual task completion time of A_R and A_P in performing task, as opposed to $t_{desired}$ which is a constraint in the optimization. Second and last, we assume a point-to-point motion of A_R and A_P from one goal to another and the path between two goals is not given in advance. This assumption is applicable to tasks that require a robot arm to stop at every goal.

The total volume swept by A_R in performing an entire task, can be expressed as:

$$V_R = \bigcup vol_R(t) \quad , \quad t \in [0, t_{actual}] \tag{2}$$

where $vol_R(t)$ is the volume occupied by A_R at time t.

Similarly, the swept volume of A_P, A_O, and A_T can be expressed as V_P, V_O, and V_T, respectively. To design a compact work cell, the total volume of work cell must be minimized and can be calculated as:

$$V = \min(V_R \cup V_P \cup V_O \cup V_T) \tag{3}$$

In (3), the space gaps between the work cell components are not explicitly taken into account; in this paper, these gaps are practically considered in the representation of swept volume.

C. Constraints

The following constraints must be satisfied:

1) Task completion time constraint

$$t_{actual} \leq t_{desired} \tag{4}$$

where $t_{desired}$ is a user-specified value that defines the desired task completion time of the work cell.

2) Collision constraint

$$vol_R(t) \cap vol(t)_P \cap vol_T(t) \cap vol_O(t) = 0 \quad , \quad t \in [0, t_{actual}] \tag{5}$$

The above constraint ensures that any potential collisions among the work cell components are prevented.

3) Joint limit constraint

$$\theta^{j,\min} \leq \theta_i^j \leq \theta^{j,\max} \quad , \; i = 1 \ldots n, j = 0 \ldots 6, \quad t \in [0, t_{actual}] \tag{6}$$

where $\theta^{j,min}$ and $\theta^{j,max}$ are the minimum and maximum limits of the robot arm and table joints.

4) Calculation or design time constraint

The calculation time for designing a compact work cell must be within a user-defined design time limit. This constraint is imposed for practicality since an optimization can require extensive computational resources (e.g., several hours or days). For the purpose of this study, we set the design time limit to 30 minutes.

D. Design Variables

The design variables are as follows:

(a) base placement, $({}^{P}x_B, {}^{P}y_B, {}^{P}z_B)$

(b) goal order, π

(c) the configuration q_i, i= 1…n

The parameters $({}^{P}x_B, {}^{P}y_B, {}^{P}z_B)$ correspond to the distance between the robot arm base B and the positioning table P. The orientation of the base of A_R relative to A_P is not optimized in this study since it may not be practical in some applications (e. g., a robot arm base has to be mounted on a wall just to satisfy the derived base orientation.).

The goal order π is derived which a permutation of possible order of goals. The configurations of A_R and A_P are derived by finding q_i for every g_i, i= 1..n.

III. Problem Analysis and Overview of Proposed Method

A. Problem Analysis

The compactness of a work cell is a spatial requirement to ensure that the work cell components occupy a minimal space. The task completion time, on the other hand, is a temporal requirement to reduce the task completion time and eventually achieve high productivity. Designing a compact work cell and satisfying the desired task completion time constraint is very complex. For example, the swept volume of the robot arm depends on the base placement and the motion of the robot arm (See Figure 2). In the two base placement settings shown, the change in the work cell size results into a corresponding change in the task completion. In Fig. 2(b), for instance, in order to achieve a minimal task completion time, the robot arm and the table have to move in a coordinated manner that results into a large swept volume.

Generally, the work cell size can be affected by (a) the intrinsic properties of the work cell components (e.g., the sizes and shapes of the robot arm and the table) and (b) the properties related to the task specifications such as $t_{desired}$ and goal pose, i.e., position and orientation. These properties are interrelated and analyzing their individual effects on the work cell size as well as on the resulting task completion time t_{actual} is too complicated for the following reasons: 1) collisions are possible among work cell components, 2) the problem is very heterogeneous, dealing with a combinatorial problem to derive goal order and an analytical problem on the motion coordination of the robot arm and the positioning table, and 3) the kinematic redundancy of the system leads to several possible configurations of the robot arm and the positioning table.

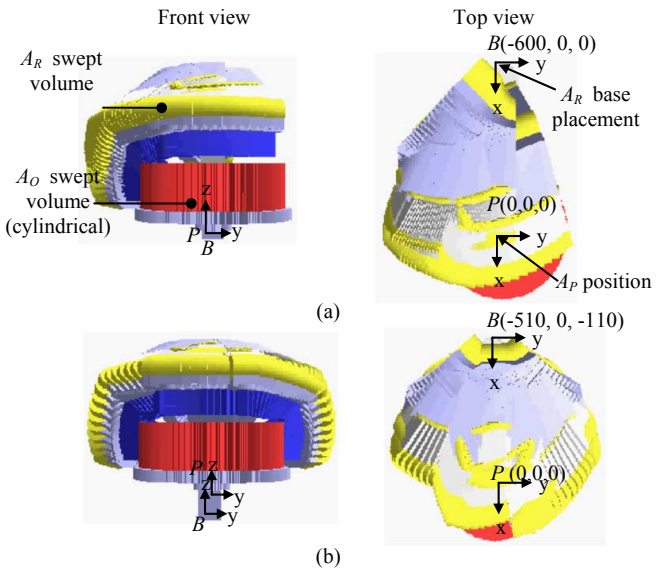

Fig. 2 Swept volume of work cell components. In (a) and (b), the same task is done but in different base placement settings of the robot arm. In (a), the work cell size is 0.484m^3 and the task completion time is 2.134s. In (b), the work cell size is 0.535m^3 and the task completion time is 1.577s.

B. Overview of Proposed Method

We propose a practical solution to this problem (Figure 3). The proposed method selects the most compact work cell or a work cell with the least size. The work cell size is calculated approximately through a method using bounding box, discussed in Section IVA. In minimizing the work cell size, the base placement optimization, goal rearrangement, and motion coordination are employed. The base placement optimization selects the best placement of A_R relative to A_P, discussed in detail in Section IVB. The goal rearrangement derive the goal order that is suitable for minimizing the work cell size and satisfying the task completion time constraint, discussed in detail in Section IVC. Since the system has 7 DOF, the redundancy is resolved through the motion coordination between A_R and A_P. Two motion coordination schemes are proposed and are discussed in detail in Section IVD. In selecting configurations, the collision and joint limit constraints are considered and thereby satisfied. After deriving the robot arm and table configurations, the task completion time t_{actual} is calculated using (1). As a constraint,

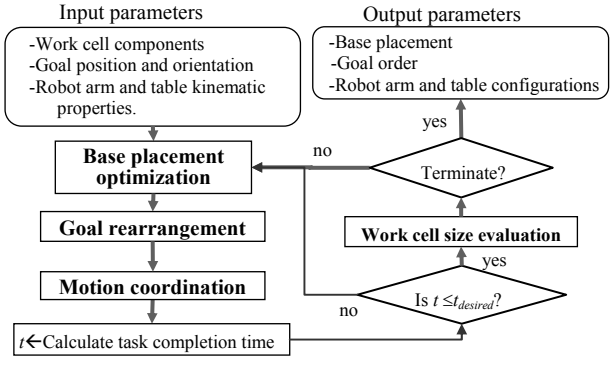

Fig. 3 Proposed method.

t_{actual} must be less than or equal to the desired task completion time $t_{desired}$. If this constraint is satisfied, then the work cell size is evaluated. Otherwise, the optimization will proceed to another possible solution. The above cycle terminates when the calculation time exceeded the design time limit (i.e., 30 minutes).

IV. Details of the Proposed Method

A. Work Cell Size Evaluation

The work cell size is determined by calculating the swept volume of the work cell components. The swept volume of A_R is very complex while that of A_P and A_T is a simple cylindrical volume, as a result of the rotation of A_P about the z-axis (See Fig. 2). In this paper, the swept volume is calculated by finding the least-size bounding box that encloses all work cell components. The space gaps between the work cell components are also enclosed in the bounding box. The occupied volume of the work cell components is calculated by first determining the maximum and minimum values of the x, y, and z coordinates of their swept volumes. Then, the difference of maximum and minimum values of *x, y,* and *z* values are calculated to derive the volume of the bounding box. In the case of A_T, the calculation of the swept volume is included with that of A_R since it is being held by the end-effector of A_R. As for A_P and A_O, the volume can be easily calculated as the bounding box of the swept cylindrical volume.

The approximate calculation of a work cell size is conducted due to the following reasons. 1) Calculating the exact swept volume of the work cell components, particularly that of the robot arm, is very complex and can require extensive calculation time. 2) In real manufacturing setting, work cells are designed as cubicles, which are rectangular in shape.

B. Base Placement Optimization

The base placement optimization is important in designing a work cell since the distance of the robot arm A_R relative to the positioning table A_P primarily dictates the floor area of the work cell and affects the resulting motion of A_R and A_P. In particular, Fig. 2 shows that the base placement is crucial in the work cell size with a corresponding tradeoff in the task completion time. Although it would be appealing to position A_R near A_P so as to reduce the floor area; but, this may result into a large swept volume of A_R and/or a long task completion time and therefore would violate the $t_{desired}$ constraint. Furthermore, the existence of potential collisions and the nonlinearity of the robot arm complicate the base placement optimization. Hence, it can not be treated as a simple nonlinear optimization. As a solution, a direct search method is employed using simulated annealing (SA).

The SA, which is based on a stochastic local neighbor search method, can be robust and capable of avoiding local minima by selecting a solution that can be worse than a current solution (i.e., a base placement design with a work cell size larger than that of the previous design). The probability of accepting a worse solution is too high at the start and is subsequently reduced as the optimization proceeds. In this paper, we define a neighborhood as a set of solutions found by incrementing or decrementing the design variables by a constant step size (i.e. 10 [mm]). See [16] for detailed discussion on SA.

C. Goal Rearrangement

We briefly describe the algorithm for the goal rearrangement in this section. See [15] for details. To reduce the calculation time, the goals are clustered into groups based on their locations on the planar boundaries defined by the approximate box enclosing object A_O. With the clustering, the algorithm derives (a) the order of clusters and (b) the goal order in every cluster. The order of clusters (a) is solved using the 2-opt algorithm, which exchanges the order of two clusters, while (b) is derived using the Lin-Kernighan algorithm. The stopping condition for deriving the goal order is when the exchange in the 2-opt algorithm did not result to a better solution or when the calculation time limit is exceeded.

D. Motion Coordination Schemes

In the motion coordination, the robot arm and table movements are coordinated synchronously to select configurations that are collision-free and that would satisfy a particular performance index. In this paper, we introduce a motion coordination scheme to minimize the work cell size, referred to as the *spatial* motion coordination. In our previous study [15], we propose a motion coordination scheme to minimize the task completion time, which is referred to as the *temporal* motion coordination. We emphasize on the following points in regard to these two schemes.

1) If the optimization has no $t_{desired}$ constraint or the value of $t_{desired}$ is quite large (i. e., the task completion time is not quite critical in the optimization), the spatial motion coordination is solely appropriate for designing a compact work cell.

2) The temporal motion coordination scheme may still be suitable for designing a compact work cell. It may seem contradictory but is valid for the following reasons. a) The temporal motion coordination is only applied in the motion coordination level of the optimization (i.e., selection of configurations); this means that a compact work cell can still be evaluated as the performance index of the entire optimization. b) The temporal motion coordination, which minimizes the motion time, can be favorable in the optimization in which a $t_{desired}$ constraint has to be satisfied.

We employ a graph-based search in the motion coordination. The steps in the graph search are as follows: (a) Definition and creation of vertices (i.e., the configurations of robot arm and table); (b) Calculation of edge weights (i.e., the work cell size in the spatial motion coordination scheme, and the task completion time in the temporal motion coordination scheme), and (c) Selection of vertices in the graph. The two motion coordination schemes differs in (b) and (c). That is, the steps on how the edge weights are calculated and how the vertices are selected. These steps are described in detail as follows.

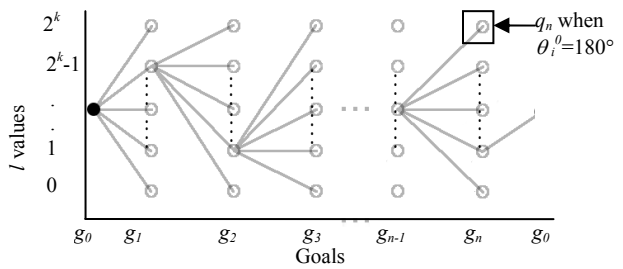

Fig 4. Search space in motion coordination . There are $n+2$ stages where the first and last stage correspond to goal g_o, the home position of the robot arm and the positioning table. The parameter l defines the rotation angle of the table, θ_i^0.

1) Definition and creation of vertices in the graph

For concreteness, we consider the search graph shown in Figure 4. The i^{th} goal in a goal order corresponds to stage i. A vertex is a configuration q_i of A_R and A_T when reaching goal g_i. The set of vertices belonging to the same column corresponds to the possible configurations of the robot arm in reaching a single goal that is rotated by A_P in several angles. In creating vertices of every goal, the joint angle value of table θ_i^0 is selected and $\{\theta_i^1 \ldots \theta_i^6\}$ are determined by solving the inverse kinematics (IK) of the robot arm; this is done to reduce the number of unknowns. We denote a parameter l to describe the possible values of $\boldsymbol{\theta_i^0}$, which is defined in (7) and (8).

$$\boldsymbol{\theta_i^0}(l) = \theta^{0,min} + l*d(k) \ , \ l \in \{0,1 \ldots 2^k\text{-}1, 2^k\} \quad (7)$$

$$d(k) = (\theta^{0,\max} - \theta^{0,\min}) / 2^k \ , \ k \geq 1 \quad (8)$$

where k is a user-defined step resolution of table rotation, d is the step, $\theta^{0,min}$, and $\theta^{0,max}$ are the maximum and minimum search limit values for θ_i^0. For a large k value, the search space is quite dense and may require a long calculation time. In this paper, the k value is 6, equivalent to the step resolution of 5.625°, which is adequate enough to obtain a good solution. To cover one whole rotation of the positioning table, $\theta^{0,min}$ and $\theta^{0,max}$ are set to -180° and 180°, respectively.

2) Calculation of the edge weights

Every edge connecting two vertices in the graph is weighted depending on the motion coordination scheme.

-- For the spatial motion coordination, the weight is denoted as $\Delta V(g_{i-1}, g_i)$, defined as the change in the work cell volume when moving from g_{i-1} to g_i. This is calculated as:

$$\Delta V(g_{i-1}, g_i) = V(g_i) - V(g_{i-1}) \quad (9)$$

where $V(g_i)$ is the total swept volume of the work cell components from t=0 until the time A_R reaches g_i or the time when the motion coordination is at stage i.

--For the temporal motion coordination, the weight is the motion time $c(g_{i-1}, g_i)$, which is calculated based on the maximum velocity of joints as shown in (10). This is done for simplicity, which assumes that A_R and A_P are able to achieve its maximum velocity at negligible amount of time.

$$c(g_{i-1}, g_i) = \max_{j=0\ldots6} (|\theta_i^j - \theta_{i-1}^j| / s^{j,max}) \quad (10)$$

where $s^{j,max}$ is the maximum speed of joint j.

3) Selection of vertices in the graph

In selecting the configurations of A_R and A_P, the greedy nearest neighbor algorithm is used, which selects a configuration q_i that has the least weight value at every stage, provided that it satisfies the collision constraint (5) for collision-free motion of A_R and A_P, and the joint limit constraint (6). In the spatial motion coordination, a q_i is selected that has the least $\Delta V(g_{i-1}, g_i)$ while in the temporal motion coordination, the q_i with the least $c(g_{i-1}, g_i)$ is selected. The selected q_i at every stage corresponds to the vertex in Fig. 4 from where the edges are emanating towards the next stage. The nearest neighbor algorithm is used since it requires less calculation time. Other algorithms (e.g., Dijkstra algorithm) can be used to derive a high-quality solution but may require long calculation time.

When two or more configurations have the same weight values, the following rules are applied:

-- For the spatial motion coordination, the configuration that has the minimum motion time is selected.

-- For the temporal motion coordination, the configuration that has the least increase in volume is selected.

The above rules are practical in the aspect of achieving a compact work cell with the task completion time constraint.

In selecting the configuration q_i, the motion of A_R and A_P from one goal to another goal are based on straight-line paths in the configuration space. If collision is detected, other configurations are tested again for collision, which is possible since the system is kinematically redundant. To detect collision, the work cell components are modeled as oriented bounding box (OBB), appropriate for modeling rectangular shapes. See [17] for details of OBB.

V. Simulations, Results and Discussion

In this section, we empirically evaluate the performance of the proposed method, i.e., the integration of the base placement (BP) optimization, the goal rearrangement and the motion coordination. The performance of the two motion coordination schemes are evaluated under various $t_{desired}$ settings.

A. Simulation and Compared Methods

First, we consider a simulation without imposing the $t_{desired}$ constraint. This can be considered as designing a compact work cell with no critical requirement in the task completion time. In this simulation, we are interested in comparing the proposed method and a method without the BP, which employs only the goal rearrangement and the motion coordination. Without employing the goal rearrangement and the motion coordination, the task cannot be executed due to collision.

Second, a simulation is conducted in which the value of $t_{desired}$ is varied to compare the performance of the two motion coordination schemes: the temporal motion coordination (TMC) and the spatial motion coordination scheme (SMC).

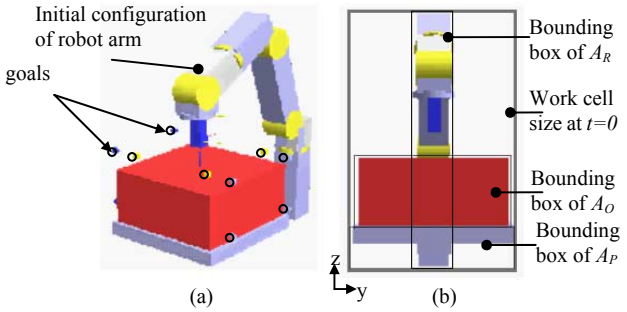

Fig. 5 Initial work cell setting (a) and the work cell size at the initial setting.

B. Simulation Settings and Parameters

Figure 5 shows the initial setting in the simulation. The positioning table is located at (0, 0, 0) while the initial base placement of robot arm, $({}^{P}x_B, {}^{P}y_B, {}^{P}z_B)$, is set to (-600[mm], 0, 0), which is derived based on an empirical method [10]. In this method, the position of the robot arm at ${}^{P}x_B$ = -600[mm] is set to be approximately 70% of its reach from g_0. The value of ${}^{P}y_B = 0$ is selected to provide a balanced reach for robot arm, since A_O occupies -200[mm]≤ y≤ 200[mm]. With ${}^{P}z_B$ = 0, the effective height of the object when placed on the table (i.e., 300[mm]), is quite appropriate to the robot arm base link (i.e., 335[mm]).

The object A_O is a parallelepiped with a size of 400[mm] by 400[mm] by 200[mm], the dimension of which is comparable to that of the robot arm links. The initial end-effector position g_0 is located at (0, 0, 500[mm]) with orientation (0, 180°, 0), just above the object; this is suitable for designing a compact work cell since the projection of the robot arm towards the floor layout coincided with that of A_T and A_O, resulting into a minimal initial work cell size (See Fig. 5).

We consider a task with 12 goals that are located on the three faces of A_O, i. e. 4 goals are sparsely located on each face. A task with only few goals is considered so that a high-quality solution can be obtained. If the number of goals is large, the simulation may require a longer design time limit. The simulation is done using a Core 2 Duo 3.0GHz processor with 4GB RAM.

C. Results and Discussion

Table I shows the derived work cell size and the corresponding task completion in the optimization without $t_{desired}$ constraint. The proposed method combining SMC has the best performance with about 38% reduction in the size of the work cell compared to the method without the BP optimization (0.283[m^3] vs 0.456[m^3]). This result shows that integrating the BP optimization with goal rearrangement and motion coordination is quite effective in designing a compact work cell. With and without the BP optimization, the SMC performs better than the TMC. This is expected since the SMC selects q_i on the basis of the work cell size. Without the BP, the work cell size obtained by the TMC and the SMC are comparable (0.484[m^3] against 0.456[m^3]), which is because of having the same base placement setting.

TABLE I
WORK CELL SIZE MINIMIZATION WITHOUT $T_{DESIRED}$ CONSTRAINT

	Without BP		Proposed method (with BP)	
	TMC	SMC	TMC	SMC
V [m^3]	0.484	**0.456**	0.373	**0.283**
t_{actual} [s]	2.134	2.367	2.139	2. 769

Figure 6 illustrates the base placement of the robot arm and the swept volume of the work cell components as a result of employing the proposed method combined with the SMC. With the BP optimization, the motion of the robot arm in the entire task execution is focused on a certain region that results into a compact work cell size; for comparison see Fig. 2(b). In Fig 6(b), there are portions where the robot arm end-effector moves into a region directly above the table. This can be seen clearly in the accompanying video of this paper. In the video, the robot arm end-effector can be seen as moving to that region with its joint links stretched. This movement does not result into substantial increase in the swept volume since that region is a part of the previously swept volume. If there is no increase in the swept volume, the configuration with the minimum task completion time is selected in the motion coordination.

Figure 7 shows the work cell size obtained by the proposed method under the various $t_{desired}$ settings. In $t_{desired}$= 1.75[s], the proposed method with the SMC did not satisfy the desired task completion time constraint while that with the TMC satisfied this constraint. The $t_{desired}$ =1.75[s] can be viewed as a very restrictive constraint for the SMC. On the other hand, it is suitable for the TMC since at the motion coordination level the TMC selects the q_i on the basis of the least motion time. In a way, this provides leverage for the proposed method with the TMC to derive a short task completion time and minimize the work cell size. In $t_{desired}$= 2.0[s] and $t_{desired}$= 2.25[s], the performance of the proposed method combined with the SMC relative to that with the TMC is about 20.87% and 22.81%,

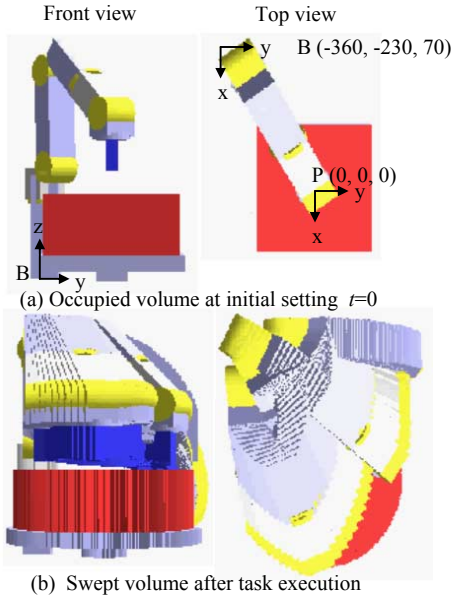

Fig. 6 Swept volume employing the proposed method combined with the SMC and without the $t_{desired}$ constraint.

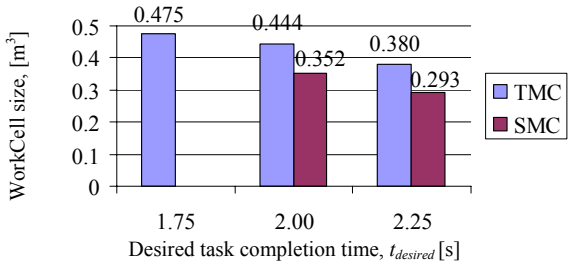

Fig 7. Work cell size optimization using the proposed method under various $t_{desired}$ settings. The spatial motion coordination (SMC) is not able to satisfy the constraint of $t_{desired}$= 1.75[s]; but it performs better than the temporal motion coordination (TMC) in the two other settings.

respectively, which shows that the proposed method with SMC is very effective in designing a compact work cell under these $t_{desired}$ settings. If $t_{desired}$=2.5[s], it is expected that the work cell size derived by the proposed method with the SMC to be less than or equal to 0.293[m^3], which is the derived solution at $t_{desired}$=2.25[s]. By looking at the corresponding task completion time of the derived minimum work cell size in Table I, which is 2.769[s], we can further say that if $t_{desired}$ >2.769[s], a minimum work cell size can be derived. Moreover, the SMC is appropriate as the motion coordination scheme for $t_{desired}$ >2.769[s]. Therefore, a reasonable basis in designing a compact work cell with respect to the $t_{desired}$ constraint is the result of the compact work cell design without the $t_{desired}$ constraint. Another basis is the result of the task completion time minimization in order to determine the minimum $t_{desired}$ value that can be satisfied. For example, given the same work cell setting, we performed the task completion time minimization using the base placement optimization, the goal rearrangement and the TMC. (Note that TMC is used as the motion coordination scheme since the task completion time is minimized in this instance.) The derived minimum task completion time is 1.577[s]. If $t_{desired}$ <1.577[s], then the $t_{desired}$ constraint becomes too restrictive and no solution will apparently be derived in the compact work cell design. Further, if the value of $t_{desired}$ is near but greater than 1.577[s] (e.g., $t_{desired}$= 1.75[s]), the TMC is applicable to be used as the motion coordination scheme, as reflected in the result in Fig. 7. The two above-mentioned bases pertain only to the two possible crucial values of $t_{desired}$. Determining, however, the range of $t_{desired}$ values on which of the TMC and the SMC will be the appropriate motion coordination scheme is difficult and is currently pursued in our study.

VI. Conclusion and Plans

In this paper, a problem dealing with a compact work cell design with a task completion time constraint is evaluated. The compact work cell design is evaluated based on the swept volume of the work cell components. To design a compact work cell and satisfy the constraint, the integration of the base placement optimization, goal arrangement and motion coordination between the robot arm and the positioning is proposed. Two motion coordination schemes are introduced based on the task completion time and the work cell size and are compared based on their applicability in satisfying the task completion time constraint.

It is observed that the work cell size and the task completion time can be conflicting requirements in the compact work cell design. Further studies will be undertaken based on this observation. Since the nearest neighbor algorithm is sensitive to local minima, other algorithms such as the Dijkstra algorithm can be used in the motion coordination.

Acknowledgment

This work was supported in part by Global COE Program "Global Center of Excellence for Mechanical Systems Innovation", MEXT, Japan.

References

[1] M. K. Jouaneh, Z. Wang, and D. A. Dornfeld, "Trajectory planning for coordinated motion of a robot arm and a positioning table: Part 1- Path specification," IEEE Tans. on Rob. and Auto., vol. 6, no. 6, 1990.

[2] A. Kusiak and S. S. Heragu, "The facility layout problem," European Jour of Oper. Res., vol. 29, pp. 229-251, 1987.

[3] M. L. Tay and B. K. A. Ngoi, "Optimising robot workcell layout," Int. J. Adv. Manuf. Technology, vol. 12, pp. 377-385, 1996.

[4] T. C. Leuth, "Automated Planning of Robot Workcell Layouts," in *Proc. of IEEE International Conference on Robotics and Automation,* Nice, France, pp. 1103-1108, 1992.

[5] Z. Shiller, "Optimal robot motion planning and work-cell layout design," Robotica, vol. 15, pp. 31-40, 1997.

[6] J. Cagan, D. Degentesh and S. Yin, "A simulated annealing-based algorithm using hierarchical models for general three-dimensional component layout, Computer-Aided Design, vol. 3, pp. 781-790, 1998.

[7] D. Hsu, J.C. Latombe, and S. Sorkin, "Placing a robot robot arm amid obstacles for optimized execution," in *IEEE Int. Symp. on assembly and task planning*, pp. 280-285, 1999.

[8] S. Mitsi, K.-D. Bouzakis, D. Sagris, G. Mansour, "Determination of optimum robot base location considering discrete end-effector positions by means of hybrid genetic algorithm," Robotics and Computer-Integrated Manufacturing, vol. 24, pp. 50-59, 2008.

[9] J. A. G. Pamanes and S. Zeghloul, "Optimal placement of robotic robot arms using multiple kinematic criteria," in *Proc. IEEE Int. Conf. on Robotics and Automation* , pp. 933-938, 1991.

[10] L.B. Gueta, R. Chiba, J. Ota, T. Arai and T. Ueyama, "Design and optimization of a manipulator-based inspection system," Society of Instruments and Control Engineers Transaction on Industrial Application, vol. 6, no. 6, pp. 41-51, 2007.

[11] J.-C. Latombe, *Robot Motion Planning*. Norwell, MA: Kluwer Academic Publishers, 1991.

[12] M. Saha, T. Roughgarden, and J.-C. Latombe. "Planning tours of robotic arms among partitioned goals," The Int. Journal of Rob. Research, vol. 25, no. 3, pp. 207-224, 2006.

[13] S.N. Spitz and A.A.G. Requicha, "Multiple-goals path planning for coordinate measuring machines," in *Proc. in IEEE Int. Conf. on Robotics and Automation*, vol. 3, pp. 2322-2327, 2000.

[14] B. Cao, G. I. Dodds and G. Irwin, "A practical approach to near time-optimal inspection-task-sequence planning for two cooperative industrial robot arms," The Int. Journal of Rob. Research, vol. 17, no. 8, pp. 858-867, 1998.

[15] L. B. Gueta, R. Chiba, J. Ota, T. Arai and T. Ueyama, "Coordinated motion control of a robot arm and a positioning table with arrangement of multiple goals," *in Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 2252 – 2258, 2008.

[16] A. Gosavi, Simulation-based optimization: parametric optimization techniques and reinforcement learning, Kluwer Academic Publishers, Norwell, MA, 2003.

[17] S. Gottschalk, "OBB-tree: A hierarchical structure for rapid interference detection," Comp. Graphics, pp. 171-180, 1996.