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

A Path Planning Method for Dynamic Object Closure by Using Random Caging Formation Testing

ZhiDong Wang[1], Hidenori Matsumoto[1], Yasuhisa Hirata[2] and Kazuhiro Kosuge[2]
[1]Dept. of Advanced Robotics, Chiba Institute of Technology, Japan
[2]System Robotics Lab., Tohoku University, Japan
zhidong.wang@it-chiba.ac.jp, {hirata, kosuge}@irs.mech.tohoku.ac.jp

***Abstract*— Object manipulation problem by multiple cooperating mobile robots using the concept of *Object Closure* is discussed in the paper. It is the condition under which the object is trapped so that there is no feasible path for the object from the given position to any position that is beyond a specified threshold distance during the transportation. We proposed the concept of *Dynamic Object Closure* for achieving object caging task that robots team is able to cage a moving object after a predefined time interval. In the paper, a method planning caging path and by using *Random Caging Formation Path Determination* algorithm (RCFP) is proposed for achieving *Dynamic Object Closure*. Some planning results are presented for illustrating the validity of the proposed algorithm.**

***Index Terms*— Dynamic Caging, Dynamic Object Closure, Cooperative Object Handling, Path Planning**

I. Introduction

Inspired from human and animal behaviors, various cooperative control methods are developed for multiple robot system. One significant example is cooperative object transportation which is mainly using strategies of grasping or pushing which are similar to cooperative motion of human being. Recently a novel object handling strategy, caging based object handling strategy, has been discussed[15][17][20]. The similar behaviors are observed from cooperative hunting in some wolves, dolphins and humpback whales groups (Fig.1). The most important advantage of robotic caging strategy is that contacts between object and robots need not to be maintained by robot's control. This is not only able to let the robot handle an object without any grasping mechanism but also able to make motion planning and control of each robot become simple and robust, and realize coordinative object handling without direct contacting force control, etc. This condition is also called as *Object Closure*. Recently, several algorithms have been proposed to solve this problem, but all of them are only discussing on caging a stationary object or with fixed caging style.

However, in the case of group hunting by animals, etc, object caging tasks are usually happened with a moving target rather than a stationary object in general. Additionally, many team playing sports, such as football and basketball, also include many factors of dynamic object caging and manipulation. Similarly, *Dynamic Object Caging* by multiple robots could be an interesting, important and challenging research topic, and will have more applications than trapping a stationary object. In this research, we defined the concept of *Dynamic Object Closure* for achieving object caging to a moving object[24]. In this paper, two margin metrics are defined, and a *Random Caging Formation Path Determination* algorithm (RCFP) is proposed for not only checking *Dynamic Object Caging* condition but also deciding the path and formation for realizing a *Dynamic Object Caging*. Simulation results are provided for illustrating the validity of the proposed algorithm.

This work is partially supported by Grant-in-Aid for Scientific Research (B)(20360120), Japan.

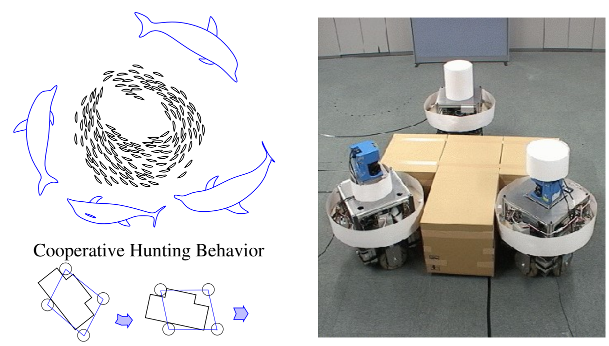


Fig. 1. Examples of Object Caging by Animals and Multiple Mobile Robots

II. Background and Related Works

In the last decade, many researchers[5][6][12][18][19] working on cooperative object handling using multiple mobile robot are focusing on grasping based manipulation which *Form Closure* or *Force Closure* condition is achieved by robots' contacts. These two closure conditions guarantee that robot mechanisms can resist any external force and can generate any acceleration to the object, as one of the most important closure conditions for researches of multiple finger/multiple arm system. By incorporating gravity force, inertia, friction force, etc as an extra closure effector, *conditional closure* based manipulation was investigated for object pushing task[9] and other task for relative small number of robots.

For easing the stringent condition used in these approaches, our research is based on the notion of *caging* defined and studied in [15][17]. The key issue is to introduce a bounded movable area for the object during its transportation and manipulation so that coordinative object handling among robots can be achieved without direct contacting force control, etc. We proposed a potential field based strategy for leading robots to a formation which could achieve the *Object Closure* and presented a sufficient and necessary condition for testing the *Object Closure* condition[20]~[22]. Our work is closest to the work by Sudsang and Ponce[17] who develop a centralized algorithm for three disc-shaped robots to push the object in sequence. But they carefully discussed the *Object Closure* in the configuration space and make our algorithm have ability to cope with large number robots and complicated object. In [13], Pereira and Kumar proposed an efficient testing algorithm for convex polygon robots and showed experiment results with three non-holonomic robots. This algorithm is also decentralized. But it is under the assumption that the robots group does not have non-essential robots and closest pair of points between two neighbor robots can be decided. Recently, Fink and Kumar demonstrated the caging based object transportation with 8 robots [3]. Also, Makita and Maeda proposed a planning method for realizing a caging style grasping with a three finger hand[10]. In [23], we proposed a complete and efficient distributed algorithm that all searching and updating procedures are implemented by using $\rho-\theta$ representation of the *C-Closure Object* and the *CC-Closure Object*.

Caging a moving object is also an instance of the motion planning domain. Planning problems with moving obstacle were addressed by augmenting the configuration space with time and a point in the free space denotes a valid configuration of the object at the given time[7][14]. Recently, this approach has been extended to real-time deformable plans[1], kinodynamic domains[4], and interactions with multiple objects mainly on assembly planning. Additionally, some new results beyond assembly planning have been reported[8][12][16] too.

III. Dynamic Object Closure

A. Definition of Object Closure

In this paper, we assume that all robots have the same size and shape, can contact with the object in any direction, and are holonomic. Also we assume that robots can estimate the geometric properties (the mass center and shape) of the object and its neighbor robots. The *Object Closure* is that there is not any path from current object's position/orientation to outside. We discuss this problem in C-space. Let A_{obj} denote the object, and A_i, $i=1,\cdots,n$, denote the caging effector i in the working space. A configuration $\boldsymbol{q}=(x,y,\theta)$ in the C-space C, is a specification of position and orientation of a caging effector or an object. *C-Closure Object*(C_{cls_i}) and *C-Closure Object Region*(C_{cls}) is defined as:

$$C_{cls\_i} = \{\boldsymbol{q}_{obj} \in C \mid A_{obj}(\boldsymbol{q}_{obj}) \cap A_i(\boldsymbol{q}_i) \neq \emptyset\} \quad (1)$$

$$C_{cls} = \bigcup_{i=1}^{n} C_{cls\_i} \quad (2)$$

$C_{cls\_i}(\theta_0)$ and $C_{cls}(\theta_0)$ is the subset (a slice) in the C_{cls_i} and in the C_{cls} respectively with orientation θ_0. Let $\boldsymbol{q}_{obj} \notin C_{cls}$ be a free initial configuration of the object. We define set $C_{free} = C \backslash C_{cls}$ and define set C_{free_obj} as follows:

$$C_{free\_obj} = \{\boldsymbol{q} \in C_{free} | \; connected(\boldsymbol{q}, \boldsymbol{q}_{obj})\}. \quad (3)$$

We define $\boldsymbol{q}_{inf} \in C_{free\_inf}$ as a generic point that is sufficiently far away from the object. In this paper, we will be concerned with the problem of keeping the position of the caged object contained but not the orientation. Thus C_{free_inf} will have the structure of a generalized cylinder. An object can escape from robots only when the $\boldsymbol{q}_{obj}$ connects to the $\boldsymbol{q}_{inf}$ in C-space. Then *Object Closure* can be defined as follow:

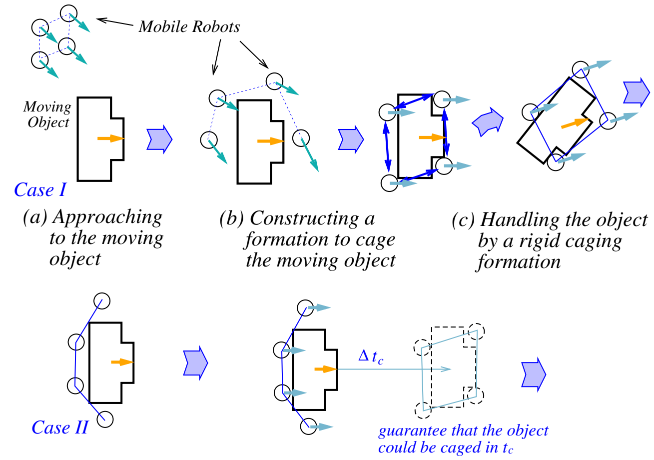


Fig. 2. Dynamic Object Caging by Multiple Mobile Robots. (a) A group of robots approach to and trap a moving object. (b) A robot team is pushing an object with a formation which does not satisfy the caging condition now but guarantees that robots can cage the object after a predefined time interval.

Proposition 1 (*Object Closure*): *Let $\boldsymbol{q}_{obj}$ is the current configuration of the object. The object is in Object Closure if, and only if, the following conditions are satisfied.*

$$\begin{cases} C_{free\_obj} \neq \emptyset, \; \{\boldsymbol{q}_{obj}\} \\ C_{free\_obj} \bigcap C_{free\_inf} = \emptyset \end{cases}$$

When *Object Closure* is achieved, there is a bounded free space(C_{free_obj}) around the $\boldsymbol{q}_{obj}$, which is entirely kept inside of the C_{cls}. On the other hand, *Object Closure* is not satisfied when a connection path exists between C_{free_obj} and C_{free_inf}.

B. Dynamic Object Caging

Similar with examples of animals' group hunting and cooperative team-playing behavior by human being, two typical cases exist for dynamic caging with multiple mobile robots. The first case is that a team of robots approaches to and finally traps an object which is moving from other place (Fig.2-(a)). Another situation is that robots are handling an object with a formation which does not satisfy the caging condition in the current moment. However from this formation, the robots team can always move to a caging formation in a predefined time interval (Fig.2-(b)). In the one word, *Dynamic Object Caging* is that:

> *there exists at least one set of paths to guarantee that by moving along the paths, the robot team can achieve Object Closure to the target object in predefined future time.*

It is easy to know that discussion on configurations of the object and robots in the current moment is not enough. Let $\mathcal{T} \subset R$ denote the *time interval*. Let the state space, X, be defined as $\mathcal{X} = \mathcal{C} \times \mathcal{T}$, denote the CT-space, consisting of configuration space and time space. In this paper, we discuss *Dynamic Object Caging* in this state space.

In $\mathcal{X}$, we can represent a moving object as an obstacle region, $\mathcal{X}_{obj}$, and robots moving around the target object as continuous and time-monotonic paths, $(\mathcal{X}_{rbt\_i}, i = 1, \cdots, n)$, around the obstacle region of the target object respectively. It is well known that all objects in CT-space should follow a constraint that they must move forward in time. For a given $t \in \mathcal{T}$, a slice of the obstacle region of the object and paths of robots, $\mathcal{X}_{obj}(t)$ and $\mathcal{X}_{rbt\_i}(t)$, can be obtained as *C-objects* at particular time which is possible be used for testing *Object Closure* condition at this instant. For convenience on discussing *Object Closure*, we use new representation in $\mathcal{X}$ which sets the obstacle region of the object as a path, and represents all robots as obstacle regions respectively. This representation does not change proprieties of *Object Closure*. Then we define *Dynamic Object Closure* as:

Proposition 2 (*Dynamic Object Closure*): *The object is in Dynamic Object Closure with a given time interval $\Delta t_c \in \mathcal{T}$ from the current moment t_c if, and only if, to any feasible configuration of the object at $t \in \mathcal{T}$, $\boldsymbol{q}_{obj}(t)$, there exists a set of feasible configurations of robots, $\boldsymbol{q}_{rbt\_i}(t), i = 1, \cdots, n$, so that the following conditions are satisfied for all $t \geq t_c + \Delta t_c$.*

$$\begin{cases} C_{free\_obj}(t, \boldsymbol{q}_{rbt\_i}(t)) \neq \emptyset, \ \{\boldsymbol{q}_{obj}(t)\} \\ C_{free\_obj}(t, \boldsymbol{q}_{rbt\_i}(t)) \bigcap C_{free\_inf}(t, \boldsymbol{q}_{rbt\_i}(t)) = \emptyset \end{cases}$$

This definition indicates that testing *Dynamic Object Closure* condition could be considered as a problem to check if the *Object Closure* condition will be guaranteed after a predefined time intervals. Configuration of a moving object is governed by the dynamics of the object and external forces applied on the object, such as friction force, etc. On the other hand, the feasible configuration region of each robot in the future time is not only governed by the dynamics of the robot but also affected by the limitation of actuator outputs. Additionally, collision free constrains to the target object and other robots should be satisfied (Fig.3). These let the testing procedure of *Object Closure* after a predefined time intervals be complicated.

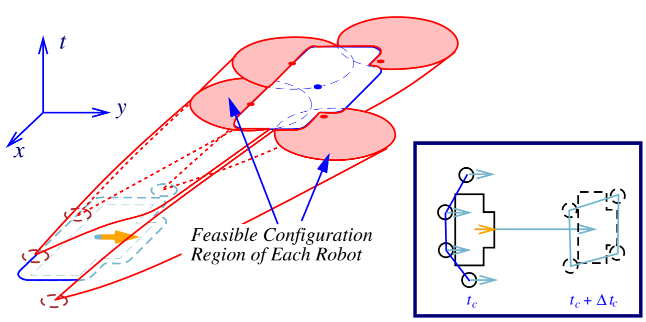


Fig. 3. Feasible Configuration Region of Each Robot during Dynamic Object Caging (in the CT-space).

We have the following assumptions in this paper for simplifying analysis of the problem.

- Motion of the target object within the predefined time interval Δt_c can be estimated completely.
- All robots are disc-shaped and current velocity and the maximum acceleration of all robots are known.
- All robots can maintain a caging formation once they reach it.

The first assumption means that we will know where the object will be at $t_c + \Delta t_c$. By using the first and second assumptions, each robot can calculate its region where it can reach after Δt_c without colliding with the moving object. Also in the case that robot is disc-shaped, feasible configuration region of each robot at $t_c + \Delta t_c$ is independent of robot's orientation. The last assumption let us can only concern *Object Closure* condition at $t_c + \Delta t_c$ when *Dynamic Object Closure* is discussed.

Of course, in the really world, position and orientation of an object are hard to be estimated precisely because of the measuring error of object motion, unexpected disturbances and other uncertainties. Then a reasonable assumption is that in the C-space, object's configuration will be in a bounded region. In general, this region is not very large if the time interval is not very long. Fortunately, different from object grasping case, caging is a loose closure strategy with certain margin and allows changing the size of a caging formation within the margin[22]. It can be said that, in many cases, the first assumption is reasonable if the margin of an obtained caging formation is larger than the uncertainty of the estimation error of object's configuration.

IV. Basics of Object Closure Testing

Here, we consider the basic method of *Object Closure* testing first. In general, C_{cls} and C_{free_obj} are complicated in shape and are very hard to calculate, especially when the shape of the object is relatively complicated or the number of robots is large. Here, the conditions for *Proposition 1* by taking slices in C-space are checked. For reducing complexity of the testing procedure, a sufficient condition of *Object Closure* can be derived as follow.

Since the evaluation for *Object Closure* must run in real time, the computation cost should be low. To solve this problem, We proposed a concept of *CC-Closure Object* in [20] and [21] which is useful on building an efficient testing algorithm even if the shape of the object is complicated. They defined a new C-space for the *C-Closure Object* C_{cls_i} and denote it as CC. C-Object of a *C-Closure Object* in CC (here, $i \neq j$) is called *CC-Closure Object*:

$$CC_{cls\_ij} = \{\boldsymbol{q}_j \in CC \mid C_{cls\_i}(\boldsymbol{q}_i) \cap C_{cls\_j}(\boldsymbol{q}_j) \neq \emptyset\} \quad (4)$$

which indicates the C-Obstacle of ith *C-Closure Object* to jth *C-Closure Object*. Then the problem in checking if two regions are connecting or overlapping can be replaced by a simpler problem: a point in a region or not.

$$C_{cls\_i}(\theta) \bigcap C_{cls\_j}(\theta) \neq \emptyset \quad \Leftrightarrow \quad \boldsymbol{p}_j \in CC_{cls\_ij}(\theta) \quad (5)$$

Here, $\boldsymbol{p}_j$ is the position of C_{cls_j}, and θ is the orientation angle. Because the size and the shape of C_{cls_i} and CC_{cls_ij} is not changed during the manipulation, they can be calculated in advanced for reducing the runtime calculation cost even shape of the object is complicated.

In testing the overlapping conditions for *C-Closure Objects*, one just needs to calculate one ρ-θ curve in advanced

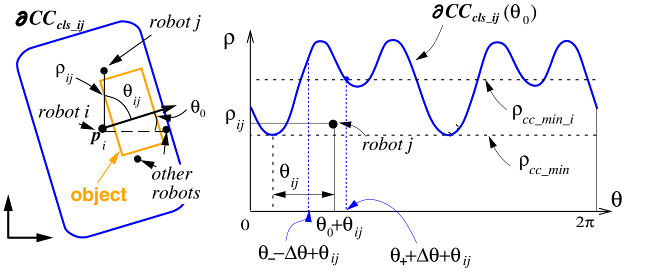


Fig. 4. A slice of *CC-Closure Object* is represented in the ρ-θ coordinates with bounded rotational angle of the object constrained by the caging edge.

and shift the θ_0 for checking if the distance to jth robot (ρ_{ij}) is under the $\partial CC_{cls\_ij}(\theta)$ curve on the part or all orientation.

Proposition 3b: (*Sufficient Condition*) *Let θ_0 satisfy*

$$\begin{cases} C_{free\_obj}(\theta_0) \neq \emptyset, \ \{\boldsymbol{q}_{obj}\} \\ C_{free\_obj}(\theta_0) \bigcap C_{free\_inf}(\theta_0) = \emptyset. \end{cases}$$

A sufficient condition of the Object Closure is

$$\begin{cases} \rho_{i(i+1)} \leq \rho_{cc\_min} & i = 1, \cdots, n-1 \\ \rho_{1n} \leq \rho_{cc\_min} \end{cases}$$

. Here, ρ_{cc_min} is the minimum value of $\partial CC_{cls\_ij}(\theta)$ curve and ρ_{ij} is the current distance from ith robot to jth robot.

The calculation cost of Proposition 3b in the run time is extremely low. It just need n inequality checking to robots when the ρ-θ curve of the $CC_{cls\_i}(\theta)$ is pre-calculated.

V. Test Formation of Dynamic Object Caging

As mentioned in the previous section, by using the sufficient condition of *Object Closure*, the calculation cost will be extreme low if configurations of object and all robots are given. In *Dynamic Object Caging*, the feasible configuration of each robot is a set of configurations, which is a compact region, rather than a single one. Then it is necessary to check if a caging formation exists in the feasible configuration regions of all robots at time $t_c + \Delta t_c$ (Fig.5).

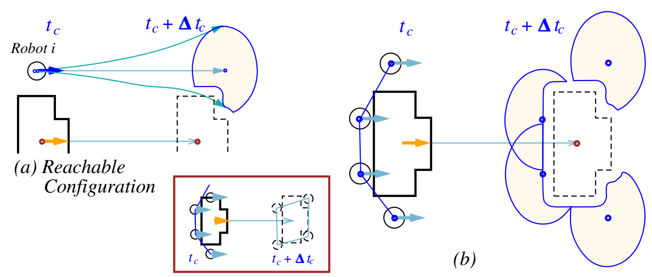


Fig. 5. Reachable Configuration Sets of Robots at t_c.

It is not very hard to calculate reachable configuration region of a robot which is moving in the free space when the dynamics of the robot and output limitation of robot actuators are known. But here, robots are moving with the target object, and collision free conditions with the moving object and other robots should be satisfied. Therefore, the shape of the feasible configuration region will be very complicated especially when the object is not a simple shape one. In this research, rather than calculating geometrical shape of each robot's feasible configuration region and obtaining a caging formation using this geometrical shape information directly, we obtain the caging formation at time $t_c - \Delta t_c$ by using probabilistic strategy. This allow the *Dynamic Object Closure* testing procedure to be running in real-time[1]. Based on the Proposition 2 and 3b, *Random Caging Formation Path Determination* (RCFP) algorithm is designed as follow:

```
procedure RandomCagingFormationPath(t):
begin
  Obtain p_obj(t);
  For All Robots ( i = 1, ..., n)
    P_i = RandomPathSet (t);
    R_i = FeasibleConfigurationSet (P_i, t, p_obj(t));
  S_ce = CagingEdgesSet (R_1, ..., R_n, ρ_cc_min);
  S_cf = ClosedCagingFormation (S_ce);
  S_pm = PathMarginSet (R_1, ..., R_n, S_cf);
  CF = CagingPathFormation (S_ce, S_pm);
  If (CF != NULL) return True;
  else            return False;
end;
```

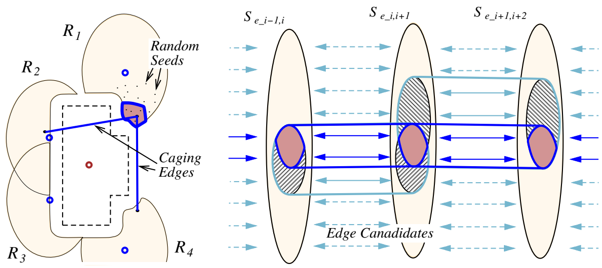
This algorithm consists of three main parts; obtaining the feasible configuration set of each robot at a future time t, checking existence of a closed caging formation from all robots' feasible configuration sets, and obtaining a caging path and formation. The last part will be described in next session in detail.

In the beginning of the first part, based on the robot dynamics, we generate a random path seed of a robot by incorporating a random acceleration under constraints of output limitations of robot's actuators. Additionally, this acceleration is set to be changed by a set of random acceleration values during the motion. Then, we will have a pseudo-random path set $\mathcal{P}_i$. Based on this path set, we can calculate a set of pseudo-random configurations as the reachable configuration at $t_c + \Delta t_c$ if the robot is in a free space. This is the procedure RandomPathSet(). Next by using the procedure FeasibleConfigurationSet(), we obtain the set of paths which are free from collision with the *C-obstacle* of the target object. The final configuration of each feasible path at the moment $t_c + \Delta t_c$ is recorded as an element of the feasible configuration set for caging formation testing. In this algorithm, testing collision free condition of a path is implemented by checking if a path goes in the C-obstacle of the moving object at all sample times[2].

After we have feasible configuration sets of the robot team, procedure CagingEdgesSet() and ClosedCagingFormation() are designed for obtaining candidates of caging formation. We implement CagingEdgesSet() to choose caging edge candidates, which satisfy Eq.8, from all combinations of feasible configurations between all neighbor set pair. Then, by checking the closed chain condition to all caging edge candidates (Fig.6), caging formation candidates can be obtained if exist and *Dynamic Object Closure* condition can be verified. By incorporating *Object Closure Margin*, it is easy to select a better caging formation which can cope

[1] in the motion planning sense

[2] Even in sampling based checking algorithm, enlarging the C-obstacle of the object with a safety margin can avoid checking error of feasibility of a path when the sampling time is relatively small.



(a) Caging Edge Candidate (b) Set of Caging Edge Candidates

Fig. 6. Set of caging formation candidates is obtained by checking connectivity of caging edges.

with relatively large uncertainty of motion estimation of the moving object.

By using random seeds to determine feasible configurations, we are not only easy to obtain collision-free reachable regions but also easy to incorporate some mature techniques to find feasible caging formations faster. Of course, as a probabilistic strategy, certain number of random seeds is necessary to find a feasible caging formation. However, because of high efficiency of caging formation testing algorithm based on *CC-Object* concept, the proposed *Random Caging Formation Testing* algorithm (RCFT) is able to run for real-time planning even with certain large number of random seeds. Additionally, how many random seeds for RCFP need here is depending on how large *Object Closure Margin* the caging formation has.

The strategy mentioned in the paper is a sufficient algorithm on Dynamic Object Caging Testing. It is because not only the condition for testing is a sufficient one, but also the random seed based path generation method is not optimized. By incorporating conventional motion planning methods of multiple robots [8], more precise testing could be achieved.

VI. Determination of Dynamic Caging Path and Formation

In general, multiple caging formations and paths to reach formations are obtained by procedure ClosedCagingFormation() by using the proposed *Random Caging Formation Path Determination* algorithm (RCFP). All these formations and paths are feasible candidates to achieve a *Dynamic Caging*, and we are interested in obtaining some paths and caging formations from those candidates with better performances. We can have different ways to consider performance of the path and caging formation candidates, such as, realizing a caging formation in shortest period, etc. In this research, for coping with error and uncertainty of sensing and control, we proposed two margin metrics, *Path Margin* and *Object Closure Margin*, on deciding the dynamic Caging path and formation.

A. Path Margin

Path Margin is an index of how far the path to achieve caging formation is from the moving object in the period t_c to $t_c + \Delta t_c$, the whole period on working toward the Dynamic Object Closure. To each feasible path i, *Path i Margin* of the particular path η_{path_i} is defined as the following, the minimum distance from the configuration of robot i to *C-Closure Object* of the object.

$$\eta_{path\_i} = \min_t(\boldsymbol{q}_{rbt\_i}(t) - C_{obj}(t)) \tag{6}$$

The *Path Margin* of a *Dynamic Caging Formation* is defined as the minimum values of *Path i Margin* of all paths ($\mathcal{CF}_{FM\_Path}$) which are need to move in other side of the target object without contacting with it on performing a caging formation.

$$\eta_{path} = \min_{i \in \mathcal{CF}_{FM\_Path}} (\eta_{path\_i}) \tag{7}$$

Large *Path Margin* will make the robots easy to reach the position for constructing a caging formation without replanning of robot 痴 path and caging formation both.

B. Object Closure Margin

For coping with error and uncertainty of sensing and control in the caging formation control of the robot team, *Object Closure Margin* is introduced to realizing a reliable caging formation control. When *Object Closure* is achieved, we define the closure margin to each pair robots as $\eta_{ij} = \rho_{min\_j} - \rho_{ij}$. The minimum of the closure margin of all pairs of neighbor robots in the system ($\eta_{cls} = \min_{i,j}(\eta_{ij})$) is defined as the *Object Closure Margin*.

In the implementation of caging formation control, the margin η_{cls} should satisfy the following condition basically.

$$\eta_{cls} - |e_{m\_ij}| - |e_{f\_ij}| > 0 \tag{8}$$

Here, e_{m_ij} is the maximum distance measuring error of ρ_{ij} and e_{f_ij} is the maximum error of the formation control of ρ_{ij}. Larger *Object Closure Margin* makes the system be able to satisfy *Object Closure* condition with relatively larger errors of formation control. The margin is easy to be calculated in the aforesaid testing algorithm, and the testing condition in the Proposition 3b will be replaced by this condition.

C. Determination of Caging Path and Formation

In this section, an algorithm is developed to decide a dynamic caging path and formation from its candidates by incorporating both margin metric *Path Margin* and *Object Closure Margin*.

Within the set of feasible path and caging formation on achieving a dynamic caging, there is one or a group of paths which *Path Margin* is the maximum value within the whole set of feasible paths, denoted as η_{path_max}. Also there is one or a group of caging formations which *Object Closure Margin* is the maximum value within the whole set of feasible caging formation, denoted as η_{cls_max}. A path and caging formation with both maximum margin values, η_{path_max} and η_{cls_max} will be the best choice on achieving a dynamic caging. However in general, a candidate is hard to have the maximum value on both margin metrics. In other word, a candidate which has maximum value on one metric will not be the maximum in another.

In this research, we developed an algorithm for obtaining a better solution in the view points of both *Path Margin* and *Object Closure Margin*. Here, we obtain the difference between the maximum value of a margin metric in the whole candidate set and the margin value of a path or caging formation. To each path candidate to reach a caging formation, the difference of its *Path Margin* from largest *Path Margin* ($\Delta\eta_{path}$) could be easily calculated by Eq.9. Similarly, to each caging formation, the difference of its *Object Closure Margin* from largest *Object Closure Margin* ($\Delta\eta_{cls}$) could be easily calculated by Eq. 10

$$\Delta\eta_{path} = \eta_{path\_min\_Max} - \eta_{path\_min} \tag{9}$$

$$\Delta\eta_{cls} = \eta_{cls\_Max} - \eta_{cls} \tag{10}$$

Here, $\eta_{path\_min\_Max}$ is the best value of the caging path margin η_{path_min}, and η_{path_min} is the best value of the caging formation margin η_{cls} in path and caging formation candidate set.

A good path and caging formation candidate will relatively large margins on both margin metrics. From this view point, the path and caging formation with the least square value of the two margin differences, $\Delta\eta_{path}$ and $\Delta\eta_{cls}$, is defined as the solution to perform a *Dynamic Caging*.

$$\mathcal{CF} = \min_{\mathcal{S}_{ce}}(\Delta\eta_{opt}^2) = \min_{\mathcal{S}_{ce}}((\alpha\Delta\eta_{path})^2 + (\beta\Delta\eta_{cls})^2) \tag{11}$$

Here, α and β are weight constants for two margin metrics.

VII. Formation-Path Planning Example

In this section, a planning example for realizing *Dynamic Caging* by using proposed *Random Caging Formation Path Determination* algorithm is presented. The initial conditions and results are shown in Fig.7 $\sim$ Fig.9 and Table I.

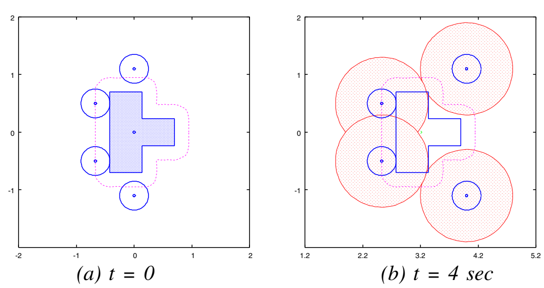


Fig. 7. (a) Initial object configuration and formation of the robots (b) Estimated configuration of the object and reachable configuration sets of robots at $t_c + \Delta t_c$.

We use a T-shaped object, the same object in Fig.1, as the caging target object. The mass, the coefficient of friction on the ground, and the object's initial velocity are $m_{obj} = 20kg$, $\mu = 0.01$, and $\dot{\boldsymbol{q}}_{obj}(0) = (1.0m/s, 0, 0)$ respectively. The system is in consist of four omni-directional mobile robots which construct a formation (*Case II* in Fig.2) to achieve a dynamic caging task. Predefined time interval for the *Dynamic Object Closure*, Δt_c, is set as $4sec$. Initial velocity of each robot is the same with the initial velocity of the object, $\dot{\boldsymbol{p}}_{rbt}(0) = (1m/s, 0)$, and the maximum acceleration of the robot is bounded by $\pm 0.1m/s^2$. At the time $t_c + \Delta t_c$, the reachable configuration region of each robot, R_{rbt} is a circular area with the radius of $0.8m$. The center positions of the regions of robot 1 and 4 are $(4.0m, 1.1m)$ and $(4.0, -1.1m)$. The center position of the region of two robots behind the object, robot 2 and 3, are $(2.78m, 0.5m)$ and $(2.78m, -0.5m)$ respectively (Fig.7).

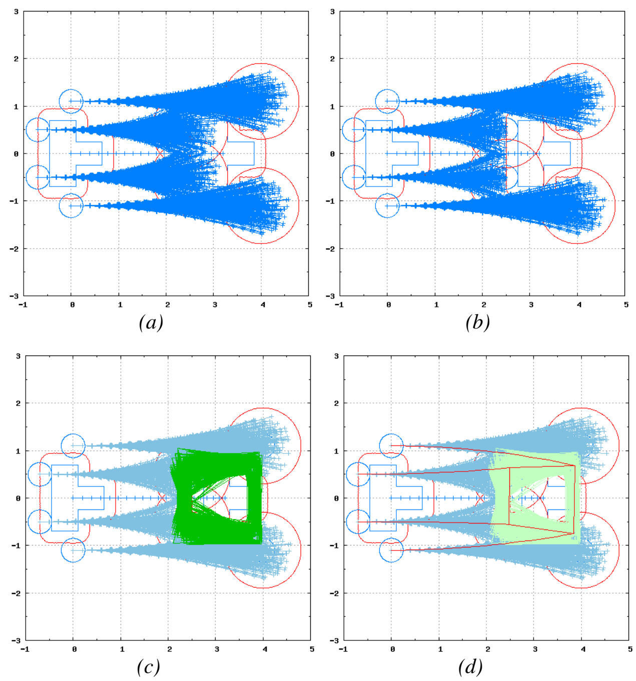


Fig. 8. Dynamic Caging Path Formation Determination: $\Delta t_c = 4sec$, 250 random seeds of paths to each robot's feasible configuration. $\rho_{cc\_min} = 1.5m$, $|e_{m\_ij}| + |e_{f\_ij}| \leq 0.15m$. (b) and (c) are reachable positions of robots and caging formation candidates. (d) is a caging path and formation candidate which has large *Path Margin* and *Object Closure Margin*

According to the geometric calculation of the *CC-object*, the maximum closure distance between two caging effectors, ρ_{cc_min}, is $1.5m$, and the error of measuring and formation control is set as $|e_{m\_ij}| + |e_{f\_ij}| \leq 0.15m$. Fig.8 shows the result of *Random Caging Formation Path Determination* algorithm with 250 random seeds of the path to each robot. Fig.8-(a) presents the distributed random seeds of robots' feasible configurations at $t_c + \Delta t_c$, and Fig.8-(b) shows path candidates without collision with the object. Fig.8-(c) shows formation candidates satisfying *Object Closure* conditions and Fig.8-(d) shows the resultant caging formation which has large *Object Closure Margin* and *Path Margin* comparing with other feasible caging formation candidates.

In Table I and Fig.9, results and performances of four

TABLE I

Performance of Caging Path and Formations Obtained by Using Different Decision Methods ($\alpha = \sqrt{3}, \beta = 1$)

	η_{path} (m)	ratio of $\Delta\eta_{path}$	η_{cls} (m)	ratio of $\Delta\eta_{cls}$	$\Delta\eta_{opt}^2$
Proposed Method	0.039	7.5%	0.117	0.3%	0.0046
Max η_{path}	0.055	0.0%	0.0003	99.8%	0.0178
Max η_{cls}	0.004	73.0%	0.133	0.0%	0.0425
Random Solution	0.001	11.2%	0.0002	99.8%	0.06421

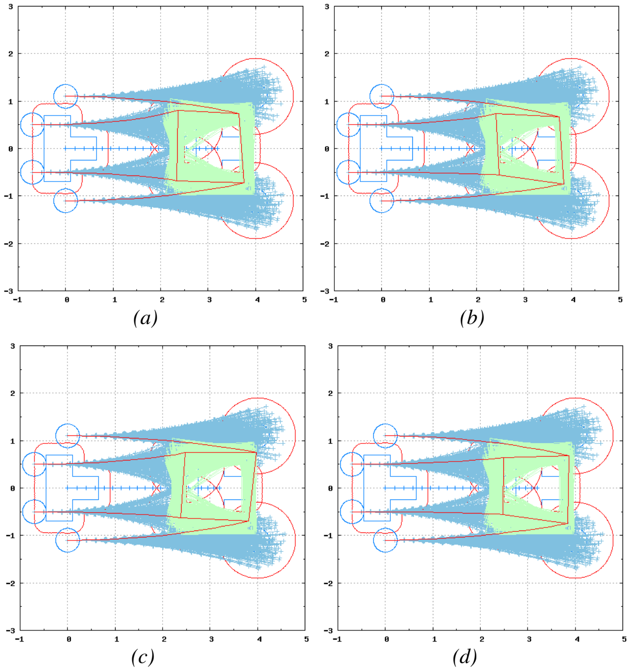


Fig. 9. Planning Results: Selected Caging Path and Formation by Different Strategies (a) Random Selection, (b) Maximum *Closure Margin* ($\eta_{cls} = 0.133m$, $\eta_{path} = 0.004m$) (c) Maximum *Path Margin* ($\eta_{cls} = 0.003m$, $\eta_{path} = 0.055m$) (d) Proposed *Random Caging Formation Path Determination* Algorithm ($\eta_{cls} = 0.117m$, $\eta_{path} = 0.039m$).

methods to obtain caging path formation are shown. Candidates with maximum value of either *Object Closure Margin*(Fig.9-(b)) or *Path Margin*(Fig.9-(c)) have poor performance on another margin metric. Fig.9-(d) shows the solution obtained by using proposed method and its performances are not the best but good on both margin metrics, within whole candidates of caging path and formation, being the top 7.5% on *Path Margin* and the top 0.3% on *Object Closure Margin*. Here, optimization weight constants are setting as $\alpha = \sqrt(3), \beta = 1$. Since the stability on maintaining the caging formation is important on object transportation and some collision avoidance control strategy can be incorporated when we implement the algorithm to robot systems, so the result obtained by the proposed method are feasible on applying to caging robot systems. It should be emphasized that α, β are weight factors on how much each metric should be near to the best value, rather than the margin value directly. Therefore, they just guarantee that a sub-optimal solution we will have, and are not critically affecting the magnitude of resultant margins of the solution.

VIII. Conclusion

In this paper, an approach to multi-robot manipulation: *Dynamic Object Closure*, a closure condition that guarantees to trap a moving object in a predefined future timing, is addressed. We proposed a probabilistic caging formation path determination algorithm (RCFP) for not only checking the sufficient condition of *Dynamic Object Closure* but also determine a sub-optimal path-formation candidate which realizing good performance on both *Path Margin* and *Object Closure Margin*. The proposed algorithm is efficient in calculating as a real-time planning and control method. Finally, some results of a planning example are presented for illustrating the proposed algorithm. Discussion on dynamic caging with uncertainty of object configuration estimation and development of an algorithm to obtain feasible robot configuration sets for dynamic caging are our future works.

References

[1] Brock O. and Khatib O., *Elastic strips: Real-time Path Modification for Mobile Manipulation*, Int.Symp.of Robotics Research,pp.117-122,1997.

[2] Fierro R., Das A.K., et al., *Hybrid Control of Formations of Robots*, ICRA2001, pp.157-162, 2001.

[3] Fink J., Hsieh M.A., and Kumar V., *Multi-Robot Manipulation via Caging in Environments with Obstacles*, ICRA2008, pp.1471-1476, 2008.

[4] Hsu D., Kindel R., Latombe J.C., and Rock S., *Randomized kinodynamic Motion Planning with Moving Obstacles*, the 4th Int. W. on Algorithmic Foundations of Robotics, 2000.

[5] Kosuge K., Hirata Y., et al., *Motion control of multiple autonomous mobile robots handling a large object in coordination*, ICRA99, pp.2666-2673, 1999.

[6] Kume Y., Hirata Y., Wang Z.D., and Kosuge K., *Decentralized Control of Multiple Mobile Manipulators Handling a Single Object in Coordination*, IROS2002, pp.2758-2763, 2002.

[7] Latombe J.-C., *Robot Motion Planning*, Kluwer Academic Publishers Press, 1991.

[8] Steven M. LaValle, Seth A.Hutchinson, *Optimal Motion Planning for Multiple Robots Having Independent Goals*, IEEE Trans. on Robotics and Automation, Vol.14, No.6, pp.912-925, 1998.

[9] Lynch K.M. and Mason M.T., *Stable Pushing: Mechanics, Controllability, and Planning*, Int. J. Robotics Research, 16(6), pp.533-556, 1996.

[10] Makita S., Maeda Y., *3D Multifingered Caging: Basic Formulation and Planning*, IROS2008, pp.2697-2702, 2008.

[11] Nakamura Y., Nagai K., and Yoshikawa T., *Dynamics and Stability in Coordination of Multiple Robotic Mechanisms*, Int. J. Robotics Research, Vol.8, No.2, pp.44-61, 1989.

[12] Ota J., *Rearrangment of Multiple Movable Objects*, ICRA04, pp.1962-1967, 2004.

[13] Pereira.G.A.S., Kumar V., et al., *Decentralized Algorithm for Multi-Robot Manipulation via Caging*, Int. J. Robotics Research, 23(7/8), pp.783-795. 2004.

[14] Reif J.H. and Sharir M., *Motion Planning in the Presence of Moving Obstacles*, IEEE Symp. on Fundations of Comp. Sci., pp.144-154, 1985.

[15] Rimon E. and Blake A., *Caging Planar Bodies by One-Parameter Two-Fingered Gripping Systems*, IEEE Trans. Robotics and Automation, 18(3), pp.299-318, 1999

[16] Stilman M. and Kuffner J., *Navigation among movable obstacles: Real-time reasoning in complex environments,* Humanoids04, 2004.

[17] Sudsang A., Rothganger F., and Ponce J., *Motion planning for disc-shaped robots pushing a polygonal object in the plane*, IEEE Tran. on Robotics and Automation, 18(4). pp.550-562, 2002.

[18] Sugar T. and Kumar V., *Multiple Cooperating Mobile Manipulators*, ICRA99, pp. 1538-1543, 1999.

[19] Wang Z.D., Nakano E. and Takahashi T., *Solving Function Distribution and Behavior Design Problem for Cooperative Object Handling by Multiple Mobile Robots* IEEE Tran. on Systems, Man, and Cybernetics, A, 33(5), pp.537-549, 2003

[20] Wang Z.D. and Kumar V., *Object Closure and Manipulation by Multiple Cooperating Mobile Robots*, ICRA2002, pp.394-399, 2002.

[21] Wang Z.D., Kumar V., Hirata Y., and Kosuge K., *A Strategy and a Fast Testing Algorithm for Object Caging by Multiple Cooperative Robots*, ICRA2003,pp.938-943, 2003.

[22] Wang Z.D., Hirata Y., and Kosuge K., *Control a Rigid Caging Formation for Cooperative Object Transportation by Multiple Mobile Robots*, ICRA2004,pp.1580-1585, 2004.

[23] Wang Z.D., Hirata Y., and Kosuge K., *An Algorithm for Testing Object Caging Condition by Multiple Mobile Robots*, IROS2005,pp.2664-2669, 2005.

[24] Wang Z.D., Hirata Y., and Kosuge K., *Dynamic Object Closure by Multiple Mobile Robots and Random Caging Formation Testing*, IROS2006,pp.3675-3681, 2006.