A "Constrain and Move" Approach to Distributed Object Manipulation

Majid Nili Ahmadabadi and Eiji Nakano

***Abstract*—Studying the system dynamics, this research is an attempt to understand and design the basic robot behavior, information system, and distributed cooperation strategies required by a group of cooperative behavior-based mobile robots for handling an object. A new concept to develop distributed cooperation strategies to carry a load is introduced. In this method, the task of transferring the object is divided into two independent executable subtasks: constraining and moving the load. Each subtask is assigned to a group of distributed mobile robots. Based on this idea (*constrain-move* concept), two distributed cooperation strategies to turn the object about a fixed point and to move it along a straight line are introduced. The *constrain-move* concept is generalized for lifting and lowering the object. It is noted that weight of the payload can be used by the robots as a natural constraint on the object. Utilizing this natural constraint, a distributed cooperation strategy and an information system to lift and lower the object are introduced. The coordination protocols are devised in such a way that the robots can control movements of the object using their own sensory information and some static data communicated between the team members. Simulation and experimental results are given to support the proposed approach.**

***Index Terms*—Distributed robots, mobile robot, multirobots, object manipulation.**

I. Introduction

THERE has been much research concerning multiple cooperative mobile robots [5], robotic hands, and multiple manipulators [14], [20], [21], [36], [42].

The key driving forces for many researchers to study cooperative object manipulating robots are the industrial needs and the desire to understand human abilities and its intelligent cooperative behavior (e.g., [13]). Sophistication of the task from kinematics, dynamics, and control points of view must be added to these factors. In addition, implementation of the cooperative robots is regarded as a new approach to construct flexible and robust robot systems [2].

Centralized model-based methods are widely studied to control these systems, e.g., [20] and [36]. Another approach is to assign one of the team members as the team leader and make the rest of the group follow it, e.g., [16], [17], and [35]. In contrast to these two architectures, a team of distributed robots[1] does not completely depend on a central unit or one of its members. Having this characteristic, the system could be made more fault tolerant when compared with centralized and leader–follower teams.

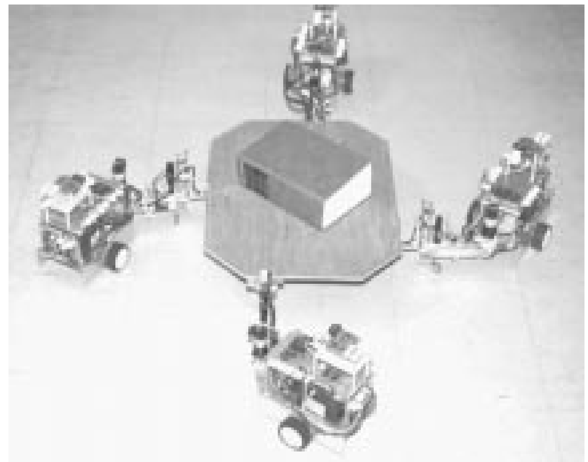


Fig. 1. Ichiro, Jiro, Saburo, and Shiro lift and carry an object toward the goal.

Robot actions and the information required to execute the coordination protocol vary with complexity of the cooperation strategy. The coordination protocol can be simplified if a proper task allocation scheme is chosen and a suitable robot mechanism is designed [23]. Also, the system must be developed in such a way that the robots are less dependent on the complicated sensors and explicit communication [8], [18].

In this research, we are interested in studying a distributed group of nonholonomic mobile robots handling an object from its initial position to the goal (see Fig. 1). The robots suspend the object on its bottom face. In their way, the robots shall not turn the object over and must follow a specified path.

In our system, each robot has simple mechanism, sensory system, and behaviors. But as a team, the robots execute the complicated task.

In this paper, learning from mechanical systems, the *Constrain-Move* strategy for object manipulation is introduced from task distribution and task decoupling points of view. Using *Constrain-Move* strategy, simple and practical distributed cooperation strategies for dynamic reorienting and moving the object along a defined path are given. Then, using weight of the object,

Manuscript received June 15, 1999; revised October 12, 2000. This paper was recommended for publication by Associate Editor A. Bicchi and Editor V. Lumelsky upon evaluation of the reviewers' comments. This paper was presented in part at the 1997 IEEE International Conference on Robotics and Automation (ICRA)m Albuquerque, NM, the 1996 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Osaka, Japan, 1996, and the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Grenoble, France, 1997.

M. N. Ahmadabadi is with the Department of Electrical and Computer Engineering, Faculty of Engineering, University of Tehran, Tehran, Iran (e-mail: nily@sofe.ece.ut.ac.ir).

E. Nakano is with the Advanced Robotics Laboratory, Graduate School of Information Science, Tohoku University, Sendai 980-8579, Japan (e-mail: nakano@robotics.is.tohoku.ac.jp).

Publisher Item Identifier S 1042-296X(01)04806-6.

[1]Architecture of a team of robots is called distributed if all of the robot agents are equal with respect to control [5].

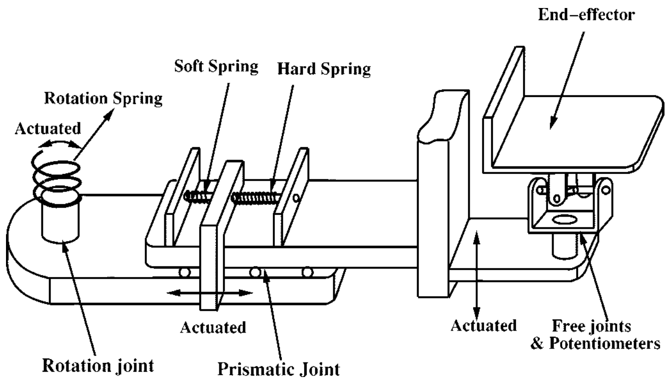


Fig. 2. Schematic view of the robot arm.

the proposed idea is generalized for constructing distributed object lifting protocols. In all strategies, the task mechanism and the robot hardware are exploited as the resources to simplify the system. Simulation and experimental results are also given to support the proposed ideas and to show the necessity of having compliant elements in the robot arms for executing these cooperation strategies.

The experimental system is introduced in the next section. A mathematical model of the object manipulation task with multiple robots is given in the third section. After that, related researches are reviewed and an overview of our approach is given in the fifth section. Based on the proposed strategy, two distributed strategies for object turning and moving tasks are developed in Sections VI and VII. Sections VIII and Section IX are devoted to extend our strategy for object lifting/lowering tasks. Simulation and experimental results are reported in Sections X and XI. A conclusion of this paper is given in the last section.

II. Experimental System

A team composed of four robots (Ichiro, Jiro, Saburo, and Shiro) called Hamcar[2] is developed (Fig. 1). Each robot, weighing 2.7 kg each (battery included), has a T801 Transputer-based computer system and a broadcasting communication unit. The mechanical part consists of a nonholonomic mobile base, an arm, a lifting mechanism, and a compact end-effector.

The mobile part, 24 cm in length and 13 cm in width, has two driving-steering wheels and a caster. Orientation of each robot is measured by a combination of a compass and a gyroscope. Dead-reckoning method is used to calculate the position of the mobile base. The arm is connected to the mobile base through a rotational spring and an actuated joint.

To have enough degrees of freedom for this nonholonimic system, each robot arm is connected to the lifting mechanism through a prismatic joint (Fig. 2). The prismatic joint system is composed of two sliders and two springs. The prismatic joint is driven by a lead screw. The compliant element in the prismatic joint has a higher stiffness value in the pushing direction than it does in the pulling direction. This simplifies control of the robot-arm interaction forces. The arm length is 16 cm when it is fully folded and it can be extended up to 22 cm. Deflection in the compliant elements are measured by rotational potentiometers.

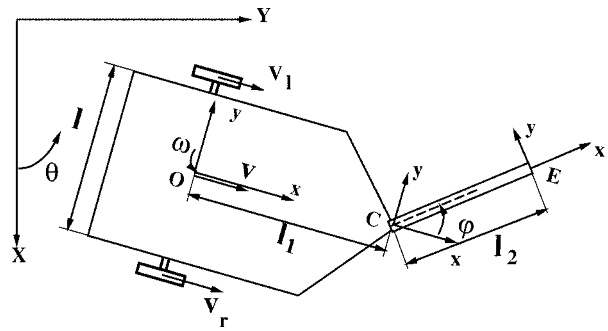


Fig. 3. Kinematic model of the mobile base and the arm.

The end-effector is shaped like a 90° angle (Fig. 2). The bottom face of the end-effector goes beneath the object and its vertical face is against one of the object sides. The end-effector is connected to the lifting bolt through three free joints, those together are equivalent to a ball joint. These joint angles are measured by three potentiometers.

A. Kinematic Model of the Robot

The robots are controlled kinematically and it is assumed that the static and quasi-static forces dominate the mobile robot dynamics. Also, the arm is low weight and the object dynamics are dominant.

Fig. 3 shows a kinematic model of the mobile base and the robot arm. The prismatic joint in the arm is not modeled here, because it is used just in some special cases. A coordinate system is attached to the mobile base at the center point of its axle (O). The robot velocity ($\mathbf{v}$) is along the $\mathbf{x}$ axes of the vehicle coordinate system.[3] $\omega = \dot{\theta}$ denotes the angular speed of the mobile base. v_r and v_l are translational speed of the right and the left wheels, respectively. φ is the arm angle in the mobile coordinate system.

For the mobile base we have

$$v = \frac{v_r + v_l}{2} \tag{1}$$

and

$$\omega = \frac{v_r - v_l}{l}. \tag{2}$$

Velocity of point E in the world coordinate system is

$$\mathbf{v}_{\boldsymbol{E}} = \mathbf{v} + \boldsymbol{\omega} \times (\mathbf{l_1} + \mathbf{l_2}) + \dot{\boldsymbol{\varphi}} \times \mathbf{l_2}. \tag{3}$$

By substituting values of $\mathbf{v}$ and $\boldsymbol{\omega}$ in (3) by (1) and (2) we have

$$\mathbf{v}_{E_x} = \frac{v_r + v_l}{2}\cos(\theta) - \frac{v_r - v_l}{l} l_1 \sin(\theta) - \left(\frac{v_r - v_l}{l} + \dot{\varphi}\right) l_2 \sin(\theta + \varphi) \tag{4}$$

[2]Hamcar is a Persian word that means cooperative. The team member names mean first, second, third, and fourth son in Japanese, respectively.

[3]Boldface parameters refer to vectors or matrices and ordinary characters are magnitude of the vectors or other scaler parameters.

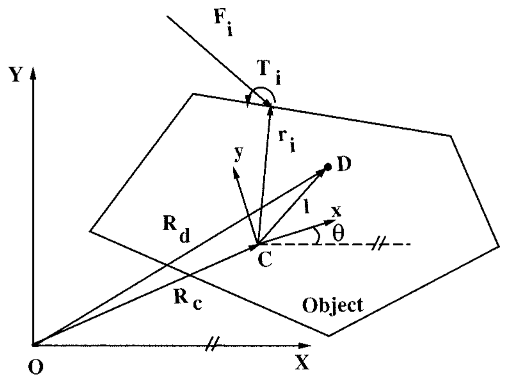


Fig. 4. A simple model of an object being rotated in a plane around point D by a group of robots.

and

$$\mathbf{v}_{E_y} = \frac{v_r + v_l}{2}\sin(\theta) + \frac{v_r - v_l}{l} l_1 \cos(\theta) + \left(\frac{v_r - v_l}{l} + \dot{\varphi}\right) l_2 \cos(\theta + \varphi). \quad (5)$$

In the matrix from we have

$$\begin{bmatrix} \mathbf{v}_{E_x} \\ \mathbf{v}_{E_y} \end{bmatrix} = \mathbf{J}(\theta, \varphi) \begin{bmatrix} v_r \\ v_l \\ \dot{\varphi} \end{bmatrix} \quad (6)$$

where $\mathbf{J}(\theta, \varphi)_{2\times 3}$ is the Jacobian matrix.

There are some situations where the robot is required to push the bulky object in the arm direction (along the $\mathbf{x}$ axis in the effector coordinate system) or change the arm-object angle without pushing or pulling the object. Mathematically, for the mobile-arm joint (C) in the arm coordinate system ($\mathbf{E}$) we have

$$\mathbf{v}_{C_x} = v\cos(\varphi) + l_1 \omega \sin(\varphi) \quad (7)$$

and

$$\mathbf{v}_{C_y} = -v\sin(\varphi) + l_1 \omega \cos(\varphi) \quad (8)$$

and

$$\begin{bmatrix} \mathbf{v}_{C_x} \\ \mathbf{v}_{C_y} \end{bmatrix} = \begin{bmatrix} Q \\ R \end{bmatrix} \quad (9)$$

where Q and R are two real and independent values corresponding to the desired pushing force and the required angular velocity of the arm around point E, respectively.

III. Mathematical Description

In the next section, we will compare the basic ideas in different object manipulation schemes from the task allocation and the coordination points of view. Therefore, a simple mathematical model of the system is developed in this paper.

Assume we wish to turn a two-dimensional object[4] (Fig. 4) around point D. The object is of mass M and I is its moment of inertia at its center of gravity C about the $\mathbf{Z}$ axis with unit vector $\mathbf{k}$. In Fig. 4, the robot-object interaction forces and torques are denoted by $\mathbf{F}_i$ and $\mathbf{T}_i$: $i = 1, 2, \cdots, n$, where n is the number of the object manipulating robots. $\mathbf{r}_i$ denotes the position of the ith robot-object contact point in the object coordinate system.

[4]We consider a two-dimensional system for simplicity of the discussions. Similar equations can be developed for a three-dimensional case.

The desired error function of the object configuration is defined as

$$\ddot{e}_r + C_r \dot{e}_r + K_r e_r = 0 \quad (10)$$

where $e_r = \theta_{\text{desired}} - \theta$ and C_r and K_r are the desired rotational damping and spring coefficients of the object, respectively. Then, to make the object angle follow its path, the net torque on the object must be[5]

$$\sum_{i=0}^{n} (\mathbf{r}_i \times \mathbf{F}_i + \mathbf{T}_i) = I(C_r \dot{e}_r + K_r e_r + \ddot{\theta}_{\text{desired}})\mathbf{k}. \quad (11)$$

We call this relation, (11), the *move* equation.

Let $\mathbf{a}_c$ and $\mathbf{a}_d$ represent the acceleration of the object center of gravity and of point D, respectively. Assume $\boldsymbol{\omega} = \dot{\boldsymbol{\theta}}$ is the angular velocity of the object and $\boldsymbol{\alpha} = \dot{\boldsymbol{\omega}}$. Then we have

$$\mathbf{a}_d = \mathbf{a}_c + \boldsymbol{\omega} \times \boldsymbol{\omega} \times \mathbf{l} + \boldsymbol{\alpha} \times \mathbf{l}. \quad (12)$$

The object can be turned around point D if this point is kept stationary at its initial position $\mathbf{R}_{d0}$.[6] Assume the error function in position of point D is

$$\ddot{\mathbf{e}}_d + \mathbf{C_p}\dot{\mathbf{e}}_d + \mathbf{K_p}\mathbf{e}_d = \mathbf{0} \quad (13)$$

where $\mathbf{e}_d = [\mathbf{R}_{d0_x} - \mathbf{R}_{d_x} \quad \mathbf{R}_{d0_y} - \mathbf{R}_{d_y}]^T$ and $\mathbf{C_p}$ and $\mathbf{K_p}$ are two 2×2 diagonal matrices of the desired damping and the stiffness coefficients of the object around the initial position of D; $\mathbf{R}_{d0}$. Equation (12) shows that there is a nonlinear coupling between the rotational velocity of the object ($\boldsymbol{\omega}$) and $\mathbf{a}_d$ if the rotation center and the mass center of the object are not identical. Therefore, to realize the path described by error function in (13), the effect of the object rotation on acceleration of point D must be canceled out. In other words

$$\sum_{i=0}^{n} \mathbf{F}_i = M(\mathbf{Q} + \mathbf{C_p}\dot{\mathbf{e}}_d + \mathbf{K_p}\mathbf{e}_d) \quad (14)$$

where $\mathbf{Q} = [-\boldsymbol{\omega} \times \boldsymbol{\omega} \times \mathbf{l} - \boldsymbol{\alpha} \times \mathbf{l}]^T$.

We refer to this equation as the *constrain* equation.

After changing the desired path of point D to a straight line and fixing the goal configuration of the object to its initial value, the above formulation also describes the dynamics of the system when moving the object on a straight line.

IV. Related Works

Cooperation strategies for object manipulation mainly differ in the level of distributability, dependency on the model of the overall system, nature of the system (being dynamic, quasi-static, or static), utilization of the system resources, the accuracy of object path following, and the volume of data to be communicated by the robots. Therefore, the existing strategies are discussed from these points of view.

A. Centralized Systems

The main concept of the centralized strategies is that the central controller calculates the desired wrenches on the object

[5]The equations of translational and rotational motion of the object can be unified in a compact matrix form [20], however, here we keep them separated for further discussions.

[6]This is equivalent to say that point C must move on a circle of radius l centered at D.

[(11) and (14)] and the required internal forces to stabilize the robot-object contacts, e.g., [14], [15], [20], [36], and [42].

In most of the centralized systems, all of the robots are engaged in both accomplishing object moving (11) and object constraining subtasks (14), e.g., [20]. Therefore, theoretically, they have a high efficiency in accomplishing the task, as the system can be optimized when operating.

To simplify the calculations in some systems, task of a group of robots is constraining the object and the rest of them turn the load, e.g., [6], [32], and [33]. In these approaches, even though the task of the robots are different, yet the central unit must calculate the robot-object wrenches.

In some of the centralized systems, each robot arm has its own controller and executes a local feedback loop. In a group of these works, some parts of the central unit tasks are transferred to the robot arm controllers, e.g., [15].

Regardless of the local controller tasks, all of the above-mentioned systems depend heavily on the central unit. Therefore, these systems are very vulnerable to failure of the central unit. In addition, a communication medium with high bandwidth is required in such systems.

B. Leader–Follower Approaches

Some researchers have proposed some leader–follower schemes to make decentralized cooperation strategies, e.g., [16], [17], [21], and [35]. In these approaches, the collaboration and the coordination strategies are simplified.

Leader–follower systems are based on the idea that the leader moves its effector-object contact point on the defined path and the other robots follow the object while keeping their effector-object contact forces in a defined ranges. In other words, the leader must constrain (11) and move (14) the object by itself.

In a leader–follower system, the possible wrenches to be applied to the object is restricted by the nature of the leader-object contact (i.e., joint, friction, or frictionless contact) and the maximum wrench that the leader can produce. Also, abilities of the follower robots are not utilized in the directions the leader is active. Also, to have a reliable system, we must have more than a would-be leader in the system which means a lower efficiency and a higher cost.

C. Some Other Schemes

The system presented in [11] and [12] is based on a combination of distributed and centralized architectures. In this system, the robots suspend a payload and carry it along a desired path. The object path planning and force distribution are done by one of the robots and some of the staff robots solve local kinematic and dynamic path planning problems.

In [1], a semi-autonomous system to perform cooperative object handling tasks by a small team of human and robotic agents is introduced. The system architecture and the role of each agent are the main topics of this paper.

To move an object on a desired path, Brown and Jennings [4] divided the task into pushing and steering jobs and allocated each subtask to one mobile robot. Then, each robot executes its job while being compliant to the object movements. The system is quasi-static and there is no strict control on the object path. In addition, the robots must let the object slides on them, which is not practical for most of object manipulating systems, especially when the dynamic forces are present.

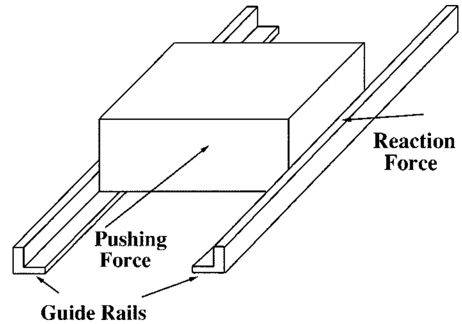


Fig. 5. The guide rail constrains the object in all directions except along the desired path.

Mataric [19] proposed a synchronized turn-taking approach to show the effects of using direct communication in sharing tasks between two box pushing robots. Each robot also acts as a remote sensor for its teammate.

In ALLIANCE architecture [29], [30], a message broadcasting mechanism is applied to enable each robot to take a proper action at any moment.

In light of information invariants [8], [9], Donald and his colleagues [7] designed some quasi-static object manipulation protocols to control the object angle and its position in one direction.

Wang *et al.* [38], [39] developed a cooperative behavior-based robot system for object manipulation (BeRoSH). There is no control on the object path.

Rus and her colleagues [32] proposed a quasi-static cooperation strategy to turn an object. In their method, one member of the team pushes the object against the stationary robots. The resulting torque makes the object to slide on the fixed robots and to rotate.

In their system, the object must slide on the robots, and turning the object around a defined point or moving it along a straight line is not possible.

V. An Overview of the New Approach: A Lesson from Mechanical Systems

When designing a mechanical system, the first thing to consider is to arrange the bearings and fixtures to constrain the parts properly. In other words, the parts are free in the directions they are supposed to move and are fixed in the others. Having a properly arranged constraining parts, an actuator can move the system in the designed way. This means, the actuator is not concerned with constraining undesirable movements of the moving parts. Moreover, since the bearings can bear some undesirable wrenches, small errors in directions of the actuator forces are durable.

For example, when moving an object on a guide rail (Fig. 5), the object-rail interaction forces constrain the object so that it can only move along the rails.

In such situations, the main job is to control the object along its desired path in such a way that the interaction forces do not exceed their limits or a jamming does not occur.

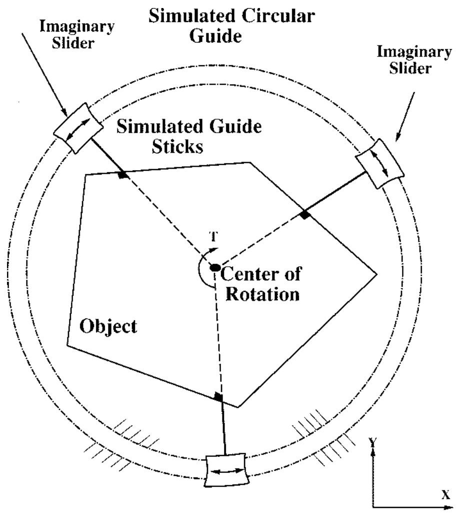


Fig. 6. To turn the object, the robots constrain it so that the desired rotation center does not move. It is like fixing some simulated (virtual) sticks to the object and jointing the sticks to some sliders moving inside a virtual guide.

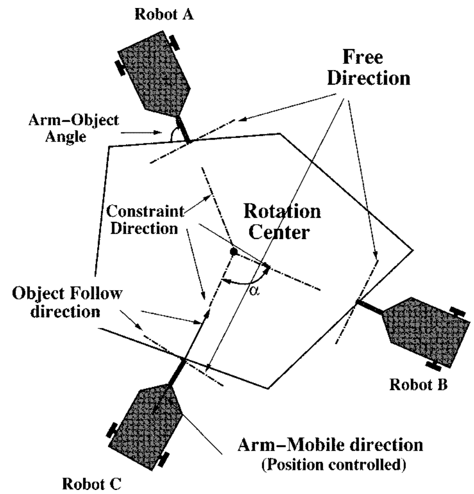


Fig. 7. Three robots (A, B, and C) with arm angle $\alpha < \pi$ grasp the object through friction contacts and make a force direction closure on it. Each mobile base is controlled such that the robot-object contact point does not move in the arm direction toward the mobile base. Robots follow the object in the other directions.

The main concept of the proposed distributed cooperation strategy (*Constrain-Move strategy*) is to simulate the physical rules governing the systems mentioned above. In other words, the object manipulation task is divided into two independent subtasks. A group of the robots constrain the object along the constraint directions (14) and another team of them move the object in the desired direction (11).[7] Doing so, the coordination protocol can be simplified.

As an example, to turn the object around a point in a plane (Fig. 6), the robots must constrain the object in $\mathbf{X}$ and $\mathbf{Y}$ directions and produce an appropriate torque around the object rotation axis—along its free direction. In other words, the robot-object interaction forces in the constraint directions shall make a force direction closure [22] on the object.

When moving the object on a straight line, the constraint-making robots must make a torque closure on the object and immobilize it in the direction perpendicular to its desired path.

In this paper, it is assumed that the object is initially grasped by the robots. Each constraint-making robot measures the undesirable movements of the object and compensates them. These robots are compliant to the desired motions of the object. The task of the other team of robots is pushing the object to move it.

VI. A Control Strategy for Distributed Reorientation

As in Fig. 6, the constraint on the object must be such that the dynamic forces of the object (14) can be canceled out by the constraint-making robots. Therefore, the constraint forces must span a force plane for a two-dimensional case.

There are different ways to place such constraints on an object. Among them, we are interested in those requiring minimum explicit coordination between the robot movements. In most of real world applications, the robots can only grasp the payload through a set of friction contacts. To have a stable friction contact, the robot-object interaction force must always remain inside the friction cone.

In our robot system (Hamcar, Fig. 1), the object rests on the end-effectors. Therefore, each robot may pull the object toward itself as far as the contact is stable. But, in order to avoid resorting to sophisticated control systems, the robots are instructed not to pull the object.

In Hamcar, the planer robot-object contact force, perpendicular to the robot arm, is the result of the mobile-arm joint motor torque. Therefore, if this joint is controlled as a free joint (zero torque joint), the robot-object interaction force will be along the robot arm. Therefore, main behaviors to stabilize robot-object contact are as follows.

- *Do not pull the object == Zero stretch in arm spring by controlling the mobile-arm joint movement along the arm [first raw in (9)*].
- *Zero torque at mobile-arm joint == Free joint*.
- *Keep the arm in the friction cone == Control arm-object angle by controlling the mobile-arm joint movement perpendicular to the arm [second raw in (9)]*.

Hence, at least three robots are needed to immobilize the object center of rotation.[8]

To make a force direction closure, as shown in Fig. 7, the robot arm directions must intersect at the desired rotation center. Also, the angle between each pair of three arm directions (α) should be smaller than 180°.

To constrain the object passively, the constraint-making robots are position controlled in their robot-side arm directions (called position controlled direction in Fig. 7). To keep the constraint on the object when rotating it, the robots control their arm-object angle too. The main behaviors to make such constraint on the object are as follows.

- *Keep robot-object angle at θ_0 == control mobile-arm joint movements perpendicular to the arm [first raw of (9)]*.

[7]These directions henceforth are referred to as constraint and free directions, respectively.

[8]If the robot-object contact is naturally stable, i.e., joint contact, two robots would be sufficient [26]

- *Do not move mobile-arm joint in the arm direction == resist mobile-arm joint movements in arm-mobile direction [Fig. 7 and second row of (9)].*
- *Follow the object in the arm perpendicular direction (equivalent to keep robot-object angle behavior).*

Additional robots are required to turn the object. Like for constraint-making robots, these robots can only push the object along their arm directions. In [26] and [28], it is discussed that, if we use a robot(s) to turn the object, the position of the object turning robot(s) must be decided so that it (they) can turn the object in both directions.

To make the initial system arrangement,[9] some geometrical information on the bottom face of the object is required, and we do not have to have an accurate model of the object. Knowing position of the object center of gravity, the system arrangement can be modified in a way that some dynamic load distribution indexes are improved.[10]

Some equivalent sets of simple and basic information systems for the robots are given in [23] and [28].

VII. A Control Strategy for Distributed Object Carrying Task

In *constrain and move* strategy, the constraint-making robots must resist all movements of the object that are not along the desired path. Also, they must constrain the angular movements of the object (see Fig. 8).

A. Constraining Strategy

Considering the contact stability, at least three members of Hamcar must cooperate to constrain the object (see Fig. 8). Two additional robots are needed to move the object along the path.

The constraint-making robots are arranged in such a way that their arms are perpendicular to the desired path of the object, means controlling arm-object angle at a designed value (β).[11] Each robot moves its mobile base so that its contact point does not move in its arm direction. These two behaviors of the robots are equivalent to move the mobile-arm joint on a line parallel to the designed path, passing through the desired position of the mobile-arm joint, and control the arm-object angle simultaneously[12] (see Fig. 8). Therefore, the main behaviors to constrain the object are as follows.

- *Keep robot object angle at β degrees.*
- *Do not pull the object.*
- *Return the robot-object contact point on its defined path.*

Realization of these behaviors are similar to those explained in Section VI.

A strategy to reduce number of the object moving robots to four is introduced in [28].

[9]Supervisor-based, negotiation-based, and leader–follower systems are some methods that can be applied to find a proper initial configuration of the robots.

[10]This is one of subjects to be studied.

[11]Angle β is decided on the basis of geometry of the object and the direction of the desired path.

[12]In addition to the information each object turning robot needs, the robots must know the desired path.

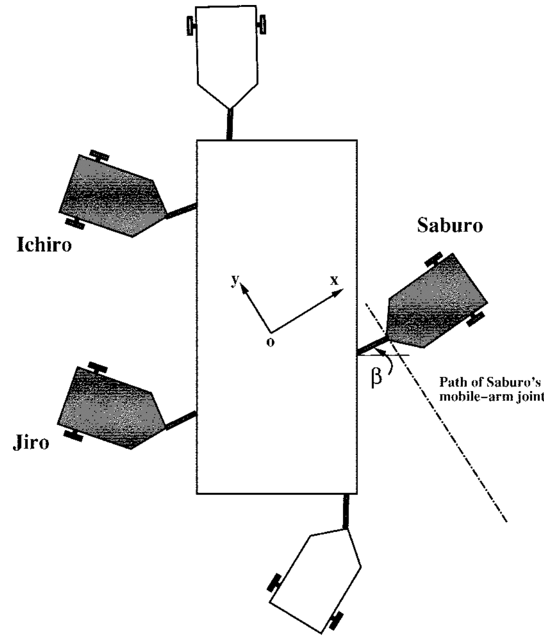


Fig. 8. Five robots, having friction contact with the object, move the load along a straight line, Y direction. Ichiro, Jiro, and Saburo constrain the object and the remaining two robots, white ones, transfer it to the goal.

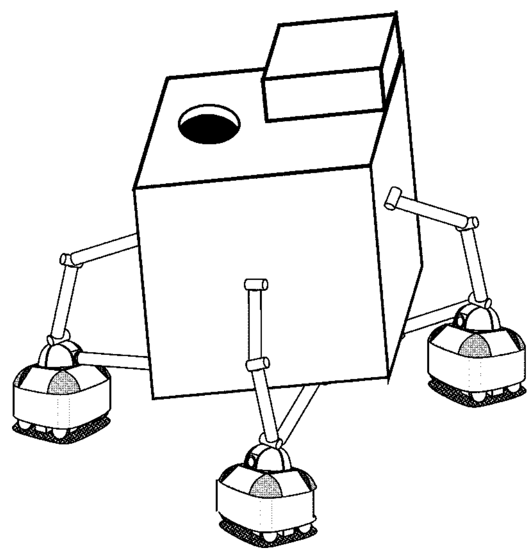


Fig. 9. Robots constrain the object with their arms and use their other arms to lift it.

VIII. Constraining to Lift

As a part of their mission, the robots must lift up the object and lower it down at its goal position. When lifting or lowering the object, the main concern is keeping the object stable; i.e., preventing it from tilting much and consequently turning it over.

The concept of *constrain and move* can be directly used to fulfill the above requirement. The object could be stabilized if and only if the forces and torques tend to destabilize the object can be canceled out by the robot-object interaction forces. This condition can be satisfied if the robots constrain the object from its sides. This is equivalent to making some vertical flexible guides for the object and move it up and down (see Fig. 9).

In this system, the states of the object constrained by the robots are its roll, pith, and yaw angles. The XY position of the

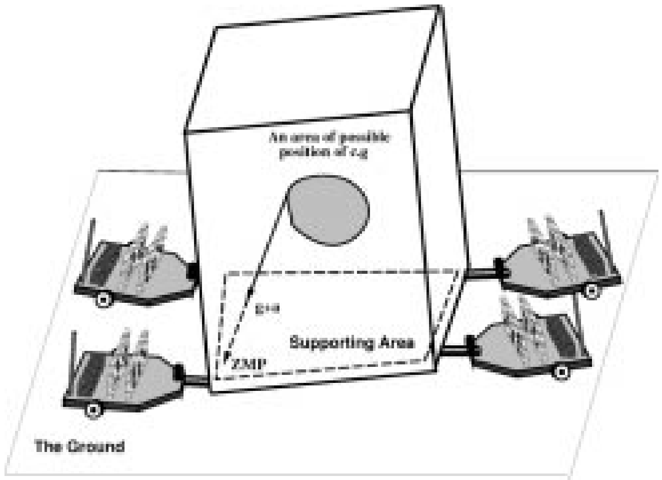


Fig. 10. ZMP and stability. A set of limits on the roll and pitch angles of the object can be found so that the ZMP remains inside the supporting area.

object is also confined passively. Vertical movement of the object is controlled by the lifting robots as long as the robot-object contacts are stable.

A. Implementing a Natural Constraint

At least four big robot arms are needed to constrain the object and a group of robots to lift it. Now we wish to remove the constraining arms from our system, in order to reduce the number and the complexity of the robots.

For lifting up, transferring, and lowering down the object, a group of cooperative fork lifting robots sustains the object on its bottom face [24]. It means that we can take benefit from gravity to partially constrain the object. Doing so, stabilizing the robot-object contact and controlling the robot-object interaction forces in a distributed fashion become much simpler.

The object inertial forces act to bring the object Euler angles to their stable values provided that the object ZMP (zero moment point) is inside the supporting area (see Fig. 10). In fact, these forces act like a nonlinear spring and produce the returning torques about the edges of the supporting area. In this strategy, the object lifting robots coordinate their motions so that the object ZMP remains inside the supporting area. This reduction in the number of the robots costs the lifting robots executing a distributed coordination protocol to fulfill the ZMP requirement.

IX. Lifting and Lowering an Object with Multiple Robots

Based on the relative position of each robot to the object center of gravity (c.g.), the required force to lift the object may vary from one robot to another. Also, there is a possibility that some of the robots start lifting the object with some delay or stop working for some time. Therefore, the distributed cooperative object handling strategy must cope with such situations with minimum volume of inter-robot communication.

A. Assumptions

We make the following general assumptions for our system to be of some practical usage.

- Mass of the object and its center of gravity are unknown. But, some very rough estimation can be made about the object weight and the possible area where its center of gravity lies.
- Geometry of the object is known (a top view).
- When arranged, the robots can lift the object cooperatively.[13]
- Robot-object contact is a friction-type contact.
- Each robot knows its own position and orientation.
- The object position and Euler angles are available to the robots.
- The maximum force each robot can apply is known.
- Robots are equipped with message broadcasting units.

B. A Mapping

As discussed, the object can be stabilized as far as its ZMP remains inside the supporting area.[14] Therefore, the first way to ensure the object stability is to measure the object acceleration and control it. This method in its original form necessitates implementing acceleration sensors and acceleration control.

Another way is that each robot measures its interaction force with the object and gains some kinematic information to decide what to do according to a distributed cooperation strategy. This method seems to be applicable for static systems, however, deeper considerations are needed before using it in a dynamic system. On the other hand, some force sensors must be installed in the robots and a force controller shall be used.

One of the assumptions that we made in Section IX-A is that some limits for the mass of the object and position of its center of gravity can be assumed. For an arrangement of the robots, knowing the maximum lifting power of the robots and using the lowest estimation of the object mass, maximum acceleration of the object can be calculated.

Therefore, a set of limits on the roll and pitch angles of the object can be found so that the ZMP remains inside the supporting area (see Fig. 10). Stability of the object can be ensured as long as the the object roll and pitch angles are smaller than these limits.

We have actually done a mapping by substituting the object acceleration with its Euler angles as the control parameters. This mapping is a function of the acceleration space to the configuration space. This mapping, conservative however, helps us to work with some kinematic information without neglecting the system dynamics.

Some compliance is provided at the end-effector (in the plane parallel to the object lower surface). Therefore, if the object tilt angle is small enough, there is no need for the robots to move while lifting or lowering the object.

C. Two- and Three-Dimensional Systems

To control the object tilt angle it in a two-dimensional space, each robot needs to know on which side of the object center of gravity it is located and in which direction the object has been rotated.

[13]This means, a proper arrangement of the robots can be made, e.g., [34] and [41].

[14]In this work, the object is called stable if its states are remained in defined ranges.

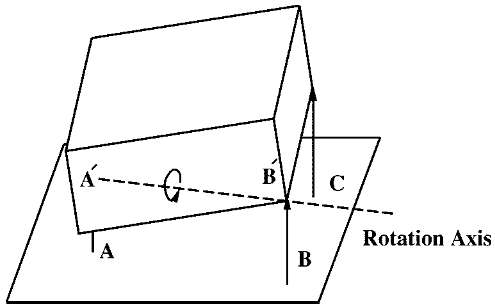


Fig. 11. Three robots lift the object in a three-dimensional space. Unlike a two-dimensional system, the direction and position of the rotation axis change according to the relative motion of the lifting robots.

The problem is much more complicated when the robots are not on a straight line—a three-dimensional case. This complexity is related to the fact that the direction of the rotation axis may change in such systems. The rotation vector is the function of the robots' relative velocity and position of their contact points (see Fig. 11).

Considering the above discussions, it is clear that the simple methods developed for two-dimensional object manipulating robots (e.g., [7] and [29]) cannot be directly applied to object lifting/lowering systems.

D. Control Parameter

Difference in lifting speed of the lifting robots affects both position and direction of the instantaneous object rotation axis. Therefore, to control both roll and pitch angles of the object, lifting velocities of the cooperative robots must be controlled strictly. Coordinating lifting speed of the robots is a difficult task for a distributed team of robots without explicit communication, especially when the robots are working under different loads, some of the robots start operating with some delays, or halt their movement for a while.

Any angle sensor detects the target configuration in its coordinate system. Also, limits on the object Euler angles are usually defined in the object coordinate system. Therefore, the mapping function between each sensor and the object coordinate system must be known (see Fig. 12). But, when the roll and pitch angles of the object are small, the object tilt angle τ is measured as

$$\tau^2 = (\alpha_i - \alpha_{s_i})^2 + (\beta_i - \beta_{s_i})^2 \tag{15}$$

when α_i (β_i) is the object roll (pitch) angle in ith coordinate system and α_{si} (β_{si}) is the stable roll (pitch) angle of the object in the same coordinate system.

An interesting point about τ^2 is that it is independent of the measuring system configuration. For most of the objects, zero pitch and roll angles ($\alpha_{s_i} = \beta_{s_i} = 0$) is the referenced configuration. Therefore, if τ^2 is kept smaller than its maximum permitted value

$$\tau^2 \leq Min(\min(\alpha^2_{\max}), \min(\beta^2_{\max})) \tag{16}$$

the object roll and pitch angles do not exceed their specified limits.

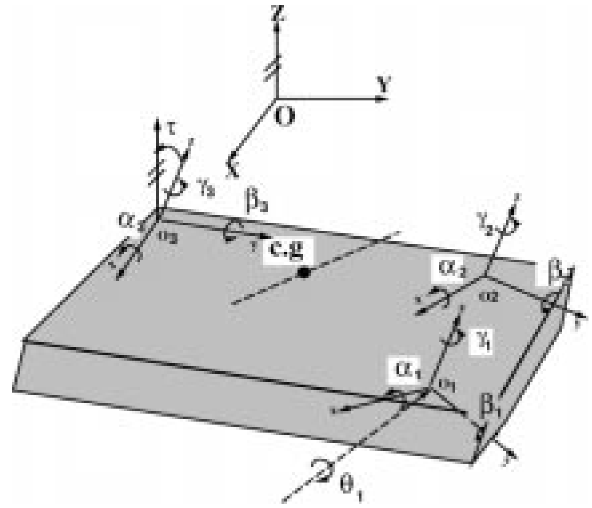


Fig. 12. Euler angles of the object in three different coordinate systems. Other robot position vectors are also defined as vectors connecting the robot-object contact point (e.g., O_1) to others (O_2 and O_3) in the robot end-effector coordinate system.

E. A Distributed Cooperation Strategy for Lifting and Lowering the Object

To reduce the object tilt angle when it reaches its critical value, the robots must cooperate to turn the object around a suitable axis in a proper direction. Coordinating the robots to keep the rotation axis in an arbitrary orientation is a difficult task. The task would be simplified if the robots use some special rotation axis. In this work, these rotation axis are the lines connecting each pair of robot-object contact points.

> *The Cooperation Strategy:* When lifting (lowering) the object cooperatively and the tilt angle of the object reaches its limit value, the robot(s) having the lowest (highest) contact with object moves upward (downward) faster, while the other ones stop.

In this protocol, since all of the robots except the one(s) having the lowest (highest) virtual contact point[15] with object are stopped, the rotation axis of the object is aligned with one of the lines connecting a pair of robot-object virtual contact points. Fig. 13 shows three robots implementing the above cooperation strategy to lift or lower the object.

Stability of the proposed strategy is proved in [24].

1) Information system: Each robot is required to know the object angle and check whether it is the lowest/highest robot. There are some equivalent sensory systems [8] to obtain this data. These information systems are discussed in [23] and [25]. Here under, we introduce the most applicable distributed information system in which the minimum static data communication is used.

Each robot is equipped with a sensory system measuring the object Euler angles in its end-effector coordinate system (Fig. 12) and calculates the object tilt angle. Also, having the object Euler angles and position vector of the other robots (all represented in its end-effector coordinate system, see Fig. 12),

[15] A virtual contact point, as defined in [27], is the projection of the contact point in to a flat plane.

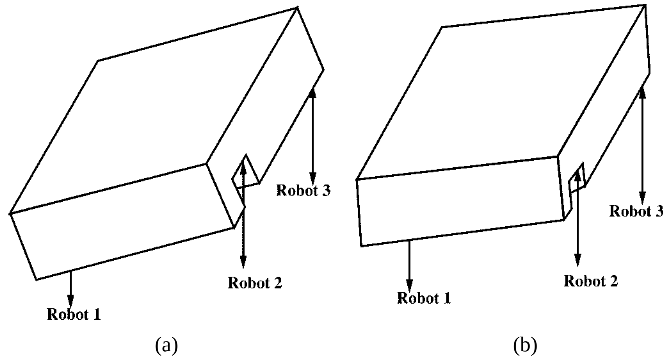


Fig. 13. An example of executing the proposed cooperation strategy. (a) The group is lifting the object and the tilt angle is reached its critical limit. In this situation, robot one is the lowest one and moves upward faster while the others are stopped. (b) To reduce the tilt angle of the object when lowering it, the third robot moves downward faster and the other robots stop.

each robot can check if it has the lowest/highest virtual contact point with the object or not.

As far as the robot-object contacts are stable, the position of the robot-object contact positions are constant, when viewed from other robots end-effector. Therefore, each robot is required to broadcast its effector-object position in the world coordinate system only when initializing the system or when there is a considerable change in its contact position with the object. A full description of the implemented communication protocol is given in [10].

F. Robot Controller and Behavior

A simple controller for realizing the cooperation strategy is designed. Organization of the robot controller is based on the subsumption architecture [3].

Fig. 14 shows the controller. In this architecture, the *maintain contact* module is the lowest competitive part working to keep the robot/object contact stable.

The *accelerate* behavior tries to apply more force to the object when it is activated. In other words, it is a wild behavior and tries to accelerate the object. In conjunction with other behaviors, this module enables the robot to work under different and varying loads.

To prevent the system from moving faster than a specified speed, the lifting speed of each robot is restricted by the third layer—*do not go fast*.

When the robot is close to its goal, the *stop at goal* module takes control and damps the speed of the end-effector.

The *reduce object angle* unit aims at keeping the object tilt angle smaller than the defined value. This module (Fig. 15) consists of three layers: hurry up,[16] slow down, and stay firm. When the object tilt angle is going to reach its defined limit and the robot is the lowest (highest) one, it accelerates. When a robot is not the lowest (highest) member, it reduces the motor command to stop. If the end-effector speed is going to become negative (positive), the robot stays firm.

Because of having dynamic interaction forces between the robots and the object, realizing *stay firm* behavior with implementing ordinary controllers is a hard task. But, *stay firm* can be realized easily if the lifting mechanism is not back-derivable.[17]

[16]Actually, hurry up and accelerate behaviors are the same modules. Here, they are separated to simplify the explanations.

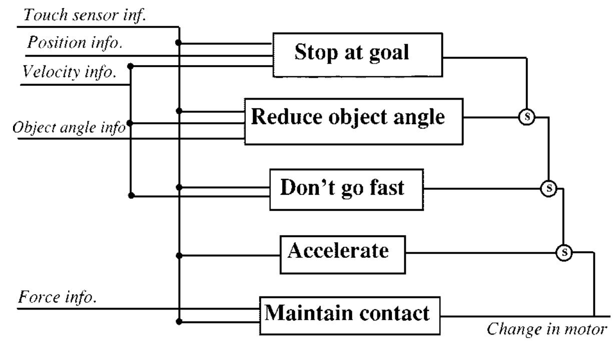


Fig. 14. Robot controller to lift and lower the object in cooperation. The robot behaviors are organized in a subsumption architecture.

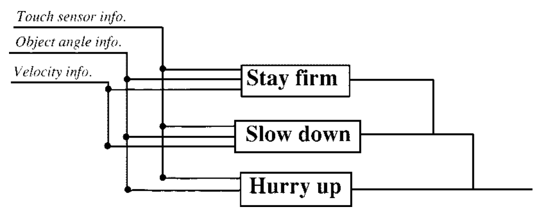


Fig. 15. A closeup of the *reduce object module* behavior of the robot.

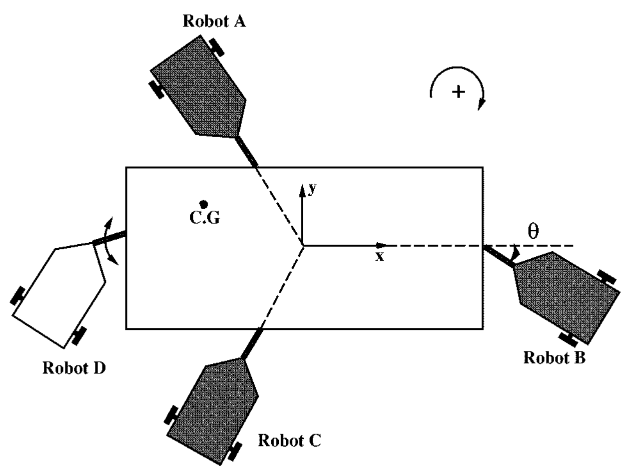


Fig. 16. Geometrical model of the simulated system. Robots A, B, and C constrain the object. By changing its arm direction, robot D turns the object in both directions. There is an error in robot B arm angle with the object.

X. Simulation Results

A. Object Reorientation

A series of dynamic computer simulations are conducted to verify the proposed cooperation strategy and study the effect of incorporating compliant elements in the robot arms.

Fig. 16 shows the model used for the simulations. In this model, the object is a 0.5×0.3 rectangle of unit mass and moment of inertia. The constraint-making robots (A, B, and C) are located at (0.15, -0.05), (0.0, 0.25), and (-0.15, -0.05), respectively. It is assumed that there is an error (θ) in orientation of robot B. Also, to observe the effect of this error on the re-

[17]This is what we call encoded robot behavior [1].

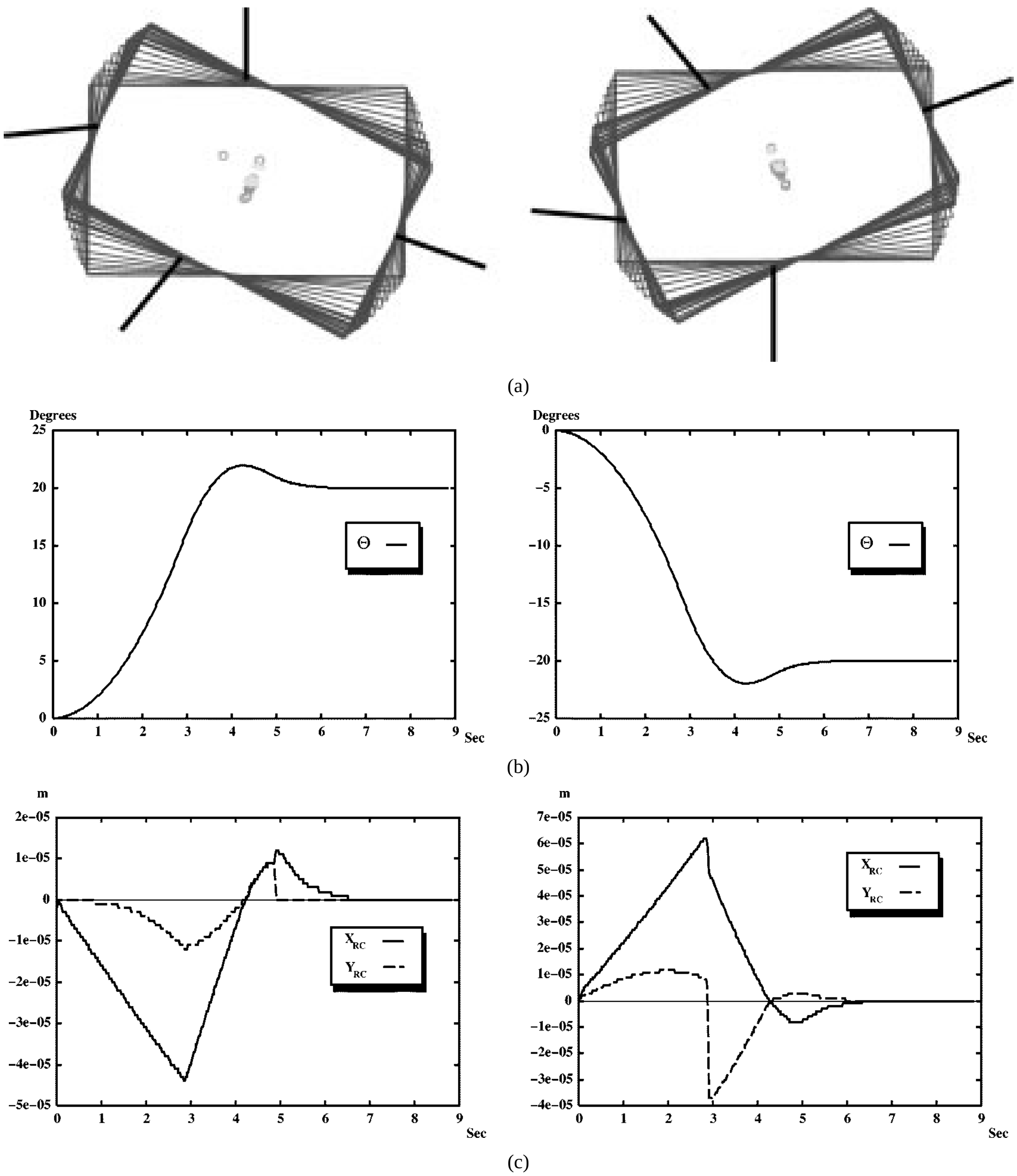


Fig. 17. (a) Locus of the object center of rotation. (b) Orientation of the object as a function of time. (c) Position of the object desired center of rotation. The robots are stiff and there is no error in the orientation of the robot arms.

sults, it is taken that the arm-object angles are constant when turning the object. The maximum force applied by the object turning robot (left-hand side robot) is set to $1N$. Fig. 17 shows the simulation results when the spring and the damping coefficient of the robot arms are 10^6 N/m and 500 Ns/m, respectively, and there is no error in orientation of the robot arms. The object center of gravity is located at (0.2, 0.2) m from the desired center of rotation.

Displacement of the desired center of rotation is quite small. Similar results were obtained for different positions of the object center of gravity. But, due to dynamic coupling between angular and linear movements of the object, some of the object instantaneous center of rotations are scattered around the desired one.

There are two main explanations for these center of rotations. When rotating the object, some robots must move forward or extend their arms to stabilize their contact with the payload. To keep constraining the object, these robots compensate their forward movement whenever the object starts pushing them back. When the robots move backward in their arm direction, velocity of their contact point can have a nonzero component in their arm-mobile directions. Therefore, during this short period, the restriction on the possible position of the object center of rotation is partially relaxed. Having limited stiffness in the robot arms is the second reason for getting such results.

In Fig. 18, there is 2° error in the arm-object angle of the right-hand side robot. Since the object has a very high rotational stiffness in the clockwise direction, the left-hand side robot cannot turn the object in this direction. In this figure, locus of the object's instantaneous center of rotation is very dense around the desired center of rotation.

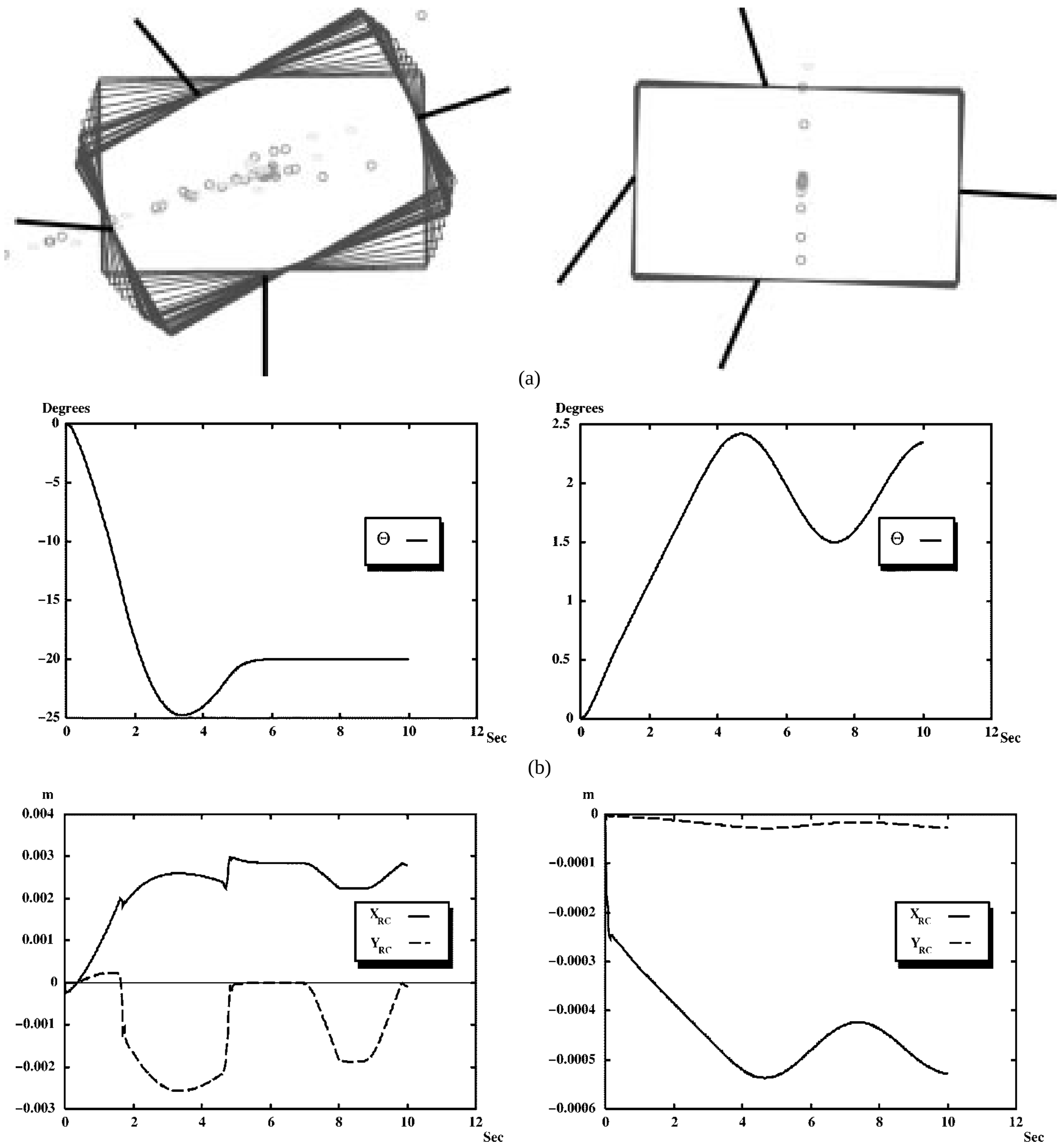


Fig. 18. (a) Locus of the object center of rotation. (b) Object angle versus time. (c) Position of the desired center of rotation. The robots are stiff and there is two degrees error in orientation of the right-hand side robot arm.

It is discussed in [23] and [26] that, by showing a compliant behavior, a robot can cover its position error to some extent. In Fig. 19, the spring and the damping coefficient of the robot arms are 1000 N/m and 50 Ns/m, respectively, and there is 2° error in the angle between the right-hand arm and the object. Compliance in each robot arm accommodates some considerable motion for its contact point in arm-mobile direction. Therefore, the jamming problem does not occur and the robots can rotate the object in both directions. Also, as the object's stiffness matrix has a component countering the clockwise rotations, some residual forces of small magnitude are observed.

Left-hand side figures in Fig. 19 shows the situation where the object is rotated 20° in the counterclockwise direction. Compared with the results in Fig. 18, position of the center of rotations are more scattered around. Therefore, we have more vibrations in the object and a larger position error.

B. Moving the Object

A team of four robots are used to simulate the dynamic object carrying system. The physical parameters of the object and the robots are the same as in Section X-A.

Position of the robot-object contact points in the object coordination system, attached to the object center of geometry, are (all in meters) as follows: (0.15, 0.1), (−0.15, −0.1), (−0.6, 0.25), and (−0.06, −0.25).

In the developed cooperation strategy, the arm is perpendicular to the designed path of the object. Also, the arm-object angle must be smaller than the friction cone angle (Fig. 8). Therefore, there is a limit on the maximum angle of the desired path. Consequently, the robots may have to move the object to the goal in more than one step. A simple path planning method to find the via points and paths are given in [27].

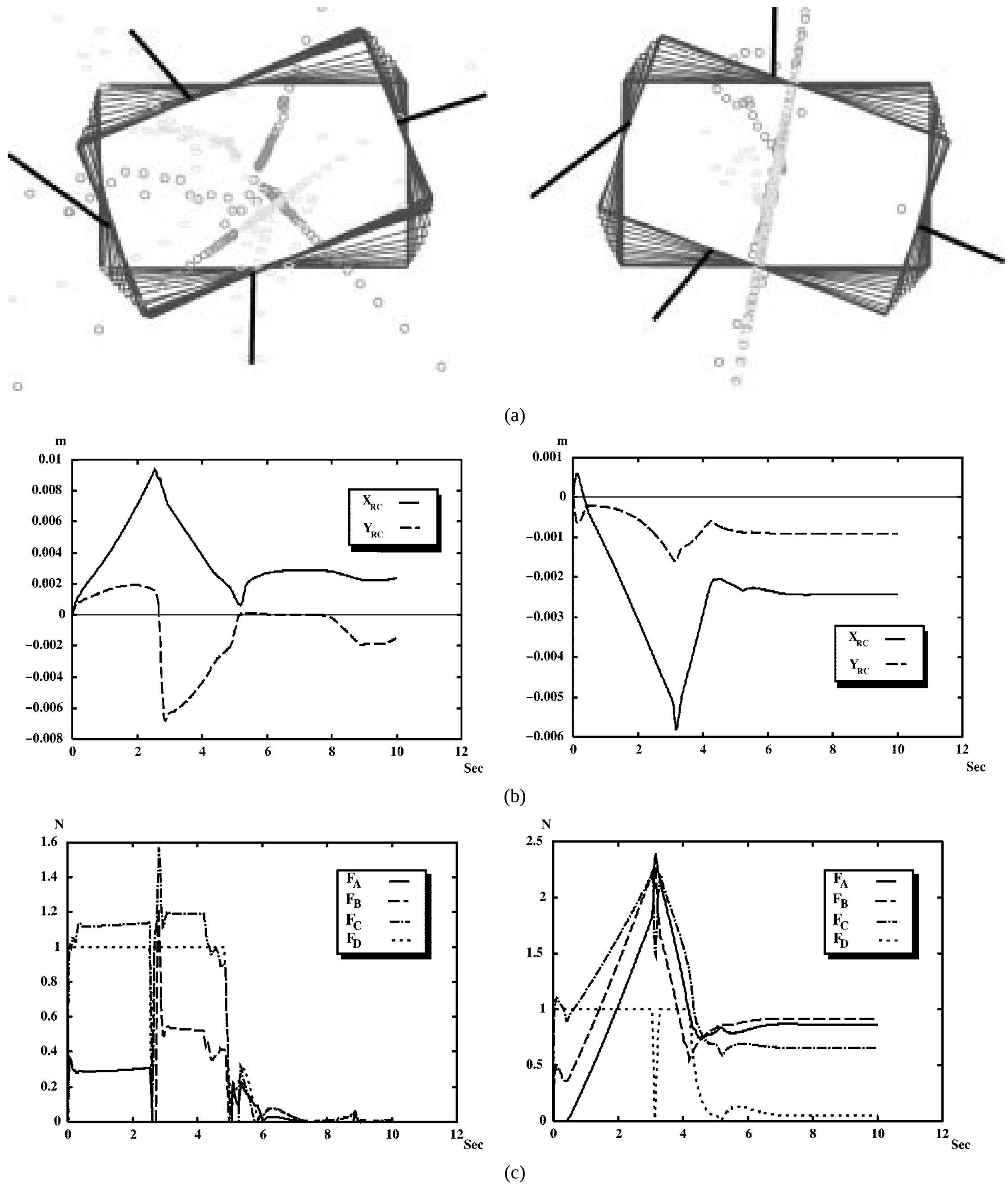


Fig. 19. (a) Locus of the object center of rotation, (b) position of the object desired center of rotation, and (c) the robot-object interaction forces. There is 2° error in the arm-object angle of the right-hand side robot. The spring and the damping coefficients of the arms as 1000 N/m and 50 Ns/m, respectively.

The above restriction on the path of the object is related to the type of the robot-object contact. In other words, the robots can move the object on an arbitrary straight path if their contact with the object is naturally stable (e.g., joint contact) and the robot arms are sufficiently long. This condition on the length of the arm will also be lifted if the mobile bases are holonomic.

Fig. 20 shows the robots when moving the object to its subgoal. The robots transfer the load to its final destination from there.[18]

C. Lifting/Lowering Task

A cube of 0.5 m × 0.3 m × 0.2 m size, resting on four three-dimensional springs and dampers, is chosen as the object. Also, goal of each robot in the Z direction is set to 0.09 m above the ground.[19] Moreover, it is assumed that the maximum permitted object tilt angle and the maximum speed of each lifter are about 2.3° and 0.011 m/s, respectively.

A coordinate system is fixed at the object center of geometry and, its X axis is aligned with the long side of the object. XY coordinates of the robot-object contact points are located at (0.2, −0.13), (0.0, 0.13), and (−0.2, −0.13), respectively. The lifting

[18]After stopping at the subgoal, the robots change their arm-object angle according to the requirements of the second portion of the object path.

[19]Goal of the robot in the Z direction is a fuzzy parameter having the value of 9 cm and 5 mm base width.

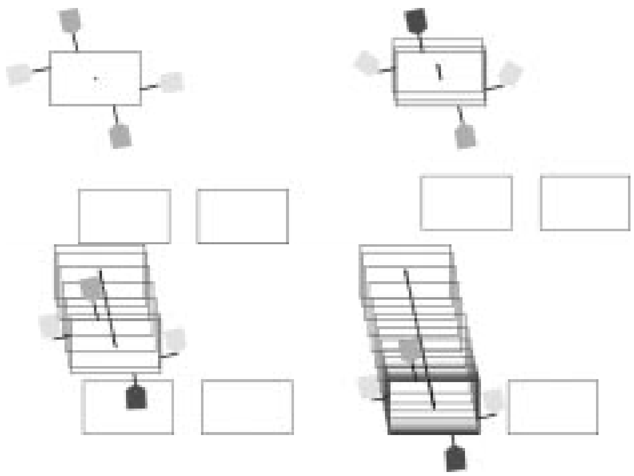


Fig. 20. Because of having friction contact with the object, the robots are not able to move it to the goal directly. Therefore, they first transfer the load to an intermediate goal.

mechanism is assumed to be nonback-drivable and, to prevent excessive shocks on the system, a spring-damper unit is located between the fork and the lifting mechanism.

Simulation results reported in [24] show that when the load distribution on the robots is relatively even, the robots are able to keep the object tilt angle in the desired range by controlling the maximum speed of the lifting actuators.

Fig. 21 shows the simulation results when mass of the object is 8 kg and its center of gravity is located at (0.0, 0.05, 0.0) in the object coordinate system. In this simulation, load on the second robot is higher than on the others. Therefore, its teammates have to wait for it many times. In fact, the second robot is not able to suspend the object in the appropriate time. Hence, *Reduce Object Angle* behavior is activated many times to keep the object tilt angle in its safe region.

In the position curve of Fig. 21, we can observe some delay in the upward movement of the first and the third robots after the object Euler angles are decreased. This behavior is due to nonback-drivability of the lifting mechanisms [27].

Fig. 22 shows the simulation results when mass of the object is 8 kg and its center of gravity is located at (0.04, −0.01, 0.0) in the object coordinate system. The results indicates that the robots successfully implement the coordination strategy and control the object tilt angle.

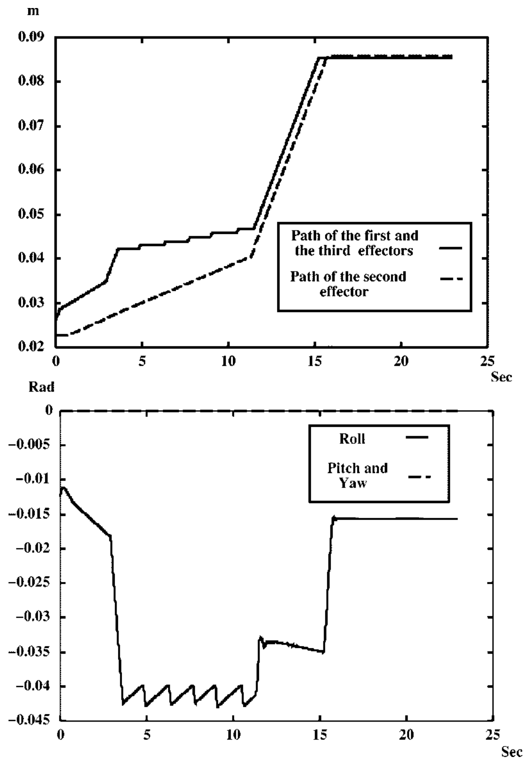


Fig. 21. Path of the end-effectors in the vertical direction and Euler angles of the object during the lifting process of an 8-kg object with its c.g. at (0.0, 0.05, 0.0) of the center of geometry.

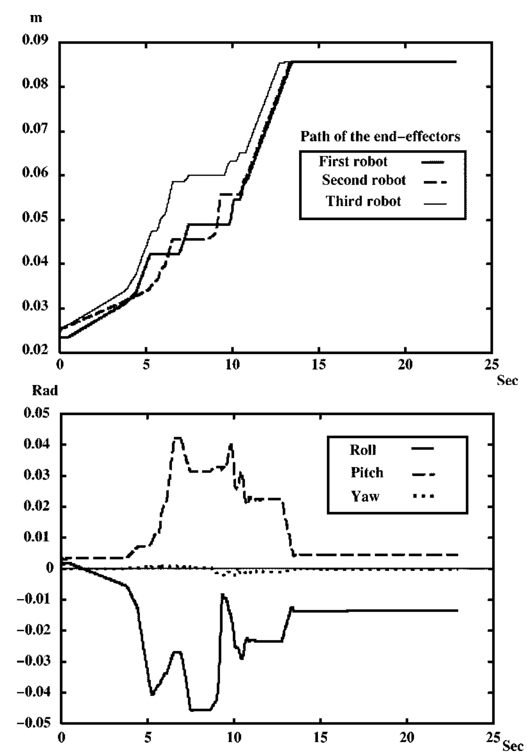


Fig. 22. Path of the end-effectors in the vertical direction and Euler angles of the object during the lifting process for an 8-kg object with its c.g. at (0.04, −0.01, 0.0) of the center of geometry.

XI. Experimental Results

The main goal of performing the reported experiments are to show the applicability of the proposed strategies and the detailed experimental data are not discussed here.

A. Object Transferring Task

In the first experiment, the constraining strategy for object reorientation is tested on Hamcar robots. In this experiment, there is no robot to turn the object and the robots are expected to keep the desired center of rotation in the planned location.

In the second experiment, the total object reorientation strategy is tested.

In the third experiment, the constraining strategy for object carrying task is tested when three robots constrain the object on a straight line and an external force is exerted on the object.

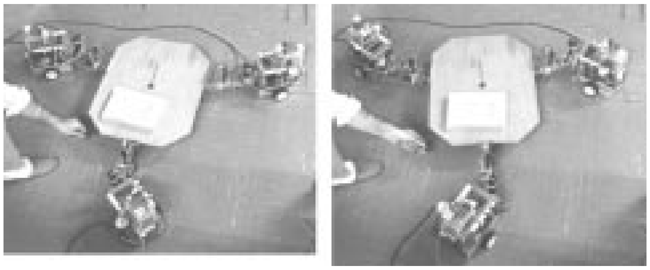


Fig. 23. Ichiro, Jiro, and Saburo constrain the object. The object rotates about the designed center of rotation when it is pushed by a sharp pencil.

The robots are expected to keep the object on its path. The last experiment is devoted to check the total strategy to move the object on a straight line in one direction.

Fig. 23 shows the robots when constraining the object. As expected, the robots make a force direction closure on the object. Therefore, the system cannot resist the external torques but cancels all of the external forces out.[20]

Some errors in position of the object center of rotation are observed. These are related to having error in controlling and initializing the robot-object angles, existence of compliance (software and hardware), presence of friction forces in the system, and error in dead-reckoning of the mobile bases.

Fig. 24 shows Hamcar when turning the object. In this experiment, three robots constrain the payload and the fourth one turns it.

Fig. 25 shows the result of a simple experiment when three robots constrain the object to move it in the direction toward the camera. As desired, the object is free only along its desired path, and it is moved when a small force is applied.

Fig. 26 shows the experimental result when a robot pushes the constrained object toward its goal. It is noteworthy that all of these experiments are done at least ten times and about 30% failure is observed when the length of the desired path is set to 1 m. The main reason for this problem is error in the dead reckoning data. To reduce this problem, we intend to improve our dead reckoning scheme by adding new hardware to the system and compensating the wheel slippage and systematic system errors.

B. Object Lifting/Lowering Task

A hard cover book of size 30 × 21 × 4 cm weighing 1.7 kg is chosen for the experiments. Successful experimental results when lifting uneven load distribution are given in [24]. In this paper, the results showing particular features of the system are reported. In these experiments, the robots are expected to lift/lower the object even when there are some delays or failures in the system.

Being able to communicate with the other robots through the environment is one of the main characteristic of the developed system. To show this feature, Saburo started lowering the object with 1.5-s delay (Fig. 27). In this experiment, without having any information about delay in Saburo's start time, Ichiro and Jiro moved the object downward until the object tilt angle reached its assigned limit and waited for Saburo to correct the object tilt angle. After the object angle entered the safe region, Ichiro and Jiro resumed moving downward.

[20]The same response is observed when turning a crank.

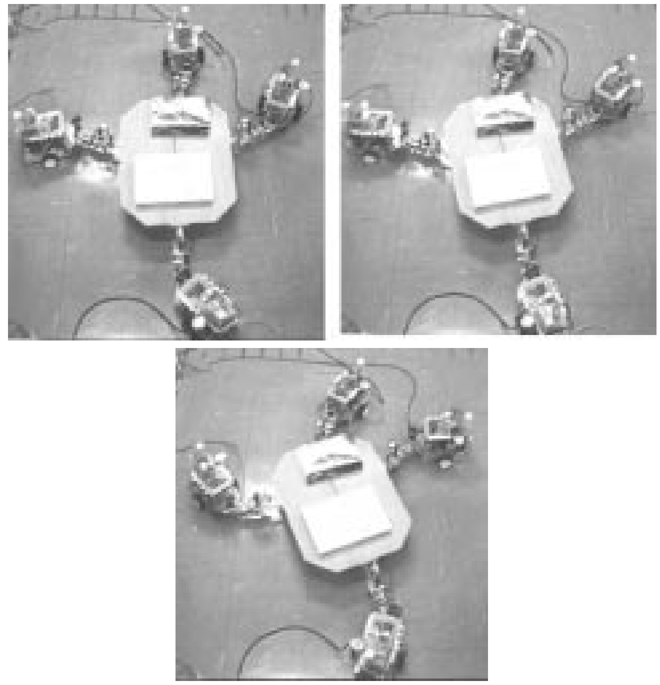


Fig. 24. Three shots of the experiments when a group of robots constrain the object and Shiro, the robot on the upper side of the object, turns it.

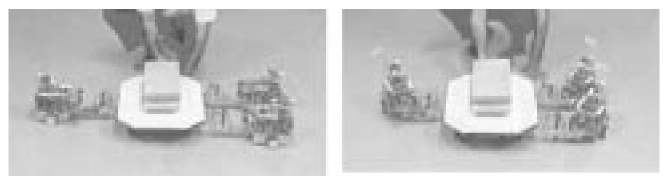


Fig. 25. Ichiro, Jiro, and Saburo constrain the object. The external force component in the desired direction moves the object toward its goal while the robots constrain the object.

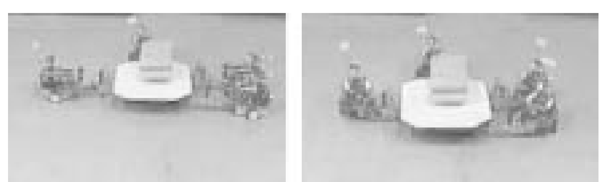


Fig. 26. Three robots constrain the object and another robot pushes it toward its goal.

The system showed the same performance, say a simulated fault tolerance, under external disturbances. In this test, some external forces are applied to the system (Fig. 28). Jiro stopped under these external disturbances, and Ichiro and Saburo continued their job as far as the object angle was in the safe area. Meanwhile, Jiro did its best to overcome the external force. When the external disturbance was removed, the robots continued to move the object upward.

XII. Conclusion

Learning from the mechanical systems, a new strategy, called *constrain-move* strategy, to decompose the object handling mis-

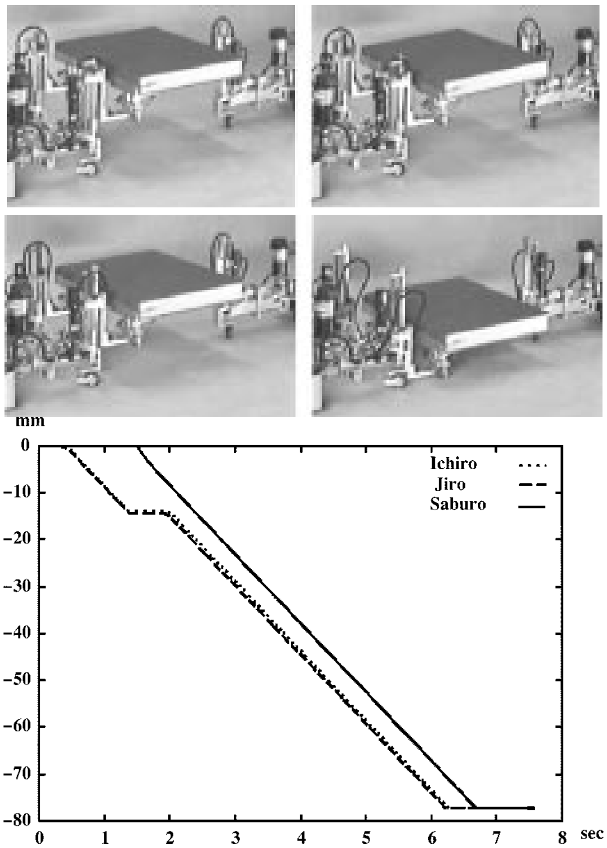


Fig. 27. The 1.5-kg object lowering process and path of the end-effectors. Saburo (the robot on the right-hand side) begins its operation with 1.5-s delay without informing its teammates.

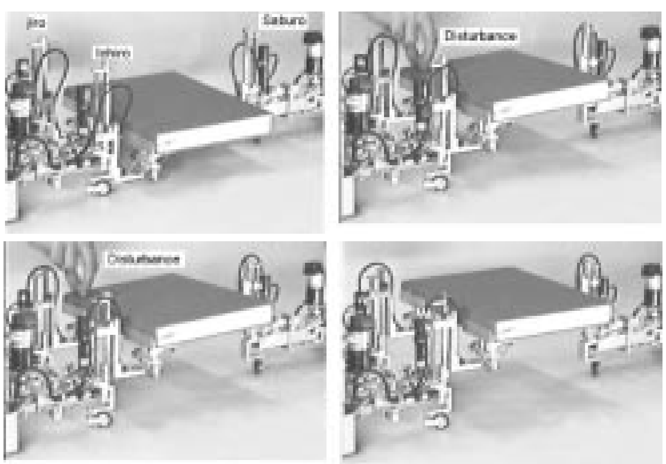


Fig. 28. Lifting an object of 1.5-kg mass cooperatively when Jiro is under external disturbances.

sion into two independent subtasks is introduced. Based on the proposed idea, the cooperative robots passively constrain the objects in all directions but along the desired path and then move it.

In this approach, the object is free along the desired path and has small compliance (high stiffness) in the other directions. In fact, the resulting compliance is used to allocate the task to the distributed robots.

Using *constrain-move* strategy, two distributed cooperation protocols for a group of multiple mobile robots to reorient an object about a fixed point and to move it along a straight line are introduced. The cooperation strategies are designed in such a way that stabilizing the robot-object contact is simplified.

The robots implement a position control scheme to constrain the object. Therefore, any error in position of the rigid robots locks the object movement. This problem is solved by incorporating some mechanical compliant elements in the robot arms.

Constrain-move concept is also used to develop a distributed cooperation strategy to lift up and lower the object cooperatively.

It is discussed that the robots may use the weight of the object as the constraining force. Therefore, there would be no need for a group of robots to constrain the load, and consequently, the size and complexity of the object handling team can be reduced.

Ability of the system to operate under different and unknown loads and its fault tolerance, a direct result of communicating through the object, are demonstrated through computer simulations and experiments.

Our robots are controlled by behavior-based controllers, however, application of the proposed cooperation strategies is not restricted to such type of controllers.

We are currently working on generalizing the *constrain-move* concept to develop distributed cooperation strategies for carrying the object along an arbitrary path. In addition, we would like to feedback some force information into the robot controllers to improve their performance.

REFERENCES

[1] J. Adams, R. Bajcsy, J. Kosecka, V. Kumar, M. Mintz, M. Mandelbaum, C. Wang, Y. Yamamoto, and X. Yun, “Cooperative material handling by human and robotic agents: Module development and system synthesis,” *Expert Syst. Applicat.*, vol. 11, no. 2, pp. 89–97, 1996.

[2] A. Asama, “Trends of distributed autonomous robotic systems,” in *Distributed Autonomous Robotics System*. Tokyo, Japan: Springer-Verlag, 1994.

[3] R. A. Brooks, “A robust layered control system for mobile robots,” *IEEE J. Robot. Automat.*, vol. RA-2, pp. 14–23, Feb. 1986.

[4] R. G. Brown and J. S. Jennings, “A pusher/steerer model for strongly cooperative mobile robot manipulation,” in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, vol. 3, 1995, pp. 562–568.

[5] Y. Cao, A. S. Fukunaga, A. Kahng, and F. Meng, “Cooperative mobile robots: Antecedents and directions,” in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, vol. 1, 1995, pp. 226–243.

[6] M. Cherif and K. Gupta, “Planning quasistatic motions for reconfiguring objects with a multi-fingered robotic hand,” in *Proc. 1997 IEEE Int. Conf. Robotics and Automation*, 1997, pp. 986–991.

[7] B. R. Donald *et al.*, “Analyzing teams of cooperative mobile robots,” in *Proc. IEEE Int. Conf. Robotics and Automation*, 1994, pp. 1896–1903.

[8] B. R. Donald, “On information invariants in robotics,” *Artif. Intell.*, vol. 72, pp. 217–304, 1995.

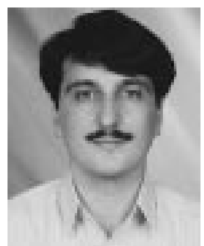
[9] B. R. Donald, D. Rus, and J. Jenning, “Information invariants for distributed manipulation,” *Int. J. Robot. Res.*, vol. 16, no. 5, pp. 673–702, Oct. 1997.

[10] K. Hanada, T. Takahashi, and E. Nakano, “Software architecture for constructing a mobile robot with highly independent modules,” in *Proc. Fourth Int. Conf. Control, Automation, Robotics, and Vision*, vol. 2, Singapore, 1996, pp. 2902–2907.

[11] M. Hashimoto, F. Oba, H. Nakahara, K. Imamaki, and T. Eguchi, “Trajectory generation and tracking control methods for a multiple transfer robots system,” in *Proc. IEEE/RSJ Int. Workshop on Intelligent Robots and Systems*, 1991, pp. 799–804.

[12] M. Hashimoto, F. Oba, and T. Eguchi, “Dynamic control approach for motion coordination of multiple wheeled mobile robots transporting a single object,” in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 1993, pp. 1944–1951.

[13] R. Ikeura and H. Inooka, "Variable impedance control of a robot for cooperation with a human," in *Proc. IEEE Int. Conf. Robotics and Automation*, 1995, pp. 3097–3102.
[14] O. Khatib *et al.*, "Vehicle/arm coordination and multiple mobile manipulator decentralized cooperation," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robotics and Systems*, vol. 2, Osaka, Japan, 1996, pp. 546–553.
[15] M. Koga, K. Kosuge, K. Furuta, and K. Nosaki, "Coordinated motion control of robot arms based on the virtual internal model," *IEEE Trans. Robot. Automat.*, vol. 1, pp. 77–85, Feb. 1992.
[16] K. Kosuge and T. Oosumi, "Decentralized control of multiple robots handling an object," in *Proc. 1996 IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Osaka, Japan, 1996, pp. 318–323.
[17] K. Kosuge, Y. Hirata, H. Asama, H. Kaetsu, and K. Kawabata, "Motion control of multiple autonomous mobile robots handling a large object in coordination," in *Proc. IEEE Int. Conf. Robotics and Automation*, 1999, pp. 2666–2673.
[18] M. T. Mason, "Kicking the sensing habit," in *AAAI Fall Symp. Series Working Notes*, Nov. 14–17, 1991.
[19] M. J. Mataric, M. Nilsson, and K. T. Simsarian, "Cooperative multi-robot box-pushing," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, vol. 3, 1995, pp. 556–561.
[20] Y. Nakamura, *Advanced Robotics: Redundancy and Optimization*. Reading, MA: Addison-Wesley, 1990.
[21] E. Nakano, S. Ozaki, T. Ishida, and I. Kato, "Cooperational control of the anthropomorphuos manipulator MELARM," in *Proc. Fourth Int. Symp. Industrial Robots*, 1974, pp. 251–260.
[22] V. D. Nguyen, "Constructing force-closure grasps," in *Proc. 1986 IEEE Int. Conf. Robotics and Automation*, 1986, pp. 1368–1373.
[23] M. N. Ahmadabadi and E. Nakano, "A cooperative multiple robot system for object lifting and transferring tasks," in *Proc. ROBOMECH'96*, Japan, 1996, pp. 742–745.
[24] ——, "A cooperation strategy for a group of object lifting robots," in *Proc. 1996 IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, vol. 1, Osaka, Japan, 1996, pp. 125–131.
[25] ——, "Robot behavior and information system for multiple object handling robots," in *Distributed Autonomous Robotics System 2*. Tokyo, Japan: Springer-Verlag.
[26] ——, "Constrain and move: A new concept to develop distributed object transferring protocols," in *Proc. 1997 IEEE Int. Conf. Robotics and Automation*, Albuquerque, NM, 1997, pp. 2318–2325.
[27] M. N. Ahmadabadi, "A Study on Cooperative Object Handling Task with Multiple Mobile Robots," Ph.D. dissertation, GSIS, Tohoku Univ., Sendai, Japan, 1997.
[28] M. N. Ahmadabadi and E. Nakano, "Task allocation and distributed cooperation strategies in a group of object transferring robots," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Grenoble, France, 1997, pp. 435–440.
[29] L. E. Parker, "Heterogeneous multi-robot cooperation," Ph.D. dissertation, MIT, Cambridge, MA, 1994.
[30] ——, "ALLIANCE: An architecture for fault tolerant multirobot cooperation," *IEEE Trans. Robot. Automat.*, vol. 14, pp. 220–240, Apr. 1998.
[31] G. A. Pratt *et al.*, "Stiffness isn't everything," in *Proc. Fourth Int. Symp. Experimental Robotics (ISER'95)*, Stanford, CA, 1995.
[32] D. Rus, "In-hand dextrous manipulation of piecewise-smooth 3-D objects," *Int. J. Robot. Res.*, vol. 18, no. 4, pp. 355–381, Apr. 1999.
[33] ——, "Fine manipulation planning for dextrous manipulation," PhD dissertation, Dept. Comput. Sci., Cornell Univ., 1992.
[34] J. Sasaki *et al.*, "Cooperative grasping of a large object by multiple mobile robots," in *Proc. IEEE Int. Conf. Robotics and Automation*, 1995, pp. 1205–1210.
[35] D. J. Stilwell and J. S. Bay, "Toward the development of a material transport system u sing swarms of ant-like robots," in *Proc. IEEE Int. Conf. Robotics and Automation*, 1995, pp. 766–771.
[36] M. Uchiyama and P. Dauchez, "A symmetric hybrid position/force control scheme for the coordination of two robots," in *Proc. IEEE Int. Conf. Robotics and Automation*, 1988, pp. 350–355.
[37] P. Velletri, "Study of a robot manipulator incorporating passive compliance," M.S. thesis, GSIS, Tohoku Univ., Sendai, Japan, 1996.
[38] Z. D. Wang, E. Nakano, and T. Matsukawa, "Realizing cooperative object manipulation using multiple behavior-based robots," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, vol. 1, Osaka, Japan, 1996, pp. 310–317.
[39] Z. D. Wang, M. N. Ahmadabadi, E. Nakano, and T. Takahashi, "A multiple robot system for cooperative object transportation with various requirements on task performing," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 1999, pp. 1226–1233.
[40] Y. Yoshikawa, "Passive and active closures by constraining mechanisms," in *Proc. IEEE Conf. Robotics and Automation*, 1996, pp. 1477–1484.
[41] Y. Yoshikawa, M. Kurisu, and T. Kozuka, "Deciding grasping position and regrasping action by cooperative multiple mobile robots," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, vol. 1, 1996, pp. 118–124.
[42] Y. Yoshikawa and X. Zheng, "Coordinated dynamic hybrid position/force control for multiple robot manipulators handling one constrained object," *Int. J. Robot. Res.*, vol. 12, pp. 219–230, June 1993.
[43] A. Asama, "Trends of distributed autonomous robotic systems," in *Proc. DARS'94*, Riken, Japan, 1994.
[44] M. N. Ahmadabadi and E. Nakano, "Robot behavior and information system for multiple object handling robots," in *Proc. DARS'96*, Saitama, Japan, 1996.



Majid Nili Ahmadabadi was born in 1967. He graduated from Sharif University of Technology of Iran in 1990. He received the M.Sc. and Ph.D. degrees in information sciences from the Graduate School of Information Science, Tohoku University, Sendai, Japan, in 1994 and 1997, respectively.

In 1997, he joined the Advanced Robotics Laboratory at Tohoku University. Later, he moved to the Department of Electrical and Computer Engineering, Faculty of Engineering, University of Tehran, where he is the Head of Robotics and AI Laboratory. He is also a Senior Researcher at the Institute of Theoretical Physics and Mathematics. His main research interests include distributed robotics and artificial intelligence, mobile robots, and cooperative learning in multi-agent systems.

Dr. Ahmadabadi initialized the Iranian National Robot Contests in 1999 and is the President of the Executive Committee of these games.

Eiji Nakano was born in 1942. He graduated from the University of Tokyo in 1965, and finished the graduate studies in 1970.

In 1970, he joined the Mechanical Engineering Laboratory of the Ministry of International Trade and Industry, and started to research robotics. In 1987, he moved to Tohoku University as a Professor at the Faculty of Engineering. In 1993, he moved to the Graduate School of Information Sciences of the same university, where he is the Head of Advanced Robotics Laboratory. His main research interests include office messenger robot, application of intelligent robot technologies for the disabled, mobile robots for rough terrain, omnidirectional mobile robot, multiple cooperative robot system, self adjustable manipulator, and practical visual sensing system.

Prof. Nakano founded the Robotics Society of Japan and acted as its Director from 1983 to 1987. He was a Director of the Society of Instrument and Control Engineers in 1988–1989. He initialized the Intelligent Robot Contest in 1989 and is the President of the Executive Committee of these games. He was Director of the Society of Biomechanism (1995–1997), Chairman of the Committee of the Development of Forestry Robot in the Forestry Ministry of Japan (1990–), Chairman of the Board of Directors of the Central Committee of International Robot Games Festival (1999–), and President of the Research and Development Consortium of Intelligent Patient Care System (1999–).