Dynamic Object Closure by Multiple Mobile Robots

Ozawa, Y. (Tohoku Univ.) Wang, Z. (Waseda, Japan) Kosuge, K. (Tohoku Univ.)

Abstract: In this paper, we discuss the manipulation of planar objects by multiple cooperating mobile robots using the concept of Object Closure. 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. A Random Caging Formation Testing algorithm (RCFT) is proposed for checking Dynamic Object Closure condition. Some simulation results are presented for illustrating the validity of the proposed algorithm.

1 Introduction

Cooperative control of multiple mobile robots system is usually inspired from human and animal behaviors. One of the most significant examples is cooperative object transportation which is mainly based on strategies of grasping or pushing based cooperative motion of human being. Recently a novel object handling strategy, caging based object handling strategy, has been discussed[2][3][4]. The similar behaviors are observed from cooperative hunting in some wolves, dolphins and humpback whales groups. 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.

Figure 1: Examples of Object Caging by Animals and Multiple Mobile Robots

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. 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 paper, we define the concept of Dynamic Object Closure for achieving object caging to a moving object. A Random Caging Formation Testing algorithm (RCFT) is proposed for checking Dynamic Object Caging condition. Simulation results are provided for illustrating the validity of the proposed algorithm.

2 Dynamic 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 \( A_i \), \( i = 1, \ldots, n \), denote the caging effector \( i \) in the working space. A configuration \( \mathbf{q} = (x, y, \theta) \) 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:

\[
C_{cls,i} = \{ \mathbf{q}_{obj} \in C \mid A_{obj}(\mathbf{q}_{obj}) \cap A_i(\mathbf{q}_i) \neq \emptyset \}
\]

(1)

\[
C_{cls} = \bigcup_{i=1}^{n} C_{cls,i}
\]

(2)

Let \( \mathbf{q}_{obj} \notin C_{cls} \) be a free initial configuration of the object. We define set \( C_{free} = C \setminus C_{cls} \) and define set \( C_{free, obj} \) as follows:

\[
C_{free, obj} = \{ \mathbf{q} \in C_{free} \mid connected(\mathbf{q}, \mathbf{q}_{obj}) \}
\]

(3)

We define \( \mathbf{q}_{inf} \in C_{free, inf} \) as a generic point that is sufficiently far away from the object. An object can escape from robots only when the \( \mathbf{q}_{obj} \) connects to the \( \mathbf{q}_{inf} \) in C-space. Then Object Closure can be defined as follow:

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

\[
\begin{aligned}
& C_{free, obj} \neq \emptyset, \{ \mathbf{q}_{obj} \\
& C_{free, obj} \cap C_{free, inf} = \emptyset
\end{aligned}
\]
Figure 3: 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.

When Object Closure is achieved, there is a bounded free space ($C_{\text{free, obj}}$) around the $q_{\text{obj}}$, which is entirely kept inside of the $C_{\text{cls}}$ (Fig.2-(b)). On the other hand, Object Closure is not satisfied when a connection path exists between $C_{\text{free, obj}}$ and $C_{\text{free, inf}}$.

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.3-(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.3-(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 $T \subset \mathbb{R}$ denote the time interval. Let the state space, $X$, be defined as $X = \mathcal{C} \times T$, denote the CT-space, consisting of configuration space and time space.

In $X$, we can represent a moving object as an obstacle region, $X_{\text{obj}}$, and robots moving around the target object as continuous and time-monotonic paths, $(X_{\text{rbt,i}}, i = 1, \cdots, n)$, around the obstacle region of the target object respectively. For a given $t \in T$, a slice of the obstacle region of the object and paths of robots, $X_{\text{obj}}(t)$ and $X_{\text{rbt,i}}(t)$, can be obtained as $C$-objects at particular time which is possible be used for testing Object Closure condition at this instant. Then we define Dynamic Object Closure as:

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

\[
\begin{align*}
C_{\text{free, obj}}(t, q_{\text{obj}}(t)) &\neq \emptyset, \{q_{\text{obj}}(t)\} \\
C_{\text{free, obj}}(t, q_{\text{rbt,i}}(t)) \cap C_{\text{free, inf}}(t, q_{\text{rbt,i}}(t)) &= \emptyset
\end{align*}
\]

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 constraints to the target object and other robots should be satisfied (Fig.4). These let the testing procedure of Object Closure after a predefined time intervals be complicated.

Figure 4: Feasible Configuration Region of Each Robot During Dynamic Object Caging.

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

- Motion of the target object within the predefined time interval $\Delta t_c$ can be estimated completely.
- All robots are disc-shaped and current velocity and the maximum acceleration of all robots are known. By using the first and second assumptions, each robot can calculate its region where it can reach after $\Delta t_c$ 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.

Of cause, 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. 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[6]. 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.

3 Basics of Object Closure Testing

In general, $C_{\text{cls}}$ and $C_{\text{free, obj}}$ 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. Following the definitions in Eq.3, we define their slices along $\theta = \theta_0$ to be $C_{\text{free, obj}}(\theta_0)$ and $C_{\text{free, inf}}(\theta_0)$. For reducing complexity of the testing procedure, a sufficient condition of Object Closure can be derived as follow.

**Proposition 3:** Let $\theta_0$ satisfy
\[
\begin{align*}
& C_{\text{free, obj}}(\theta_0) \neq \emptyset, \{q_{\text{obj}}\} \\
& C_{\text{free, obj}}(\theta_0) \cap C_{\text{free, inf}}(\theta_0) = \emptyset.
\end{align*}
\]
A sufficient condition of the Object Closure is that for all $\theta \in [0, 2\pi)$,
\[
\begin{align*}
& C_{\text{cls}, i}(\theta) \cap C_{\text{cls}, i+1}(\theta) \neq \emptyset \quad i = 1, \ldots, n - 1 \\
& C_{\text{cls}, n}(\theta) \cap C_{\text{cls}, 1}(\theta) \neq \emptyset.
\end{align*}
\]

Since the evaluation for Object Closure must run in real time, the computation cost should be low. But calculations involved in computing $C_{\text{cls}, i}$ and checking the condition in Proposition 3 directly are still hard because of the geometrical complexity of $C_{\text{cls}}$. To solve this problem, we proposed a concept of CC-Closure Object in [4] and [6] which is useful on building an efficient testing algorithm even if the shape of the object is complicated.

We defined a new C-space for the C-Closure Object $C_{\text{cls}, i}$ and denote it as CC, C-Object of a C-Closure Object in CC (here, $i \neq j$) is called CC-Closure Object:
\[
C_{\text{cls}, ij} = \{q_j \in CC \mid C_{\text{cls}, i}(q_i) \cap C_{\text{cls}, j}(q_j) \neq \emptyset\},
\]
which indicates the C-Obstacle of $i$th C-Closure Object to $j$th 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.
\[
C_{\text{cls}, i}(\theta) \cap C_{\text{cls}, j}(\theta) \neq \emptyset \iff q_j \in C_{\text{cls}, ij}(\theta)
\]

Because the size and the shape of $C_{\text{cls}, i}$ and $C_{\text{cls}, ij}$ 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.

Let's represent the region of the sliced CC-Closure Object at a fixed orientation into the $\rho$-$\theta$ coordinates (spherical coordinates). In the $\rho$-$\theta$ coordinates, a line connecting the $i$th robot to $j$th robot is a vertical line segment. If the distance between the robot $i$ and $j$ is smaller than $\partial CC_{\text{cls}, ij}(\theta)$, $p_i$ is in the $CC_{\text{cls}, ij}(\theta)$ which indicates that $C_{\text{cls}, i}(\theta)$ is in contact or overlapping with the $C_{\text{cls}, j}(\theta)$.

### 4 Test Algorithm of Dynamic 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 compact set of configurations, rather than a single one. 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).

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. Based on the Proposition 2 and 3b, Random Caging Formation Testing (RCFT) algorithm is designed as follow:

1: procedure RandomCagingFormationTest(t):
2: begin
3: Obtain $p_{\text{obj}}(t)$;
4: For All Robots ($i = 1, \ldots, n$)
5: $p_i = \text{RandomPathSet}(t)$;
6: $R_i = \text{FeasibleConfigurationSet}(p_i, t, p_{\text{obj}}(t))$;
7: $S_{\text{c}} = \text{CagingEdgesSet}(R_1, \ldots, R_n, \text{pos, min})$;
8: $C.F = \text{ClosedCagingFormation}(S_{\text{c}})$;
9: If ($C.F \neq \text{NULL}$) return True;
10: else return False;
11: end;

This algorithm consists of two main parts; obtaining the feasible configuration set of each robot at a future time $t_c$ and checking existence of a closed caging formation from all robots' feasible configuration sets.

With 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, 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 ver-
fied. Additionally by incorporating Object Closure Margin, it is easy to useto select a better caging formation which can cope with relatively large uncertainty of motion estimation of the moving object.

5 Simulation Examples

In this research, a Java based simulator system is developed and some simulations have been done for illustrating the validity of the proposed Random Caging Formation Testing algorithm. The simulation condition and results are shown in Fig.7 ~ Fig.8.

![Figure 7](image1)

**Figure 7:** Simulation: (a) Initial object configuration and formation of the robots (b) Estimated configuration of the object and reachable configuration sets of robots.

A T-shaped object, the same object in Fig.1, is used as the target. The mass, the coefficient of friction on the ground, and the object’s initial velocity are \( m_{\text{obj}} = 20kg \), \( \mu = 0.01 \), and \( \dot{\theta}_{\text{obj}}(0) = (1.0 m/s, 0, 0) \) respectively. Four omni-directional mobile robots construct a formation (Fig.3-(b)) to achieve a dynamic caging task. \( \Delta t_c \), predefined time interval for the Dynamic Object Closure testing, is set as 4sec. In the strategy proposed in this research, all robots are controlled without contacting the object before a caging formation is achieved. Then, only friction force is applied on the object, and \( \vec{p}_{\text{obj}}(t) = (-0.1m/s^2, 0) \) and \( \vec{p}_{\text{obj}}(t_c + \Delta t_c) = (3.2m, 0) \). To each robot, velocity is the same with the object at the initial moment, \( \vec{p}_{\text{obj}}(0) = (1m/s^2, 0) \). 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, \( R_{\text{obj}} \) 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). In this simulation, \( \rho_{cc,\text{min}} \), the maximum closure distance between two caging effectors, is 1.5mm, and the error of measuring and formation control is set as \( \left| e_{c,\text{obj}} \right| + \left| e_{f,\text{obj}} \right| \leq 0.15m \).

In Fig.8, random seeds of feasible configurations of each robot and caging formation candidates are shown for examples with various numbers of random seeds. In our simulation, average number of feasible paths without collision with the moving object, success rate and average number of caging formation candidates found by the Random Caging Formation Testing algorithm with various numbers of random seeds of the paths are shown. For each case, average of data is calculated from 500 trials of simulation with exactly same conditions but different random seeds. High success rate can be realized when the number of random seeds is more than 200. We can have reliable result of caging formation testing when the random seeds number is more than 350. The time cost of procedure RandomCagingFormaionTest() with 350 random seeds of each robot’s path is 65msec ~ 105msec on a PC-AT Linux machine (P4, 2.6GHz). The results show that the proposed algorithm has demonstrated good ability on testing Dynamic Caging Formation based on a sufficient condition of Dynamic Object Closure in real-time.

6 Conclusion

In this paper, we proposed and defined a novel approach to multi-robot manipulation: Dynamic Object Closure, a closure condition that guarantee to trap a moving object in a predefined future time. We presented a probabilistic caging formation testing algorithm for checking the sufficient condition of Dynamic Object Closure. The proposed algorithm is efficient in calculating as a real-time planning and control method. This is contributed by the efficiency of the original Object Closure testing produces and the efficiency of calculation of the Object Closure Margin which are incorporating with the concept of CC-Closure Object. Finally, some simulation results are presented for illustrating the proposed algorithm.

References


1A1-B17(4)